Wenyin Liu
Josep Lladós (Eds.)

LNCS 3926

Graphics Recognition

Ten Years Review and Future Perspectives

6th International Workshop, GREC 2005
Hong Kong, China, August 2005
Revised Selected Papers

Springer

Lecture Notes in Computer Science 3926

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Moshe Y. Vardi
Rice University, Houston, TX, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Wenyin Liu Josep Lladós (Eds.)

Graphics Recognition

Ten Years Review and Future Perspectives

6th International Workshop, GREC 2005
Hong Kong, China, August 25-26, 2005
Revised Selected Papers

Springer

Volume Editors

Wenyin Liu
City University of Hong Kong
Department of Computer Science
83 Tat Chee Ave., Kowloon, Hong Kong
E-mail: csliuwy@cityu.edu.hk

Josep Lladós
Universitat Autònoma de Barcelona
Computer Vision Center, Department of Computer Science
Edifici O, campus UAB, 08193 Bellaterra, Spain
E-mail: josep@cvc.uab.es

Library of Congress Control Number: 2006929221

CR Subject Classification (1998): I.5, I.4, I.3.5, I.2.8, G.2.2, F.2.2, H.4

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN 0302-9743
ISBN-10 3-540-34711-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-34711-8 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11767978 06/3142 5 4 3 2 1 0

Preface

This book contains refereed and improved papers presented at the 6th IAPR Workshop on Graphics Recognition (GREC 2005). This year is the tenth anniversary of GREC, which was started in 1995 and has been held every 2 years: GREC 1995 in Penn State University, USA (LNCS Volume 1072, Springer, 1996); GREC 1997 in Nancy, France (LNCS Volume 1389, Springer, 1998); GREC 1999 in Jaipur, India (LNCS Volume 1941, Springer, 2000); GREC 2001 in Kingston, Canada (LNCS Volume 2390, Springer, 2002); and GREC 2003 in Barcelona, Spain (LNCS Volume 3088, Springer, 2004).

GREC is the main event of IAPR TC-10 (the Technical Committee on Graphics Recognition within the International Association for Pattern Recognition) and provides an excellent opportunity for researchers and practitioners at all levels of experience to meet colleagues and to share new ideas and knowledge about graphics recognition methods. Graphics recognition is a particular field in the domain of document analysis, which combines pattern recognition and image processing techniques for the analysis of any kind of graphical information in documents from either paper or electronic formats. In its 10 year history, the graphics recognition community has extended its research topics from the analysis and understanding of graphic documents (including engineering drawings vectorization and recognition), to graphics-based information retrieval and symbol recognition, to new media analysis, and even stepped into research areas of other communities, e.g., sketchy interfaces and on-line graphics recognition, so as to face up to new challenges. These continuous changes show that we are a dynamic, active, and promising scientific community.

The program of GREC 2005 was organized in a single-track 2-day workshop. It comprised several sessions dedicated to specific topics. For each session, there was an overview talk, followed by a number of short presentations and concluded by a panel discussion. Session topics included "Engineering Drawings Vectorization and Recognition," "Symbol Recognition," "Graphic Image Analysis," "Structural Document Analysis," "Sketching and On-Line Graphics Recognition," and "Curve and Shape Processing." In addition, a special session of panel discussion was dedicated to the 10th anniversary of GREC, which focused on the summary of the achievements of GREC in the past 10 years and the planning of GREC in the next 10 years.

Continuing with the tradition of past GREC workshops, the program of GREC 2005 also included two graphics recognition contests: a symbol recognition contest, organized by Ernest Valveny and Philippe Dosch, and an arc segmentation contest, organized by Liu Wenyin. In these contests, test images and ground truths are prepared in order for contestants to have objective performance evaluation conclusions on their methods.

After the workshop, all the authors were invited to submit enhanced versions of their papers for this edited volume. The authors were encouraged to include ideas and suggestions that arose in the panel discussions of the workshop. Every paper was evaluated by two or three reviewers. At least one reviewer was assigned from the attendees of the workshop. Papers appearing in this volume were selected and most of

them were thoroughly revised and improved based on the reviewers' comments. The structure of this volume is organized in eight sections, reflecting the workshop session topics.

We want to thank all paper authors and reviewers, contest organizers and participants, and workshop attendees for their contributions to the workshop and this volume. Special thanks go to the following people: Miranda Lee for her great efforts in managing all logistic work; Wan Zhang, Tianyong Hao, and Wei Chen for their help in preparing the workshop proceedings and this volume; Karl Tombre for leading the panel discussion session dedicated to the tenth anniversary of GREC and providing an insightful summary of the discussion. Finally, we gratefully acknowledge the support of our sponsors: The City University of Hong Kong, IAPR, K. C. Wong Education Foundation, and The Hong Kong Web Society.

During the review process, we received the extremely sad news of the unexpected passing away of Adnan Amin. Adnan was an active researcher in the graphics recognition community. He participated in several GREC Workshops and was a member of the Program Committee of GREC 2005. He will be sorely missed by all of us. We would like to dedicate this book to the memory of Adnan.

The 7th IAPR Workshop on Graphics Recognition (GREC 2007) is planned to be held in Curitiba, Brazil, together with ICDAR 2007.

April 2006

Liu Wenyin
Josep Lladós

Organization

General Chair

Liu Wenyin, China

Program Co-chair

Josep Lladós, Spain

Program Committee

Sergei Ablameyko, Belarus
Gady Agam, USA
Adnan Amin, Australia
Dorothea Blostein, Canada
Eugene Bodansky, USA
Horst Bunke, Switzerland
Atul Chhabra, USA
Luigi Cordella, Italy
Bertrand Coüasnon, France
David Doermann, USA
Dave Elliman, UK
Georgy Gimelfarb, New Zealand
Jianying Hu, USA
Joaquim Jorge, Portugal
Young-Bin Kwon, Korea
Gerd Maderlechner, Germany
Daisuke Nishiwaki, Japan
Jean-Marc Ogier, France
Lawrence O'Gorman, USA
Tony Pridmore, UK
Eric Saund, USA
Jiqiang Song, China
Chew-Lim Tan, Singapore
Karl Tombre, France
Ernest Valveny, Spain
Toyohide Watanabe, Japan
Marcel Worring, Netherlands
Su Yang, China

Additional Referees

Thomas Breuel, Germany
Shijie Cai, China
Philippe Dosch, France
Alexander Gribov, USA
Xavier Hilaire, France
Pierre Leclercq, Belgium
Enric Martí, Spain
Jean-Yves Ramel, France
Gemma Sánchez, Spain
Alan Sexton, UK
Feng Su, China
Zhenxing Sun, China
Eric Trupin, France
Nicole Vincent, France
Zhiyan Wang, China

Table of Contents

Engineering Drawings Vectorization and Recognition

Symbol Recognition

Graphic Image Analysis

Structural Document Analysis

Sketching and On-Line Graphics Recognition

Curve and Shape Processing

Reports of Contests

Panel Discussion

Vectorization and Parity Errors

Alexander Gribov and Eugene Bodansky

Environmental System Research Institute (ESRI)
380 New York St., Redlands, CA 92373-8100, USA
{agribov, ebodansky}@esri.com

Abstract. In the paper, we analyze the vectorization methods and errors of vectorization of monochrome images obtained by scanning line drawings. We focused our attention on widespread errors inherent in many commercial and academic universal vectorization systems. This error, an error of parity, depends on scanning resolution, thickness of line, and the type of vectorization method. The method of removal of parity errors is suggested. The problems of accuracy, required storage capacity, and admissible slowing of vectorization are discussed in the conclusion.

Keywords: Raster Image, Vectorization, Medial Axis, Centerline, Accuracy, Error of Parity.

1 Introduction

The vectorization of monochrome images obtained by scanning line drawings consists of finding:

a) Mathematical lines (lines with zero thickness) describing appropriate raster linear objects (their location and topology)
b) Nodes (points of intersection of raster linear objects)
c) Shape functions that define the local thickness of the source raster lines.

The results of vectorization contain enough information for restoring the source image with a given accuracy.[1]

Authors of many papers use the term "skeleton" for the set of mathematical lines that are the result of the vectorization. We prefer the term "centerline", because the term "skeleton" often is used as a synonym for medial axes.

Medial axes are the locus of centers of maximal discs inscribed into the shape [1]. There is another definition equal to the previous one: medial axes of a form are a set of points internal to a form where each point is equidistant and closest to at least two points of the form boundary.

Each protrusion on the border of the geometrical form (in our case a form is a set of intersected raster lines) causes medial axis deviations. Even very small protrusions

[1] The vectorization task contains also dividing an image into linear objects and solids. The result of solid vectorization is a vector description of solid borders. But this part of the task of vectorization is beyond the scope of the paper.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 1 – 10, 2006.
© Springer-Verlag Berlin Heidelberg 2006

cause new branches and branching points (nodes). If we interpret each branch of a medial axis as a centerline, then the branches and appropriate nodes produced by deviations of borders have to be interpreted as erroneous or parasitic ones (see Fig. 1).

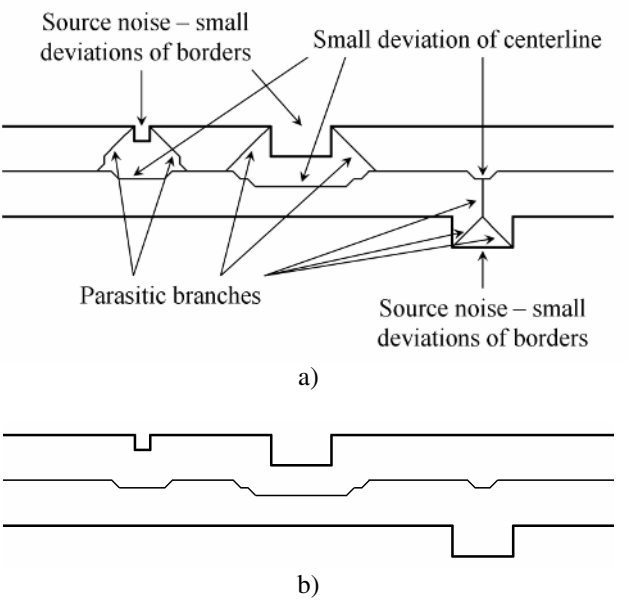


Fig. 1. Horizontal raster line with deviations of borders. a) Medial axis with parasitic branches, b) Medial axis without parasitic branches.

Medial axes have a strict geometrical definition, so they can be built without control parameters, thresholds, or any additional information except the borders of shapes.

Medial axes were suggested for a vector description of shapes. It is possible to use them for vectorization, but in no circumstances are they the results of vectorization because of parasitic branches and nodes.

There are methods of vectorization that build centerlines from medial axes by removing parasitic branches. We have already mentioned that parasitic branches are the results of noises, which are deviations of borders of raster lines from the true borders. Usually these noises have a probabilistic nature, so for building centerlines it is necessary to have some information about statistics of noises and desired signal.

So centerlines, as opposed to medial axes, do not have a strict deterministic definition [2, 3]. What part of a medial axis forms a centerline depends on the application domain, the type of line drawing, the resolving task, and many other reasons. This uncertainty is the main difficulty of the problem of vectorization and vectorization accuracy evaluation.

Nevertheless, it is obvious that in the simplest cases of horizontal and vertical raster lines with constant thickness, the centerlines have to be its axes of the symmetry. So in these cases we can easily evaluate errors of vectorization.

Fig. 2 and 3 show the results of vectorization obtained by two well-known commercial universal vectorization systems. It is easy to see that the results of vectorization have the following errors:

a) Centerlines of raster lines with thickness 2 and 4 pixels are shifted from axes of symmetry. We will call this error a parity error.
b) Many centerlines do not reach the ends of appropriate raster lines.
c) The ends of some centerlines essentially shifted from axes of symmetry.
d) A centerline showed on Fig. 3c is not horizontal.

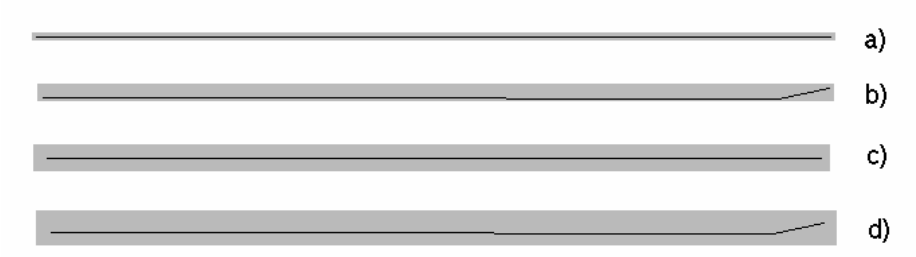


Fig. 2. Source raster lines and centerlines obtained with universal vectorization system A. Thickness of lines: a) 1 pixel, b) 2 pixels, c) 3 pixels, d) 4 pixels.

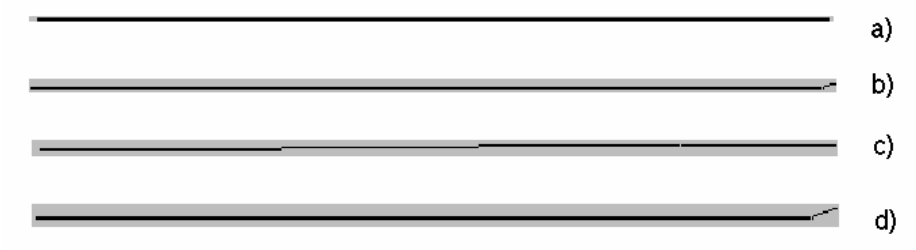


Fig. 3. Source raster lines and centerlines obtained with universal vectorization system B. Thickness of lines: a) 1 pixel, b) 2 pixels, c) 3 pixels, d) 4 pixels.

2 Three Classes of Methods of Raster Lines Vectorization

Many of the vectorization methods implemented in universal vectorization systems can be conditionally represented as three successive stages: pre-processing, raw vectorization, and post-processing.

The first stage consists of editing raster images (raster-to-raster conversion) to suppress of noises and improving the quality of the images. This stage is optional.

The second stage is a raster-to-vector conversion or raw vectorization. It can include removing parasitic branches. The result of this stage is a set of raw centerlines, which are represented with polygonal lines. The length of each segment of raw centerlines is about one pixel.

The third stage (vector-to-vector conversion) consists of processing raw centerlines to increase accuracy and data compression. Usually it includes gap closure,

smoothing, geometrical recognition, line type recognition, beautification, and compression.

Below, we will analyze only those methods of vectorization which include the stage of the raw vectorization (explicitly or implicitly).

Divide all vectorization methods into three classes:

A. Vertices of raw centerlines can be located only in the centers of pixels.
B. Vertices of raw centerlines can be located only in corners of pixels.
C. Vertices of raw centerlines can be located in arbitrary points.

Class A contains many methods based on thinning, distance transformation, and veinerization (see, for example, [3–6]).

This is not a full list of vectorization methods of class A. It is important only that all these methods have intermediate results that are thin raster lines with a thickness of 1 pixel. Such thin raster lines are often called digital lines. A raw centerline is a polygonal line with vertices located in the centers of each pixel of a digital line (see Fig. 4).

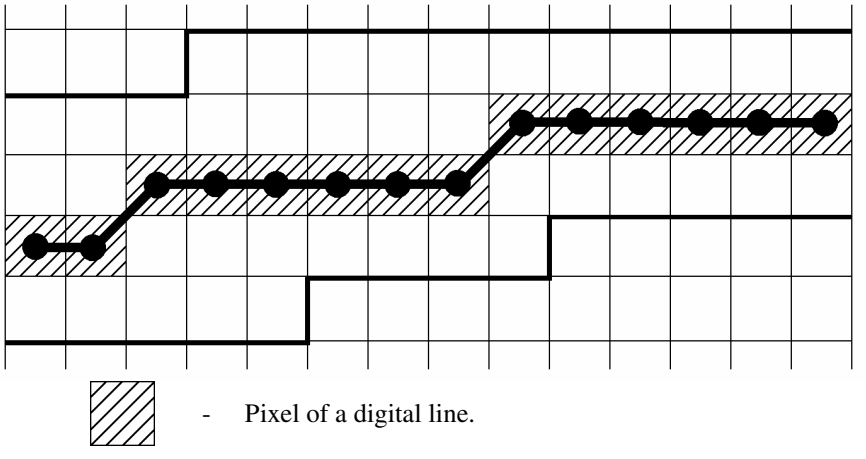


Fig. 4. Raw centerline with vertices in pixel centers

Class B contains a method that uses raster approximation of extended Voronoi cells [7]. In this method, a centerline follows between cells (along edges of cells), which are equidistant or almost equidistant from opposite sides of the raster line border.

Class C contains methods based on finding cross sections of raster lines. These cross sections pass through conjugated points on the opposite sides of the border of the raster line. The angle between a centerline and cross segments has to be as close as possible to 90 degrees and at least not less than 45 degrees. Connecting the middle points of these cross segments, it is possible to build lines that approximate the branches of the median axes. One such method is described in [8]. Another example of the vectorization method of class C is based on Voronoi tessellation [9].

3 Sources of Vectorization Errors

Above (Fig. 2 and 3) we saw that vectorization errors of horizontal raster lines with constant thickness depend on the parity of the thickness. We will show that parity errors depend only on the class of the vectorization method.

Suppose that a digital line obtained with some vectorization method of class A is a horizontal one. Because a thickness of the source raster line is even, centers of pixels of the digital line are located either above or below the axis of symmetry. Appropriate centerlines will be shifted by half a pixel, as we see it on Fig. 2b, 2d, 3b, and 3d, because class A centerlines pass through centers of pixels of a digital line.

If the thickness is odd, centers of pixels of a digital line will be located on the symmetry axis. We see this on the Fig. 2a, 2c, and 3a. So the results shown on Fig. 2 and 3 were obtained by vectorization methods of class A.

For vectorization methods of class B, a raw centerline can pass only between pixels. So we will watch an opposite result: the centerline of a raster line with even thickness will coincide with the axis of symmetry, and the centerline of a raster line with odd thickness will be shifted by half of a pixel compared to the axis of symmetry of the source line.

The vertices of a raw centerline obtained by vectorizing methods of class C can be located in arbitrary points of the raster line so these methods do not have errors of parity.

Some of the vectorization methods of class A could be implemented as methods of class C. It enhances the accuracy of vectorization because it removes errors of parity. The vectorization methods described in [10] and [11] are implemented as the methods of class A. They produce digital lines and, after that, the centers of pixels of the digital lines are used for building centerlines. But these methods can be implemented as methods of class C. It is possible to change raster-length chains with appropriate horizontal and vertical segments. Then it is possible to draw raw centerlines using the centers of these segments instead of pixel centers similarly to the algorithm described in [8].

We suppose that vectorization errors b) (centerlines do not reach the ends of appropriate raster lines) and c) (ends of some centerlines essentially shifted from axes of symmetry) have the same nature. Borders of raster lines form corners on the ends of the raster lines. So even for horizontal lines with constant thickness, median axes contain parasite branches (see Fig. 5).

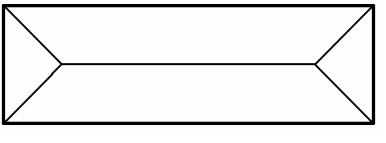


Fig. 5. Parasitic branches at the ends of horizontal raster line

After filtration of the parasitic branches, the raw centerlines do not reach the ends of the source raster lines.

Residual parasitic branches can cause a big deviation of centerlines from axes of symmetry. Some vectorization systems shorten raw centerlines to the several pixels and then build end parts of the centerline with approximation methods.

Unfortunately, such an approach often gives an incorrect and unstable solution (especially if noises, changing thicknesses, and curvatures of source centerlines are present). Building centerlines at end parts of raster lines is one of the least explored areas of the vectorization problem.

A sloped line instead of a horizontal one (see Fig. 3c) was obtained because of compression.

Many authors include the procedure of polygonal line compression [12] in the raw vectorization stage. For example, in [10] and [11], there is not a separate step for building a raw centerline. In these systems, the building of polygonal lines from digital lines is done simultaneously with compression.

We suppose that in universal vectorization systems, compression has to be done during post-processing. Deleting some vertices of the result of raw vectorization causes a loss of some information about the source raster line. We showed in [13] that before compressing raw centerlines, it is necessary to do noise filtering. Filtering can be done by smoothing or geometrical primitive recognition. Noise filtering can be also done by polygonal approximation based on the method of least squares [14, 15].

4 How to Eliminate Errors of Parity

Now that we know the cause of the shift of horizontal centerlines obtained by vectorization methods of classes A and B, it becomes clear that it is possible to eliminate these errors by modifying the source raster image. For vectorization methods of class A, it is necessary to use such a modification after which the thicknesses of any raster line will become odd. This can be done by superimposing a new raster plane on the source one.

It can be done in this way:

- The size h_{new} of a pixel of a new raster plane is equal to half the size of a pixel of a source raster plane h_{source}; $h_{new} = h_{source} / 2$;
- The shift of a new raster plane in comparison with the source raster plane in horizontal and vertical directions equals a half of a new pixel $h_{new} / 2$ (see a relative position of these raster planes on the Fig. 6).

A pixel of the new raster plane becomes a foreground pixel if it is overlapped (at least partially) with any foreground pixel of the source raster plane. The thickness of the first line on Fig. 6 in a source raster plane is one pixel (h_{source}), the thickness of the second line is two pixels ($2 h_{source}$). In a new raster plane both lines have an odd thickness ($3 h_{new}$ for the first line and $5 h_{new}$ for the second line).

Pay special attention to one effect of the suggested modification of the raster plane. Often, in vectorization methods of class A, the result of distance transformation is interpreted as a relief: the further a pixel is from the border, the greater its altitude.

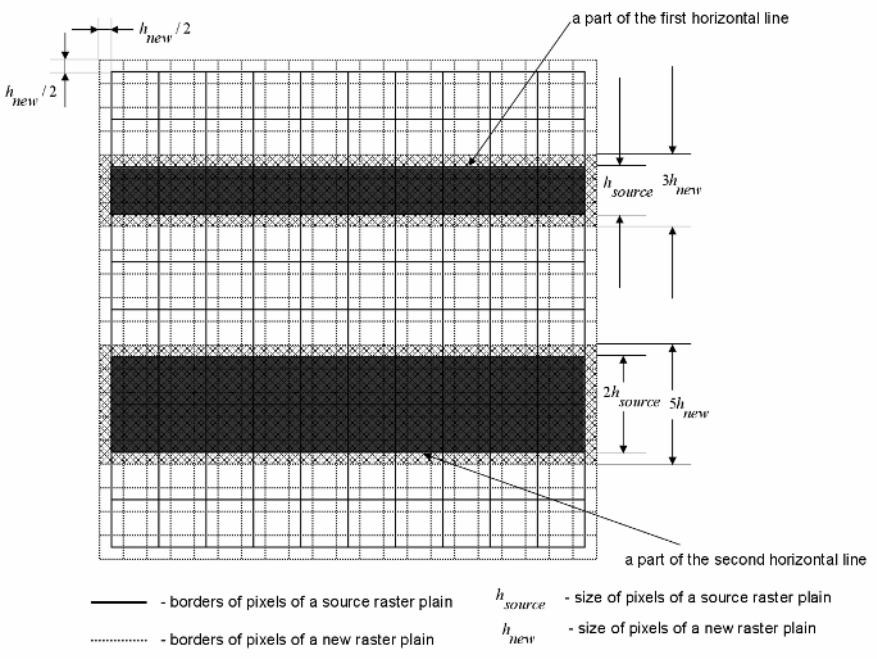


Fig. 6. A source and a new raster plane

This relief is not an arbitrary one: it does not contain any local minimums, i.e. a drop of liquid will slip until it reaches the border, if it does not stay on the ridge. We will let this property of the relief be called the "cascade" property. This kind of relief can include plateaus with a width of two pixels. Fig. 7a and 7b shows examples of undesirable trajectories of results of the vectorization of a raster linear object containing this type of plateau. When vectorization is done with the help of veinerization, it is necessary to use many further restrictions to stabilize the vectorization algorithm at these plateaus, i.e. to disallow the results shown in Fig. 7, and to find a trajectory similar in shape to the centerline (or the axis of symmetry, if one exists). Further, there is not a proof that these additional restrictions are sufficient to stabilize the vectorization algorithm. The suggested transformation of raster planes does not influence the "cascade" property of distance transformation, so the usual vectorization algorithms can be used without modification. The suggested transformation, however, adds a new property: a new relief cannot contain plateaus with a thickness two pixel. This significantly simplifies and increases the stability of veinerization algorithms, as well as some other algorithms that use distance transformation during vectorization.

Fig. 7. Undesirable results of vectorization

There will not be errors of parity for vectorization methods of class B if the thickness of horizontal lines in a new raster plane will be even. This can be achieved

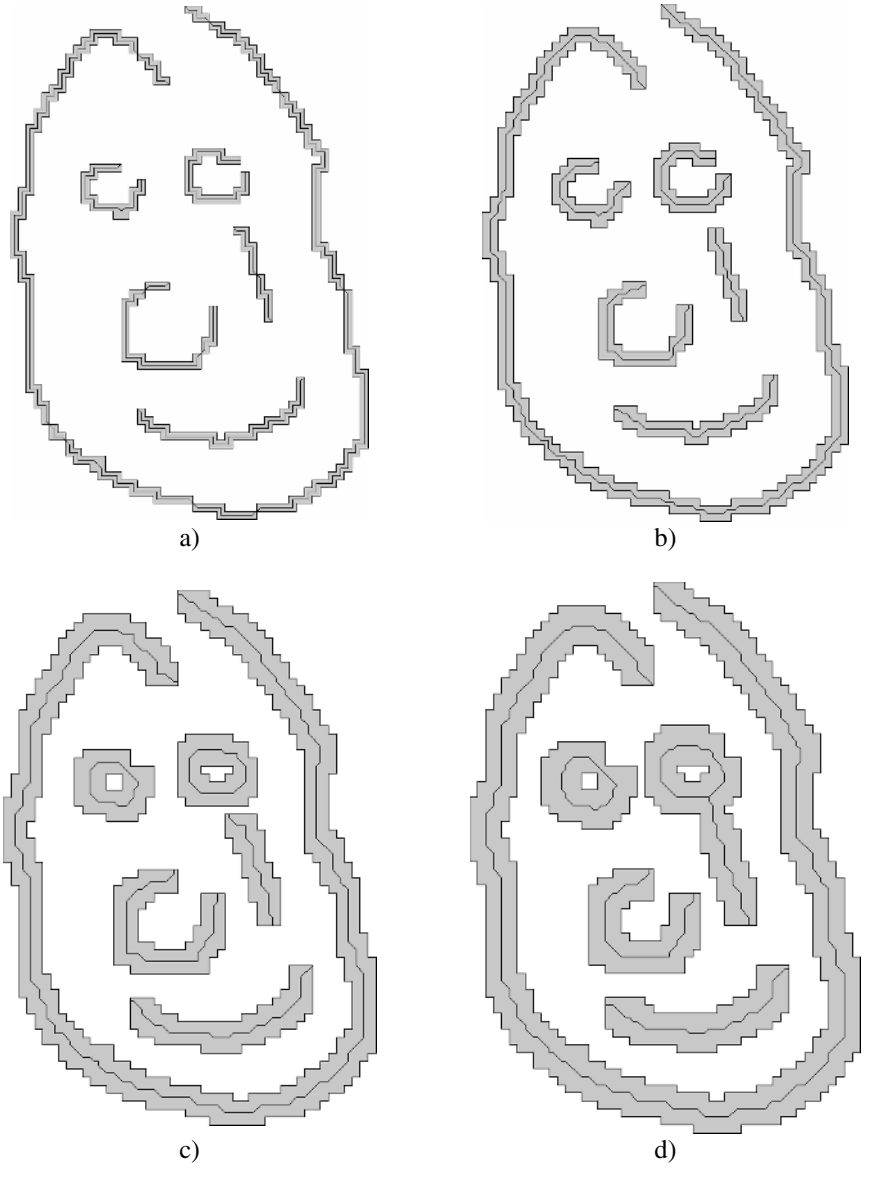


Fig. 8. The results of the raw vectorization of lines with compensation for parity errors. Thickness of lines is a) 1 pixel, b) 2 pixels, c) 3-4 pixels, d) 4-5 pixels. (Universal vectorization system ArcScan for ArcGIS).

very simply: as in the previous case, each old pixel has to be divided into four equal squares, but the new raster plane has not been shifted as compared to the source raster plane.

In Fig. 8 we show the results of the raw vectorization of free curves with different thickness. The results were produced by the universal vectorization system ArcScan for ArcGIS (ESRI), which uses the vectorization method of class A. To eliminate errors of parity this system uses modification of a raster plane described above. These figures illustrate symmetry of centerlines relative to the borders of the raster linear objects.

5 Conclusion

Thus we have determined that parity errors of vectorization of raster lines can be eliminated by transformation of a raster plane. This transformation is accompanied by increasing the quasi-resolution of the source raster image. So one of the results of such transformation is an increase in the size of the image (in pixels) by approximately four times.

This causes increasing required memory and quantity of calculation.

Is this acceptable? What is more important: precision of vectorization or vectorization speed?

There may be different answers to this question and they depend on the task that the user wants to solve. Nevertheless, we think that the main requirement for a universal vectorization system is precision. If we cannot obtain the result with required precision, the user will not use this vectorization system at all. Because we do not know users of universal vectorization systems in advance, we have to use the method that produces the result with the best achievable precision. If the customer does not need such a high level of precision, he can reduce the required memory and quantity of calculations by scanning line drawings at a lower resolution.

Of course, slowing down automatic vectorization is undesirable, but mostly it is permissible because it does not significantly increase operator time, which practically defines the cost of vectorization if the vectorization system is used intensively [2]. If it is necessary to obtain the result with high precision and it can be obtained by automatic or interactive vectorization, the operator time can be decreased by reducing the need for manual procedures of post-processing. Therefore, the cost of vectorization can be reduced in spite of slowing down raw vectorization.

Usually, universal vectorization systems have interactive modes, such as tracing and raster snapping [16], in addition to an automatic mode of raw vectorization. These interactive modes have to work in real time regardless of the size of the source document. This requirement defines an acceptable quantity of calculation for given characteristics of computers (speed, memory capacity, and so on).

Developing the universal vectorization system ArcScan for ArcGIS (ESRI) showed that it is possible to produce the necessary speed of interactive vectorization with the suggested methods of correction of parity errors.

References

1. Blum, H.: A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, Models for the Perception of Speech and Visual Form, Cambridge, MA, M.I.T. Press (1967) 362-380
2. Bodansky, E.: System approach to a raster-to-vector conversion: from research to conversion system. In: Levachkin, S., Bodansky, E., Ruas, A. (eds), e-Proceedings of International Workshop on Semantic Processing of Spatial Data (GEOPRO2002), Mexico City, Mexico (CD ISBN: 970-18-8521-X), 3-4 December (2002)
3. Deseilligny, M.P., Stamon, G., Suen, C.Y.: Veinerization: A New Shape Description for Flexible Skeletonization. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 20, No. 5 (1998) 505-521
4. Thinning methodologies for pattern recognition. Ed. by Suen, C.Y., Wang, P.S.P., Series in Machine Perception Artificial Intelligence, Vol. 8 (1994)
5. Rosenfeld, A., Pfaltz, J.L.: Sequential operations in digital picture processing. Journal of the Association for Computing Machinery, Vol. 13, No. 4 (1966) 471-494
6. Novikov, Yu.L.: An effective algorithms of raster images vectorization and their implementation in GIS. (Effektivniie algoritmi vektorizazii rastrovih izobrazhenii i ih realizaziia v geoinformazionnoi systeme) (in Russian), Ph.D dissertation, Tomsk State University, Tomsk (2002)
7. Gribov, A., Bodansky, E.: Vectorization with the Voronoi L-diagram. Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR), Vol. 2 (2003) 1015-1019
8. Elliman, D.: A Really Useful Vectorization Algorithm. Lecture Notes in Computer Science, Vol. 1941. Springer-Verlag (1999) 19-27
9. Ogniewicz, R., Kubler, O.: Hierarchic Voronoi Skeletons. Pattern Recognition, Vol. 28, No. 3 (1995) 343-359
10. Dori, D.: Orthogonal Zig-Zag: an algorithm for vectorizing engineering drawings compared with Hough Transform. Advances in Engineering Software, Vol. 28, No. 1 (1997) 11-24
11. Liu, W., Dori, D.: Sparse Pixel Vectorization: An Algorithm and Its Performance Evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21 (1999) 202-215
12. Douglas, D., Peucker, Th.: Algorithm for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer, Vol. 10, No. 2 (1973) 112-122
13. Bodansky, E., Gribov, A., Pilouk, M.: Smoothing and Compression of Lines Obtained by Raster-to-Vector Conversion. Lecture Notes in Computer Science, Vol. 2390. Springer-Verlag (2002) 256-265
14. Gribov, A., Bodansky, E.: A New Method of Polyline Approximation. Lecture Notes in Computer Science, Vol. 3138. Springer-Verlag (2004) 504-511
15. Gribov, A., Bodansky, E.: Reconstruction of Orthogonal Polygonal Lines. Lecture Notes in Computer Science, Vol. 3872. Springer-Verlag (2006) 462-473
16. Gribov, A., Bodansky, E.: How to Increase the Efficiency of Raster-to-Vector Conversion. Proceedings of the Fifth IAPR International Workshop on Graphics Recognition (GREC), Spain (2003) 225-232

A Vectorization System for Architecture Engineering Drawings

Feng Su, Jiqiang Song, and Shijie Cai

State Key Laboratory for Novel Software Technology,
Nanjing University,
210093 Nanjing, China
suf@graphics.nju.edu.cn

Abstract. This paper presents a vectorization system for architecture engineering drawings. The system employs the line-symbol-text vectorization workflow to recognize graphic objects in the order of increasing characteristic complexity and progressively simplify the drawing image by removing recognized objects from it. Various recognition algorithms for basic graphic types have been developed and efficient interactive recognition methods are proposed as complements to automatic processing. Based on dimension recognition and analysis, the system reconstructs the literal dimension for vectorization results, which yields optimized vector data for CAD applications.

1 Introduction

Automatic conversion of engineering drawings from the paper form to CAD formats has received much attention in recent years [1-5]. As a critical step in this process, a vectorization system converts a scanned drawing into the vector form, which is composed of basic graphic entities such as lines and text. In a broader sense, the recognition of certain drawing elements of higher semantic levels could also be incorporated into the vectorization process to achieve satisfactory results desired by the user.

For a vectorization system to be acceptable by users and made into effective usages, it should prove remarkable improvements in efficiency and precision, compared with manually redrawing the whole drawing with CAD tools from ground up. Complete system functions, efficient recognition algorithms and easy-to-use interaction methods are all key factors for a practical vectorization system.

This paper presents a vectorization system – VHVector for architecture engineering drawings. The system integrates both reliable automatic recognition algorithms and efficient interactive methods to achieve high usability. Necessary preprocessing and postprocessing are employed to accommodate various drawing images and produce optimized results compatible with common CAD requirements. The overall workflow of the system is outlined in Fig.1.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 11 – 22, 2006.
© Springer-Verlag Berlin Heidelberg 2006

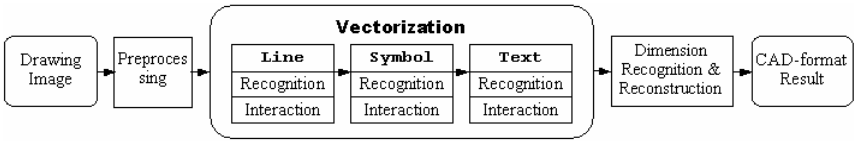


Fig. 1. Overview of the system workflow

2 Preprocessing

Scanned paper drawings may be of different formats and usually contain certain degradations like noises, stains, distortions, skewness or other defects either inherited from the original drawing or introduced by scanning. Therefore, preprocessing functions are implemented by the system to regularize the input image before vectorization:

- Image binarization, rotation, scaling, cropping for favorable image specifications
- Morphological filtering, smoothing and connected component based processing to remove noises and stains
- Raster editing tools for manual modification of the image content

Moreover, before the vectorization starts, two copies of the input image, the *working image* and the *reference image*, are created in an indexed format enabling efficient access to individual pixel values. The reference image remains identical with the input image throughout the vectorization process, while the content of the working image, though initially the same as the input image, is to be dynamically updated to reflect the current status during the processing. The system could restore the original raster that is incorrectly changed in the working image by copying from the reference image. And by switching the displaying between two images, the user could observe the differences between the vectorization result and the original image, making it much easier to discover undetected or misrecognized objects.

3 Recognition of Graphic Objects

Automatic recognition capabilities are essential for a vectorization system to achieve high efficiency and precision. The present system employs the Object-Oriented Progressive-Simplification-based Vectorization model proposed in [5] for recognition of various graphic objects in engineering drawings. This model features two important properties: 1) the system follows the line-symbol-text recognition workflow, in which graphic objects with simpler characteristics are vectorized earlier; 2) once a graphic object is recognized, pixels belonging exclusively to it are erased from the image being processed (Fig.2). As the result, the content of the working image is progressively simplified as the vectorization proceeds, and more important, intersecting or touching objects, which are common in practical drawings, could possibly be separated after one side of them is recognized and erased. Moreover, to take full advantages of this model, after the automatic recognition of one type of graphic object, it's recommended (but not obliged) to perform right away the interactive correction of all

recognition errors before move on to the recognition of the next graphic type. Vectorization in this manner not only makes it possible to exploit the interrelations between recognized and unrecognized objects to achieve better recognition result for the latter, but also avoids the accumulation and propagation of recognition errors between different graphic types, thus reduces the total amount of interactions required.

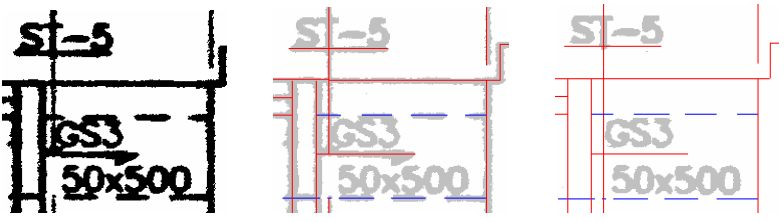


Fig. 2. The progressive simplification of the working image. Left to right: the original input image, line recognition results, the working image after recognized lines' pixels erased.

3.1 Recognition of Lines

In engineering drawings, lines cover bars (straight lines), circles, arcs and noncircular curves. Most vectorization methods for engineering drawings first convert the input image into a low-level raw-vector representation by skeletonization, and then rebuild graphic objects from those raw vectors. Due to missing original morphological information around intersecting parts, it is difficult for these methods to handle distortions at intersections and touching objects in the second phase. It also costs more computation and memory for these methods to search and merge all segments into final objects.

VHVector employs the Global Line Vectorization (GLV) algorithm [6] to recognize a line from the raster image directly in one step. It starts by scanning the image for a *seed segment*, which is a continuous rectangular region capturing the direction and the thickness of the target line. The seed segment should satisfy the fillness threshold represented by the percentage of the foreground pixels in the seed segment area, which indicates in general the raster quality of the line and is normally set to 0.9. Another threshold, length-to-width ratio, is used to balance between detection efficiency and linear precision and is set to 4.0 based on experiment results. With the seed segment attained, the algorithm extends the tracking path by the highly efficient Bresenham algorithm toward the two opposite directions indicated by the long axis of the seed segment, tuning the direction if needed in a small range according to the length distribution of perpendicular runs on each side of the tracking path, until it reaches the endpoints of the line. By this method, a line will be recognized as a whole with exact width measurement, no matter the complexity this line intersects with other objects, as long as a seed segment can be found - which is also the ground condition for a line to be recognized by human.

Furthermore, taking advantages of the intersection information recorded in the tracking process of individual lines, once the system has recognized one of a group of connected lines - called line network, other intersecting lines are tracked starting from the intersection point along the direction acquired by width analysis around

intersections. This omits the seed segment detection for most lines in a line network and generally produces more precise intersection positions than median-axis approaches.

Similarly, arcs and circles are detected by locating seed segments specific to circular shapes. The fact that the perpendicular bisectors of bar segments sequentially connecting a set of points on a circle meet at the circle centre is exploited to locate potential seed segments of arcs. Geometric parameters for circular tracking, including arc center, radius and line thickness are deduced from concentric bar segments detected and the Bresenham algorithm for circles is used for high tracking efficiency.

On the other side, however, the presented algorithm appears inadequate to handle freeform curves due to the difficulty finding an effective definition of the seed segment for tracking. Other methods, as thinning and approximating based ones, may serve as necessary complements to transform free curves into polyline or other favorable forms.

The erasion of line pixels after recognition while preserving raster parts shared by other objects is separated into two situations. As a whole, the erasion is performed in the rectangular area along the line and restricted by the line width. At the intersections, an analysis of the local contour tendency is performed for the intersecting object based on the variation and relative locations of perpendicular runs on each side of the line. If there exist branches of similar width on both sides, their connecting line serves as the rough track for preserving pixels. If only one branch is found, its pixels are preserved up to the center of the recognized line. Thresholds of the minimum size and distance of the branches are used to determine their existence and correspondence.

3.2 Recognition of Symbols

Existing symbol recognition approaches can be classified into two groups: statistical approaches and structural approaches. Statistical approaches use pixel as primitive to generate descriptors for symbol classification. Generally, they are domain-independent and insensitive to noises. However, most of these approaches rely on the proper segmentation of symbols before extracting features, thus are limited in applicability for practical architecture drawings without a robust segmentation scheme. On the other side, structural approaches decompose a symbol into primitives and relationships between them. They are relatively easy to adapt to component variations and scaling or rotating, thus resulting in a small set of models. But their effectiveness relies on the prior conversion of the symbol image into primitive vectors, which is errorprone itself.

In consideration of the significant differences between various symbols in architectural drawings, we divide symbols into two main categories, common symbols and domain-specific symbols, by the shape complexity and context characteristics and recognize them in different manners based on both raster and symbolic features.

Common symbols are normally of small sizes and have relatively simple shapes, but usually share the most amounts in total symbols of a drawing. Typical examples of such symbols include arrowheads, dots and short oblique lines (Fig.3), which are

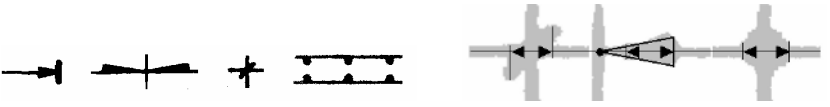


Fig. 3. Common symbols in architecture drawings. Left gives some symbol examples and right shows detected symbol locations and the matching of the adaptable symbol template.

generally basic constituents of some larger semantic structures, for example, dimension annotations. Symbols of this category usually have two notable characteristics - they are normally attached to certain lines as specific features, making it possible to locate them efficiently; their exact geometric characteristics are possibly varied or degraded, making it insufficient to recognize them purely based on matching of rigid templates. We employ a hybrid strategy to recognize these symbols directly from the binary image. Each symbol type is described by a set of discriminative shape features including symmetricity, centroid position and width variation and is associated with a specific procedure for dynamically building a raster template based on these properties. Then, to detect these symbols, connected components remaining in the working image along every vectorized line are inspected against symbol features. If successful, corresponding raster region in the reference image is matched against the raster template created and tailored using parameters retrieved from the previous step, evaluating both the normalized count of matching pixels and the conformability of symbol outline to determine the symbol type. Features of larger semantic structures, like dimensioning if available, are also exploited in detection and recognition of symbols.

Domain-specific symbols are generally comprised of a set of primitive graphic components bound together by some constraints. In architecture drawings, such symbols include elevation notations, section marks, fall marks, break signs, etc (Fig.4). We model each type of these symbols by a vector-based template, which can be seen as a constraint network describing the constituents of a symbol and relations between them. Each type of the geometric constraint, such as connecting, perpendicularity, parallelity, intersecting, is defined as a primitive and associated with a specific procedure that is used to detect symbol components related by the constraint. Moreover, certain components in a symbol template are marked as *primary components*, which serve as the leading elements for the recognition of the whole symbol and are generally chosen to be those semantically indispensable constituents of a symbol, for example, the leader line of the section and the elevation mark. More important, primary components could have additional properties describing the surrounding context of the symbol, that is, the connection between the symbol and other drawing elements, which can be effectively exploited as described below.

With predefined templates, the recognition of domain-specific symbols is started by detecting all potential primary components. Taking the fall mark shown in Fig.4 as example, every vectorized line longer than a threshold is inspected to try to find a perpendicular segment, which satisfies the definition of the primary line of the symbol, in vector data and possibly further in raster space by tracking in cases it has not been vectorized. For symbols whose primary components have no explicit connection with the context, normally they present as separate blocks from other drawing contents, thus can be identified from the connected components of proper size in the reference image by exhaustively matching and locating the primary component. Once

the primary component is found, the algorithm tries to locate other symbol components based on the constraints defined in the template, again in both the vector and the raster spaces. Though all currently considered architectural symbols are recognized solely based on line features, components in the form of solids (blobs) can also be integrated into the scheme once algorithms for detecting them are implemented. New symbol types can be added to the system by introducing corresponding template definitions.

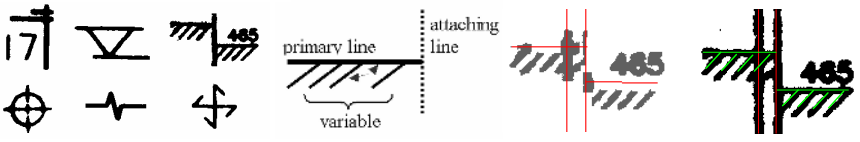


Fig. 4. Architecture specific symbols. Left to right: symbol examples, template for fall marks, recognized primary line of the symbol, other symbol components detected.

The erasion of symbol pixels after recognition is, correspondingly, separated into two cases. For common symbols, since they present as primitive features of lines, the erasion is performed to the residual raster blocks at symbol location. For domain specific symbols, the erasion is accomplished by erasing their constituent elements.

3.3 Recognition of Text

Generally, the recognition of text in engineering drawings can be divided into three successive steps: First, line-text segmentation separates raster regions belonging to text and those of other graphic objects. Next, textboxes that surround related text regions are extracted, whose orientation will be used as the direction of the potential text line. Last, individual characters inside the textbox are segmented and recognized.

Benefiting from the progressive simplification model, the line-text segmentation is implicitly performed in presented system by erasing recognized lines from the working image. Text blocks are then extracted from the remaining connected components that satisfy the size and distance thresholds, and grouped together to form the textbox based on spatial relations. To resolve the grouping ambiguity of text blocks belonging to several tightly spaced text lines, the directions of nearby recognized lines are exploited to provide additional hints about the possible combination directions.

A recognition-based character segmentation algorithm has been implemented in the present system. Every character is described by a stroke template, which defines shape properties of each stroke and normalized coordinates of a set of discriminative on-stroke/non-stroke points inside the character box. When passed a textbox, the algorithm scans it from left to right, progressively narrowing down the possible character candidates as shown in Fig.5. It first locates the leftmost stroke by aligning and matching all character templates at the left edge of the textbox, creating the initial character candidate set. Then, pixel values at predefined discriminative points of the template are checked to further filter out inappropriate candidates efficiently. Templates passing the test will be matched with the text block in the complex-to-simple order and measured for fitness in terms of the percentage of matching pixels along

strokes, which normally should be larger than 0.75 for acceptance. Once a match is found, the system performs segmentation after the rightmost stroke of the recognized character and continues this process on the new text block.

Fig. 5. Recognition-based character segmentation. From left to right are results of each step: locating the start stroke, filtering, template matching and segmentation.

4 Interactive Recognition

Interactive corrections of recognition errors are inevitable to achieve desirable vectorization results. To avoid entirely redrawing incorrect graphic objects, an efficient interactive recognition scheme is employed by the present system in a uniform operation pattern - pointing or stroking of the cursor. Usually, such operation conveys explicit information about the type and the rough location of the recognition target, so that the recognition algorithm can be invoked with relaxed thresholds or constraints, which normally yields improved recognition results with minimal interaction efforts.

For interactive recognition of lines, usually the position of a typical seed segment is given by the pointing of the cursor, and then the algorithm automatically determines the direction of the seed segment. For worse recognition conditions, such as presences of heavy noises, stains or overshort segments, when the algorithm can not figure out the right tracking direction, the user may roughly specify it by an additional pointing, i.e. a stroking of the cursor. Compared to manually digitizing the whole line, which is similarly comprised of two clicks, the proposed method relaxes the rigid demand of precisely locating the end points and in most cases comes up with appropriate measurement of line width and automatic relocation of vertices of connecting lines, which greatly improves the overall efficiency of interaction.

Sometimes, different selections of the seed segments by the operator might result in inconsistent endpoints or slightly different directions of lines, especially when the object raster line has irregular widths and the seed segments fall into different parts of it. However, owing to the capability of the vectorization algorithm to automatically tune the tracking direction, the influence of different seed segments is minimized.

For interactive recognition of symbols, similar pointing and stroking operations are used to indicate the position and optionally the direction of common symbols or the primary component of the symbol template. With symbol type explicitly provided, specialized recognition algorithms, like those based on high-level semantic structures, can also be implemented to effectively handle even badly degraded symbols.

The interactive recognition scheme is particularly useful and efficient in correcting recognition errors of text. One most common error type is the improper composition of the textbox, that is, the textbox of one string contains characters belonging to another string closely located due to the composition ambiguity. To correct such errors, usually the user has to delete then reenter and position both strings, which involves quite a number of operations. With interactive recognition, by comparison, the user just need to stroke the cursor crossing the string to indicate the proper composition

direction and extent as shown in Fig.6, then the algorithm will compute the new textbox, recognize it and try to further recompose all other affected textboxes. For example, the three composition errors in Fig.6 will be corrected by only single stroking operation.

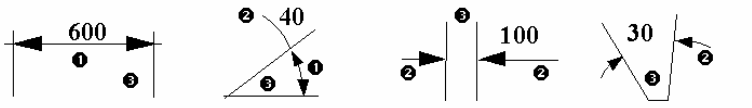
Fig. 6. Recognition-based interactive recomposition of text strings. Left illustrates the stroking operation for the correct text direction and right is the result of the automatic recomposition.

5 Dimension Sets Recognition

Dimension sets are one major type of annotation entities in engineering drawings, defining the exact geometric measurement (length or angle) of drawings entities. Based on the geometric characteristic being described, dimension sets can be categorized into two main groups namely *longitudinal* and *angular*. A dimension set is composed of a set of graphical elements, typically including a shape, a text, two symbols and two witness lines as illustrated in Fig.7. There could be variants in composition or representation of the dimension set for different annotating styles or requirements.

Several structural or syntax-based approaches for dimension recognition have been proposed [7-9]. Most of them perform a sequential searching and matching of the dimension structure, started by the successful recognition of a specific type of dimension component, usually the arrowhead or the text. However, the reliability of the recognition of individual dimension components could be affected by degradations present in scanned drawings. To avoid the recognition of the whole dimension set being blocked by the detection failure of a single fixed component, we employ a multi-entry algorithm proposed in [10] for improved robustness.

Component	Function
❶Shape	Represents geometric shape and direction of dimension
❷Tail	Extension of shape
❸Witness	Specifies the extent of the contour being dimensioned
❹Symbol	Marks the location of witness
❺Text	Specifies dimension type and value



(a) Dimensions with the shape component (b) Dimensions without the shape component

Fig. 7. Components and categories of dimension sets

For each dimension set type supported, one *component association table* (CAT) is defined, whose rows and columns correspond to the components of the dimension set and each cell describes the geometrical constraints between them. Another data structure, *dimension frame*, is used to record each incomplete dimension set already recognized during the processing and maintain a list of compatible candidate CATs. Based on these logical structures, the recognition algorithm is organized as a combination of two processes: the bottom-up detection of potential dimension frames and the top-down recognition of dimension components guided by dimension models.

The detection of initial dimension frames is based on searching all possible recognition entries, which are specific combinations of line and text components characterizing certain dimension structures. Currently, two main entries, the shape-text pair (Fig.7a) and the witness-tail-text pair (Fig.7b), are exploited as well as an entry based on special patterns of dimension text like 'ϕ8@100' or '3R10'.

With information about the potential dimension type and structure carried by the dimension frame, presently undetected components, especially those dimension symbols that have not been recognized during vectorization, are searched according to the constraints recorded in the CATs associated with the frame. Once complete, the dimension frame is converted into the final dimension set and its components are removed from the list of graphic entities representing the object geometry.

6 Dimension Reconstruction

By dimension reconstruction here, we do not mean tackling the whole problem of reconstructing the 3-D model of technically meaningful entities from the engineering drawing, which involves the complete functional analysis and matching of multiple views of an object. Instead, we restrict our objective in one preliminary step of the whole process, that is, recovering the literal dimension given by the dimension sets for every vectorized geometric entity in separate 2-D projections.

A commonly used approach for this purpose is based on the variational geometry theory, which converts geometrical constraints conveyed by dimension annotations to a set of algebraic equations about the coordinates of object vertices and then solves them to get the proper object coordinates. While theoretically well-founded, this approach appears insufficient for practical architectural drawings, in which dimensioning constraints conveyed by dimension sets could be incomplete, as those expressed otherwise by literal descriptions or conventions, or even accidentally inconsistent. To make it possible to locate such defects and still get an approximate solution or alternatively call for user intervention, we adopt a more algorithmic method [10] that introduces an intermediate structure - *grid* to integrate all dimensioning information and maintain correspondences between dimension sets and entities being dimensioned.

A grid is composed of a set of virtual grid lines parallel with the coordinate axes of the 2-D drawing space (Fig.8). Initially, the grid is populated with grid lines created at every coordinate position annotated by shape components of all longitudinal dimension sets, while each grid line maintains a list of all associated dimension sets at its position. Then, each entity to be reconstructed is represented by a set of *define points*, which uniquely determine the geometry of the entity. For example, the definition

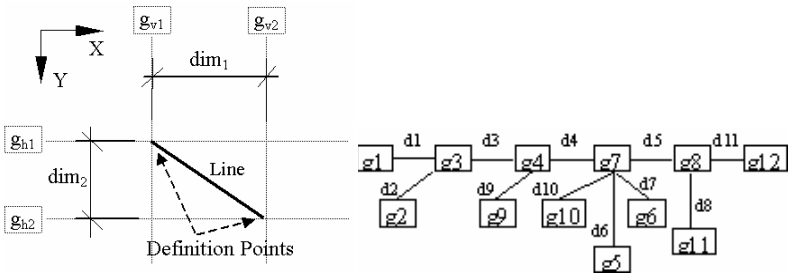


Fig. 8. Grid-based dimension analysis. Left shows grid lines for a dimensioned line, right shows a bind graph of a set of grid lines {g1-g12} and associated dimension constraints {d1-d11}.

points for an arc may consist of its two endpoints and either the center or another point on the arc, depending on which one is constrained by dimensioning. At each definition point two grid lines are created in orthogonal directions. For simplicity, we only consider longitudinal dimension sets in the grid, presuming other types of dimensional constraints are implicitly guaranteed after longitudinal ones have been satisfied.

Now that the grid contains all information about both the geometry and the dimensioning, the task of recovering literal dimension can be accomplished indirectly by adjusting coordinates of grid lines so that their distances satisfy associated dimension annotations. We use a *bind graph* for analysis of the relations between grid lines, in which every node corresponds to a grid line and an edge represents a dimensional constraint between two grid lines. Redundant dimensioning, which presents as multiple paths between two graph nodes, and incomplete dimensioning, which results in the disconnection of the graph, can be easily identified.

The whole algorithm executes as follows: first, a global scaling transformation is performed on all geometry entities with a scaling factor obtained by averaging individual geometric-to-literal length ratios of all dimension annotations in the drawing. Then, the grid structure is set up, while constraining unannotated entities with their transformed dimensions. By traversing all the edges of the bind graph, grid line coordinates are adjusted to conform to the constraints represented by each edge. Finally, definition points are retrieved from the grid and used to reconstruct geometric entities.

7 Results

We have tested the vectorization system with a number of architectural drawings. The typical results in terms of the recognition rate and speed of automatic vectorization are listed in Table 1. The sizes of drawings used for test range from A0 to A3 and most drawings are scanned with resolutions of 150-300 dpi. The typical execution time for automatic recognition of lines is less than 60 seconds for A0 size drawings up to 16000 x 12000 pixels with a recognition rate above 90%. The recognition rates of symbols and text depend on the conformity of templates used and the practical instances in the drawing. For drawings with considerable degradations or large numbers

Table 1. Experiment results of the vectorization system

Drawing type	Size (pixel)	Recognition rate / Speed (sec.)		
		Line	Symbol	Text
Section	3300*2100	1.00/14	0.90/10	0.98/17
Plane	7200*5000	0.94/25	0.85/32	0.94/45
Detail	13500*10600	0.92/47	0.73/65	0.71/112

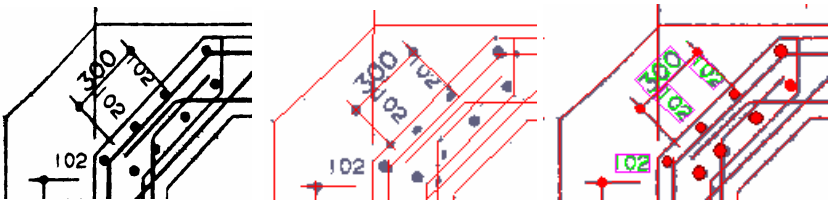


Fig. 9. Illustration of the vectorization process by an enlarged section of the input drawing. Left to right: the original drawing, line vectorization results, text and symbol recognition results.

of short segments and symbols or text of similar sizes, combination of automatic and interactive recognition methods usually yields better efficiency.

Fig.9 shows a typical vectorization process of an architectural drawing. We can see that the progressive simplification model significantly improves the system's ability for handling intersecting and touching objects. Although there still exist some insufficiencies in this model, the detection and recognition task of graphic objects have already been much simplified.

8 Summary

We have presented a vectorization system for architecture engineering drawings. Effective recognition algorithms are employed to convert lines, symbols and text in scanned paper drawings into their vector form. Based on the principle of recognizing graphic objects in the order of increasing complexity, a line-symbol-text processing workflow is employed, in which recognized objects are removed from the working image to minimize mutual interference of recognition of different graphic types. Efficient interactive recognition methods have also been proposed as the complement to automatic processing, correcting recognition errors in a uniform and effortless manner. The system also recognizes dimensioning structures present in the engineering drawing and reconstructs the entity geometry based on dimension analysis.

References

1. Tombre, K.: Analysis of engineering drawings: state of the art and challenges. Graphics Recognition - Algorithms and Systems, Springer-Verlag, Berlin, 1389:257-264, 1998
2. Dori, D., Liu, W.: Automated CAD conversion with the machine drawing understanding system: concepts, algorithm, and performance. IEEE Trans. on System, Man, and Cybernetics-part A: System and Humans. 29(4):411-416, 1999

3. Doermann, D. S.: An introduction to vectorization and segmentation. Graphics Recognition - Algorithms and Systems, Springer-Verlag, Berlin, 1389:1-8, 1998
4. Yu, Y., Samal, A., Seth, S. C.: A system for recognizing a large class of engineering drawings. IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(8):868-890, 1997
5. Song, J., Su, F., Tai, C.-L., Cai, S.: An object-oriented progressive-simplification-based vectorization system for engineering drawings: model, algorithm, and performance. IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(8):1048-1060, 2002
6. Song, J., Su, F., Cheng, J., Cai, S.: A knowledge-aided line network oriented vectorization method for engineering drawings. Pattern Analysis and Application, 3(2):142-152, 2000
7. Dori, D.: A syntactic/geometric approach to recognition of dimensions in engineering machine drawings. Computer Vision, Graphics and Image Processing, 47:271-291, 1989
8. Lai, C. P., Kasturi, R.: Detection of dimension sets in engineering drawings. IEEE Trans. on Pattern Analysis and Machine Intelligence, 16(8):848-855, 1994
9. Lin, S. C., Ting, C. K.: A new approach for detection of dimensions set in mechanical drawings. Pattern Recognition Letters, 18(4):367-373, 1997
10. Su, F., Song, J., Tai, C.-L., Cai, S.: Dimension recognition and geometry reconstruction in vectorization of engineering drawings. IEEE Proc. of Conf. on Computer Vision and Pattern Recognition, Hawaii, 1:710-716, 2001

Musings on Symbol Recognition

Karl Tombre[1], Salvatore Tabbone[2], and Philippe Dosch[2]

[1] LORIA-INPL, École des Mines de Nancy, Parc de Saurupt,
CS 14234, 54042 Nancy CEDEX, France
Karl.Tombre@loria.fr
[2] LORIA-Université Nancy 2, Campus scientifique,
B.P. 239, 54506 Vandœuvre-lès-Nancy CEDEX, France
{Salvatore.Tabbone, Philippe.Dosch}@loria.fr

Abstract. In this paper, we review some ideas which emerged in the early years of research on symbol recognition and we show how these ideas evolved into a large variety of contributions. We then propose some interesting challenges for symbol recognition research in the present years, including symbol spotting methods, recognition procedures for complex symbols, and a systematic approach to performance evaluation of symbol recognition methods.

1 Introduction

Symbol recognition is a field within graphics recognition to which a lot of efforts have already been devoted. However, a document analysis expert who is more familiar with OCR might rightfully wonder what exactly we call a symbol and how symbol recognition differs from basic character recognition.

Our feeling is that the problem is very different because of the much higher number and variety of symbols to be recognized. Except in strongly context-dependent applications, it is impossible to provide a database of all possible symbols. It is also in many cases impossible to assume that symbol recognition can be performed on clearly segmented instances of symbols, as symbols are very often connected to other graphics and/or associated with text. The well-known paradox therefore appears: in order to correctly recognize the symbols, we should be able to segment the input data, but in order to correctly segment them, we need the symbols to be recognized!

This in turn means that it is usually not possible to assume that a reliable segmentation process is available, that the symbols have been clearly extracted, normalized, etc. It is hence not reasonable to assume that a vector of general-purpose features can be computed on the segmented areas deemed to be potential symbols, in such a way that the vector can be classified by some appropriate statistical pattern recognition method. The most common approach in symbol recognition therefore relies on structural methods able to capture the spatial and topological relationships between graphical primitives; these methods are sometimes complemented by a classification step, once the candidate symbol has been segmented or spotted.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 23–34, 2006.
© Springer-Verlag Berlin Heidelberg 2006

This paper does not pretend to be yet another survey on symbol recognition methods, as several excellent surveys already exist [4, 6, 21]. We will rather try to take a step back, look at the main efforts done in that area throughout the years and propose some interesting directions to investigate.

2 A Quick Historical Overview

As previously said, the early specific work on symbol recognition, as opposed to character recognition, emphasized the use of structural pattern recognition techniques, as usual statistical classification techniques were not suitable. Early efforts included template matching techniques [13], grammar-based matching techniques [8] and recognition techniques based on structural features [11] and dynamic programming [24].

When dealing with specific families of symbols, techniques similar to OCR could be used; this is the case for symbols having all a loop [25] or for music recognition [31]. However these techniques have their own limitations, in terms of computational complexity and of discrimination power.

Very early, people therefore became aware that graph matching techniques are especially suited to specificities of symbol recognition. Twenty years ago, Kuner proposed the search for graph or subgraph isomorphisms as a way for matching symbols with models [14]. Groen et al. [9] analyzed electrical wiring diagrams by representing symbol models by graphs, and using probabilistic matching techniques to recognize symbols. Lin et al. [17] similarly matched symbols to be recognized with model graphs using graph distance computations.

Although simple, this basic idea of graph matching suffers from a number of drawbacks. In its basic principle, it is sensitive to errors and noise; as we usually cannot assume that segmentation is perfect nor reliable, this means that the graphs to be processed can also have a number of extra or missing nodes and vertices. Very early, authors dealt with the general problem of inexact graph matching [34]. In later years, seminal work by Horst Bunke's team has brought to evidence that it is possible to design error-tolerant subgraph isomorphism algorithms [3, 23]. Another possible approach is to make statistical assumptions on the noise present in the image [26].

Another problem with graph matching is the computational complexity of subgraph isomorphism methods. A lot of efforts have therefore been devoted to optimizing the matching process through continuous optimization [15] or constraint propagation techniques to perform discrete [10, 44] or probabilistic [5] relaxation.

Still, another problem remains: that of the scaling of such structural methods to encompass a large number of candidate symbols. It remains to be proven that a symbol recognition method based on graph matching can successfully scale to a large number of model symbols. Also, it is seldom feasible to directly search for subgraph isomorphisms on a whole drawing or document, without some kind of segmentation or pre-segmentation.

Therefore, although there have been a number of successful complete symbol recognition systems, these are mostly within areas with relatively few kinds of

symbols to discriminate and within areas where it is easy to localize or pre-segment potential symbols. This includes electrical wiring diagrams [8, 9, 16, 17] and flowcharts [24], typical areas where pre-segmentation can be performed quite easily through separation on the graphical layer between connecting lines and complex areas which are assumed to be symbols. Some attempts have also been made at recognition in areas where pre-segmentation is not easy; this includes work in our own group on recognition of architectural symbols by propagating basic graphical features through a network of nodes representing structural and geometrical constraints on how these features are assembled into symbols [1]. This approach makes it possible to group the information represented by each structural symbol model into a single network, but it remains prohibitively expensive and complex in terms of memory use when the number of model symbols grows. In addition, the fact that the system has to work with noisy data leads to using a number of local rules for inexact matching, and when this propagates through the network there is a real danger of recognizing everything everywhere!

3 Challenges and Research Directions

On the basis of the capabilities and limitations of structural symbol recognition methods, as surveyed above, we discuss in this section a number of interesting challenges and research directions in which our group is currently working.

3.1 The Right Information in the Right Place

Despite their limitations, structural recognition methods provide powerful tools for dealing with complex information. This stems from the large representational power of a graph, as a structure to capture pieces of information and the relationships between these pieces. Attributed relational graphs (ARG) are especially suitable for supporting the structural representation of symbols [26].

But a first challenge is to *put the correct information into the graph.* A typical natural, but often simplistic, and sometimes even wrong way of proceeding is to use the result of some raster-to-vector process to build a graph where the vertices would be the vectors and the nodes the junctions between the vectors. This leads to representing a symbol as a set of graphical features and the spatial relations between these features, represented by relational attributes. Of course, we are aware that it is not enough to have good features in the right place of the graph; the matching method also has to be robust to noise [2].

Adding higher-level topological, geometrical and relational information to the nodes and vertices of the graph can open up new possibilities in recognition problems. When some pre-segmentation methods can divide the image to be analyzed into homogeneous regions, region adjacency graphs are a good candidate as they capture a lot of interesting information [19]. When this is not possible, it may make sense to start with extracting simple graphical features which can be reliably found without prior segmentation: vectors, arcs, basic shapes, and to use a graph where these basic features are attributes of the nodes and the

vertices convey information about topological and geometrical relationships between these features (inside, above, at-right-of, touching, etc.) A good example of such use of spatial relations for symbol recognition purposes is the system built by Liu Wenyin's team in Hong Kong [18, 46].

3.2 Symbol Spotting

A way to avoid the dilemma of needing segmentation to perform recognition, and vice-versa, is to try to localize the symbols in a complex drawing without necessarily going all the way through complete recognition. This gives first pieces of information on the subareas in which to apply recognition methods which may be more computationally expensive.

In order to overcome the segmentation vs. recognition paradox, we have worked in the last years on *symbol spotting* methods, i.e. ways to efficiently localize possible symbols and limit the computational complexity, without using full recognition methods. This is a promising approach, based on the use of signatures computed on the input data. We have worked on signatures based on force histograms [38] and on the Radon transform [36, 37], which enable us to localize and recognize complex symbols in line-drawings. We are currently working on extending the Radon signature to take into account photometric information, in order to improve the results when retrieving similar symbols in graphical documents [39]. By using a higher-dimensional signature, we are able to include both the shape of the object and its photometric variations into a common formalism. More precisely, the signature is computed on the symbol, at several grey levels. Thus, the effects of noise on the boundary of the processed object become insignificant, relatively to the whole shape.

When it comes to spotting target symbols, structural approaches are powerful in terms of their representational capabilities. Therefore, we use a simple structural representation of symbols to introduce a hybrid approach for processing symbols connected to other graphical information. For this, we compute a skeleton and we organize its junction points into a graph where the graph edges describe the link between junction points (see Fig. 1). From this representation, candidate symbols are selected and validated with the signature descriptor. Fig. 2 illustrates the working of the system: when a candidate symbol is selected in the document, a number of candidate regions are retrieved.

3.3 Measuring the Progress: Performance Evaluation

As in many other areas within pattern recognition, performance evaluation has become a crucial part of our work on symbol recognition, in order to be able to compare different methods on standard datasets with metrics agreed upon by everyone. Our team co-organized the two first international contests on symbol recognition, held at GREC'03 [40, 41] and at GREC'05. The basic principles are as follows:

– The *datasets* include both real scanned symbols and synthetic data, i.e. symbols stemming from CAD systems which were degraded using a combination

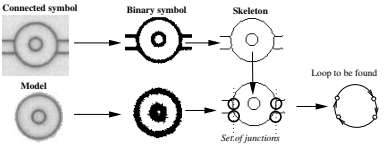


Fig. 1. Example of graph organization based on the junction points(from [39])

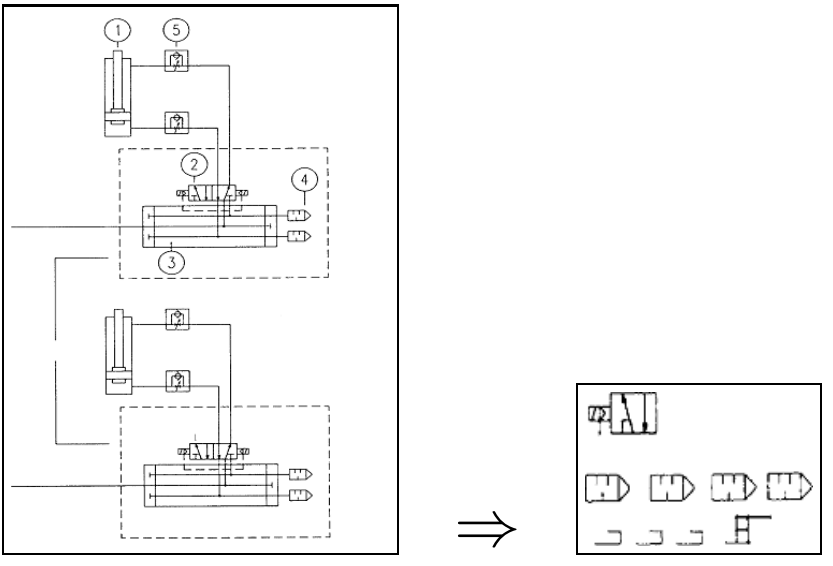


Fig. 2. Example of symbol spotting on an engineering drawing, from [48]. The user delineates a symbol (left) and a number of candidates are retrieved (right).

of an image degradation model [12] and of vectorial degradation [42]. Other basic transformations, such as scaling and rotation, are also used.

There are two types of datasets: isolated symbols (pre-segmented) for which the task is to recognize a symbol among n possible models, with various measures for an increasing n and an increasing degradation of the data, and symbols in their context (without segmentation) where there is a double task of spotting/localizing the symbol, and then recognizing it. Note that although most of the framework was in place, we finally decided not to run the symbol localization part at the second contest.

Managing a great number of heterogeneous data may be confusing for participant methods, sometimes designed for a specific purpose, as well as for post-recognition analysis steps, that could be irrelevant if the results are themselves too heterogeneous. Therefore, all datasets are classified according to several properties, increasing the readability in both cases. Basically, these properties are defined either from a technological point of view (bitmap/vectorial representation, graphical primitives used...) or from an application point of view (architecture/electronic...)

The datasets are further divided into training data made available to the participants beforehand, and test data used during the contest itself.
- The *ground-truth* definition for symbols in their context is simple and readable. It is basically based on the manual definition of bounding-boxes around each symbol of the test data, labelled by the model symbol.
- The *performance measures* for isolated symbols include the number of false positives and missing symbols, the confidence rates (when provided by the recognition method), computation time (which gives an implicit measure of the complexity of the method) and scalability, i.e. a measure of the way the performances decrease when the number of symbols increases.
  The performance evaluation for symbols in their context is based on two measures. The first is unitary and is related to each symbol. It is based on the overlapping of a ground-truth bounding-box and a bounding-box supplied by a participant method, in the case where both symbol labels are the same. The second measure allows us to compose all unitary measures for a test data, and is based on the well-known notions of precision and recall ratios. Again, computation time is used to qualify the scalability of the participant method.
- Finally, the *results analysis* is led from the data point of view (data based), as well as from the methods point of view (methods based). Indeed, if it is interesting to understand which methods give good results with a lot of data, it is also interesting to understand which data are difficult to recognize with respect to the several recognition approaches. The interest of a performance evaluation campaign is guided by these two points of view.
- The general framework provides online access to training data and description of the metrics used.

In addition, our team is leading a project financed by the French government but open to international teams, on the performance evaluation of symbol recognition and logo recognition (see `http://www.epeires.org/`). The purpose of this project is to build a complete environment providing tools and resources for performance evaluation of symbol recognition and localization methods. This environment is intended to be used by the largest possible community. A test campaign, opened to all registered participants, will be organized during its final step. In addition to providing the general framework for organizing benchmarks and contests on a more stable basis, our goal is to make available for the community a complete environment including online collaborative ground-truthing tools, reference datasets, results of already published methods on these datasets, and performance metrics which can be used for research teams throughout the world to compare their own work on symbol localization and/or recognition with the state of the art.

3.4 Complex Symbols

In many cases, a symbol is not only a set of segments and arcs, but a complex entity associating a graphical representation, a number of connection points and

text annotations. Symbol recognition should be able to deal with such *complex symbols* in order to be of practical use in a number of areas.

Figure 3 gives some examples of complex symbols from the area of aeronautics (wiring diagrams of an Airbus plane). The challenge here is to be able to

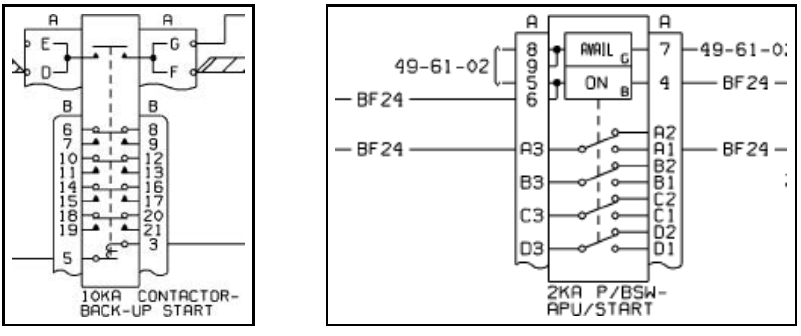


Fig. 3. Examples of complex symbols

discriminate between symbols which may differ not by their graphical shape, nor by their topology, but simply by the number of connectors or by the type of textual annotations. As an example, Fig. 4 illustrates two complex symbols from the area of electrical design which differ only by slight variations in the shape of their upper constituent sub-symbols.

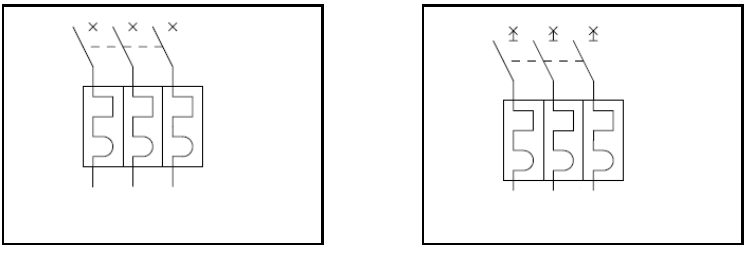


Fig. 4. Example of very similar symbols (courtesy Algo'tech Informatique)

We are still working on the appropriate strategy to deal with this kind of recognition problems. One of our ideas is to compile, from the set of reference symbols, a number of basic shapes which can be considered as the basic building blocks for drawing such symbols: rectangles, triangles, squares, disks, horizontal and vertical segments, other straight segments, arcs, etc. Some of these shapes may be filled and are thus represented by their contour. Then, very simple recognition agents would localize in the drawing all instances of these simple shapes, and progressively remove them from the drawing, to simplify it, following the basic principle applied by Song et al. to the vectorization problem [35]. Complex symbols can then be represented by rules for assembling these basic shapes,

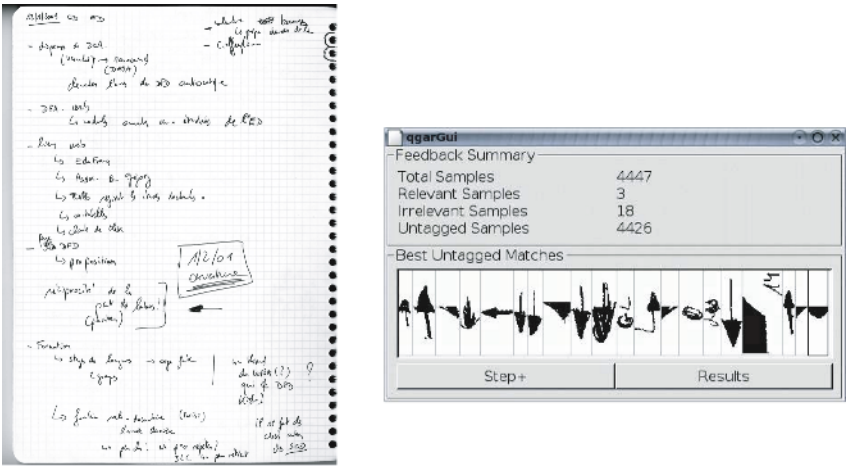


Fig. 5. Working of first prototype for dynamic recognition on scanned handwritten notebooks such as the example on the left side (from [30]). The idea in this example is to retrieve the arrows written by the user.

the annotations present in the text layer, the connection information from the vectorized connecting lines, and other spatial information.

Pasternak was one of the first to experiment with this kind of recognition strategy, with a system combining a number of simple agents triggering assembly rules for recognizing higher-level symbols [27, 28]. However, his system remained complex to adapt and to use in practical applications. We have started to work on this problem and we plan to use structural/syntactic methods such as graph grammars [7, 20, 29, 32] to describe the combination rules leading from the simple shapes and the annotations to complex symbols.

3.5 Dynamic Recognition

Until now, we have addressed the problem of recognizing a symbol among a set of known models. But there are situations, especially when browsing an open set of documentation, where nobody is able to build a library of model symbols or even to predict which symbols the user may be interested in. In that case, we have to rely on what we have called dynamic symbol recognition. The idea is that the user interactively selects an area or a region of a document which (s)he calls a symbol. The challenge is then to retrieve other instances of this symbol in the same document or in other documents available in the digital library.

The system can of course include some relevance feedback mechanism allowing the user to validate or invalidate the results of a first symbol spotting phase and then restart the whole process.

To achieve this, one of our ideas is to rely on a set of simple features which can be pre-computed on the digital library. Each document image can be divided into small images, using either a simple meshing method, or some rough document image segmentation technique. On each subimage obtained through

this subdvision or segmentation, one or several signatures, based on the Radon transform or on other generic descriptors [22, 45, 47], can be used to characterize the subimage. When the user selects a part of the image, the descriptors of this part are computed and some distance can be used to find the regions of interest having the closest descriptors. Relevance feedback allows the user to validate or invalidate the different symbols spotted in this way, and the mechanism can be iterated until the user is satisfied with the result.

The scenario sketched above only represents some preliminary ideas on this matter of dynamic recognition, which is ongoing work in our team. Figure 5 illustrates the working of our first prototype, presented in [30].

4 Conclusion

In this paper, we have reviewed some ideas which emerged in the early years of research on symbol recognition and have tried to show how these ideas evolved into a large variety of contributions, which for many of them are based on the same structural recognition paradigm. We have then proposed some challenges for symbol recognition research in the present and coming years.

We are aware that there are a number of other issues which we have not dealt with in this paper. Let us just mention the necessity of combining various approaches to achieve better global recognition results. This includes combining structural and statistical methods, but also combining various descriptors in a better way than simply putting them into a vector where each feature computed is assumed to play the same role and have the same weight. Some first results have been obtained in our group using the Choquet integral to aggregate various descriptors for better symbol spotting and recognition [33, 43].

Symbol recognition has been a research topic for many years already, and spectacular achievements have been obtained. Still, a number of issues remain open and lead to a number of research challenges for the coming years.

References

1. C. Ah-Soon and K. Tombre. Architectural Symbol Recognition Using a Network of Constraints. *Pattern Recognition Letters*, 22(2):231–248, February 2001.
2. H. Bunke. Error-Tolerant Graph Matching: A Formal Framework and Algorithms. In A. Amin, D. Dori, P. Pudil, and H. Freeman, editors, *Advances in Pattern Recognition (Proceedings of Joint IAPR Workshops SSPR'98 and SPR'98, Sydney, Australia)*, volume 1451 of *Lecture Notes in Computer Science*, pages 1–14, August 1998.
3. H. Bunke. Error Correcting Graph Matching: On the Influence of the Underlying Cost Function. *IEEE Transactions on PAMI*, 21(9):917–922, September 1999.
4. A. K. Chhabra. Graphic Symbol Recognition: An Overview. In K. Tombre and A. K. Chhabra, editors, *Graphics Recognition—Algorithms and Systems*, volume 1389 of *Lecture Notes in Computer Science*, pages 68–79. Springer-Verlag, April 1998.

5. W. J. Christmas, J. Kittler, and M. Petrou. Structural Matching in Computer Vision Using Probabilistic Relaxation. *IEEE Transactions on PAMI*, 17(8): 749–764, August 1995.
6. L. P. Cordella and M. Vento. Symbol recognition in documents: a collection of techniques? *International Journal on Document Analysis and Recognition*, 3(2): 73–88, December 2000.
7. H. Fahmy and D. Blostein. A Survey of Graph Grammars: Theory and Applications. In *Proceedings of 11th International Conference on Pattern Recognition, Den Haag (The Netherlands)*, volume 2, pages 294–298, 1992.
8. C. S. Fahn, J. F. Wang, and J. Y. Lee. A Topology-Based Component Extractor for Understanding Electronic Circuit Diagrams. *Computer Vision, Graphics and Image Processing*, 44:119–138, 1988.
9. F. C. A. Groen, A. C. Sanderson, and J. F. Schlag. Symbol Recognition in Electrical Diagrams Using Probabilistic Graph Matching. *Pattern Recognition Letters*, 3: 343–350, 1985.
10. A. H. Habacha. A New System for the Analysis of Schematic Diagrams. In *Proceedings of 2nd International Conference on Document Analysis and Recognition, Tsukuba (Japan)*, pages 369–372, 1993.
11. E. Hansen and K. P. Villanger. A Combined Thinning and Contour Tracing Approach to the Recognition of Engineering Drawing Symbols. In *Proceedings of International Seminar on Symbol Recognition, Oslo (Norway)*, pages 82–100, 1985.
12. T. Kanungo, R. M. Haralick, H. S. Baird, W. Stuezle, and D. Madigan. A statistical, nonparametric methodology for document degradation model validation. *IEEE Transactions on PAMI*, 22(11):1209–1223, November 2000.
13. W. Kikkawa, M. Kitayama, K. Miyazaki, H. Arai, and S. Arato. Automatic Digitizing System for PWB Drawings. In *Proceedings of 7th International Conference on Pattern Recognition, Montréal (Canada)*, volume 2, pages 1306–1309, 1984.
14. P. Kuner. Efficient Techniques to Solve the Subgraph Isomorphism Problem for Pattern Recognition in Line Images. In *Proceedings of 4th Scandinavian Conference on Image Analysis, Trondheim (Norway)*, pages 333–340, 1985.
15. P. Kuner and B. Ueberreiter. Pattern Recognition by Graph Matching — Combinatorial versus Continuous Optimization. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):527–542, 1988.
16. S.-W. Lee. Recognizing Hand-Drawn Electrical Circuit Symbols with Attributed Graph Matching. In H. S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis*, pages 340–358. Springer-Verlag, Heidelberg, 1992.
17. X. Lin, S. Shimotsuji, M. Minoh, and T. Sakai. Efficient Diagram Understanding with Characteristic Pattern Detection. *Computer Vision, Graphics and Image Processing*, 30:84–106, 1985.
18. Y. Liu, L. Wenyin, and C. Jiang. A Structural Approach to Recognizing Incomplete Graphic Objects. In *Proceedings of the 17th International Conference on Pattern Recognition, Cambridge (UK)*, August 2004.
19. J. Lladós, E. Martí, and J. J. Villanueva. Symbol Recognition by Error-Tolerant Subgraph Matching Between Region Adjacency Graphs. *IEEE Transactions on PAMI*, 23(10):1137–1143, 2001.
20. J. Lladós and G. Sánchez. Graph matching versus graph parsing in graphics recognition — A combined approach. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3):455–473, 2004.

21. J. Lladós, E. Valveny, G. Sánchez, and E. Martí. Symbol Recognition: Current Advances and Perspectives. In D. Blostein and Y.-B. Kwon, editors, *Graphics Recognition – Algorithms and Applications*, volume 2390 of *Lecture Notes in Computer Science*, pages 104–127. Springer-Verlag, 2002.
22. S. Loncaric. A Survey of Shape Analysis Techniques. *Pattern Recognition*, 31(8):983–1001, 1998.
23. B. T. Messmer and H. Bunke. A New Algorithm for Error-Tolerant Subgraph Isomorphism Detection. *IEEE Transactions on PAMI*, 20(5):493–504, May 1998.
24. H. Murase and T. Wakahara. Online Hand-Sketched Figure Recognition. *Pattern Recognition*, 19(2):147–160, 1986.
25. A. Okazaki, T. Kondo, K. Mori, S. Tsunekawa, and E. Kawamoto. An Automatic Circuit Diagram Reader with Loop-Structure-Based Symbol Recognition. *IEEE Transactions on PAMI*, 10(3):331–341, 1988.
26. B. G. Park, K. M. Lee, S. U. Lee, and J. H. Lee. Recognition of partially occluded objects using probabilistic ARG (attributed relational graph)-based matching. *Computer Vision and Image Understanding*, 90:217–241, 2003.
27. B. Pasternak. The Role of Taxonomy in Drawing Interpretation. In *Proceedings of 3rd International Conference on Document Analysis and Recognition, Montréal (Canada)*, pages 799–802, August 1995.
28. B. Pasternak. *Adaptierbares Kernsystem zur Interpretation von Zeichnungen*. Dissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.), Universität Hamburg, April 1996.
29. J. L. Pfaltz and A. Rosenfeld. Web Grammars. In *Proceedings of 1st International Joint Conference on Artificial Intelligence*, pages 609–619, 1969.
30. J. Rendek, B. Lamiroy, and K. Tombre. A Few Step Towards On-the-Fly Symbol Recognition with Relevance Feedback. In H. Bunke and A. L. Spitz, editors, *Document Analysis Systems VII: Proceedings of 7th International Workshop on Document Analysis Systems, Nelson (New Zealand)*, volume 3872 of *Lecture Notes in Computer Science*, pages 604–615, February 2006.
31. J. W. Roach and J. E. Tatem. Using Domain Knowledge in Low-level Visual Processing to Interpret Handwritten Music: An Experiment. *Pattern Recognition*, 21(1):33–44, 1988.
32. A. Rosenfeld. Array, Tree and Graph Grammars. In H. Bunke and A. Sanfeliu, editors, *Syntactic and Structural Pattern Recognition: Theory and Applications*, chapter 4, pages 85–115. World Scientific, 1990.
33. J.-P. Salmon, L. Wendling, and S. Tabbone. Automatical Definition of Measures from the Combination of Shape Descriptors. In *Proceedings of 8th International Conference on Document Analysis and Recognition, Seoul (Korea)*, September 2005.
34. L. G. Shapiro and R. Haralick. Structural Description and Inexact Matching. *IEEE Transactions on PAMI*, 3(5):504–519, 1981.
35. J. Song, F. Su, C.-L. Tai, and S. Cai. An Object-Oriented Progressive-Simplification Based Vectorization System for Engineering Drawings: Model, Algorithm, and Performance. *IEEE Transactions on PAMI*, 24(8):1048–1060, August 2002.
36. S. Tabbone and L. Wendling. Technical Symbols Recognition Using the Two-dimensional Radon Transform. In *Proceedings of the 16th International Conference on Pattern Recognition, Qubec (Canada)*, volume 2, pages 200–203, August 2002.

37. S. Tabbone and L. Wendling. Binary shape normalization using the Radon transform. In *Proceedings of 11th International Conference on Discrete Geometry for Computer Imagery, Naples (Italy)*, volume 2886 of *Lecture Notes in Computer Science*, pages 184–193, November 2003.
38. S. Tabbone, L. Wendling, and K. Tombre. Matching of Graphical Symbols in Line-Drawing Images Using Angular Signature Information. *International Journal on Document Analysis and Recognition*, 6(2):115–125, 2003.
39. S. Tabbone, L. Wendling, and D. Zuwala. A Hybrid Approach to Detect Graphical Symbols in Documents. In *Proceedings of the 6th IAPR International Workshop on Document Analysis Systems, Florence, (Italy)*, volume 3163 of *Lecture Notes in Computer Science*, pages 342–353, September 2004.
40. E. Valveny and Ph. Dosch. Performance Evaluation of Symbol Recognition. In *Proceedings of the 6th IAPR International Workshop on Document Analysis Systems, Florence, (Italy)*, volume 3163 of *Lecture Notes in Computer Science*, pages 354–365, September 2004.
41. E. Valveny and Ph. Dosch. Symbol recognition contest: a synthesis. In J. Lladós and Y. B. Kwon, editors, *Graphics Recognition: Recent Advances and Perspectives – Selected papers from GREC'03*, volume 3088 of *Lecture Notes in Computer Science*, pages 368–385. Springer-Verlag, 2004.
42. E. Valveny and E. Martí. Deformable Template Matching within a Bayesian Framework for Hand-Written Graphic Symbol Recognition. In A. K. Chhabra and D. Dori, editors, *Graphics Recognition—Recent Advances*, volume 1941 of *Lecture Notes in Computer Science*, pages 193–208. Springer-Verlag, 2000.
43. L. Wendling and S. Tabbone. A New Way to Detect Arrows in Line Drawings. *IEEE Transactions on PAMI*, 26(7):935–941, July 2004.
44. R. C. Wilson and E. R. Hancock. Structural Matching by Discrete Relaxation. *IEEE Transactions on PAMI*, 19(6):634–648, June 1997.
45. J. Wood. Invariant Pattern Recognition: A Review. *Pattern Recognition*, 29(1): 1–17, 1996.
46. X. Xiaogang, S. Zhengxing, P. Binbin, J. Xiangyu, and L. Wenyin. An online composite graphics recognition approach based on matching of spatial relation graphs. *International Journal on Document Analysis and Recognition*, 7(1):44–55, September 2004.
47. S. Yang. Symbol Recognition via Statistical Integration of Pixel-Level Constraint Histograms: A New Descriptor. *IEEE Transactions on PAMI*, 27(2):278–281, February 2005.
48. D. Zuwala and S. Tabbone. A Method for Symbol Spotting in Graphical Documents. In H. Bunke and A. L. Spitz, editors, *Document Analysis Systems VII: Proceedings of 7th International Workshop on Document Analysis Systems, Nelson (New Zealand)*, volume 3872 of *Lecture Notes in Computer Science*, pages 518–528, February 2006.

Symbol Spotting in Technical Drawings Using Vectorial Signatures*

Marçal Rusiñol and Josep Lladós

Centre de Visió per Computador / Computer Science Department
Edifici O, Campus UAB 08193 Bellaterra (Cerdanyola), Barcelona, Spain
{marcal, josep}@cvc.uab.es
http://www.cvc.uab.es

Abstract. In this paper we present a method to determine which symbols are probable to be found in technical drawings using vectorial signatures. These signatures are formulated in terms of geometric and structural constraints between segments, as parallelisms, straight angles, etc. After representing vectorized line drawings with attributed graphs, our approach works with a multi-scale representation of these graphs, retrieving the features that are expressive enough to create the signature. Since the proposed method integrates a distortion model, it can be used either with scanned and then vectorized drawings or with hand-drawn sketches.

1 Introduction

Symbol recognition is one of the major research activities in the field of Graphics Recognition. It has a number of applications to the analysis of technical drawings and maps at large. Examples are the interpretation of scanned drawings for validation or retroconversion, or the iconic indexing in a document image database. In the problem of retrieving images from a database by their content, applications use to formulate queries in terms of textual information as latitude coordinates, street names, etc. or graphical information as the situation of interesting elements as roads, rivers, airports, etc. as explained in [1]. On the other hand, iconic indexing allows to retrieve images from a large database by querying single elements of the drawing. Usually, architects or engineers have a great amount of technical drawings and they re-use data from previous projects for their new designs. Nowadays locating these elements requires visual examination of each document and it is a tedious task. Iconic indexing is suitable to provide solutions to this kind of problems. Often it is more natural and effective, instead of making a textual query to use a sketching interface where the symbol to spot is not stored in a database but drawn in an on-line process by the user.

In this paper we present a method to discriminate symbols in technical drawings for an indexing purpose. Effective symbol recognition methods exist in the

* This work has been partially supported by the Spanish project CICYT TIC 2003-09291 and the Catalan project CeRTAP PVPC.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 35–46, 2006.
© Springer-Verlag Berlin Heidelberg 2006

literature [2]. Structural, usually graph matching, statistical or hybrid methods may be found. However, most of the approaches require a segmentation of the symbol to achieve a classification with a high performance. When dealing with large documents, a paradox appears: to segment for recognizing or to recognize for segmenting. Graph matching or other recognition schemes are too much complex to tackle with segmentation and recognition simultaneously. So, a method to determine which symbols are likely to be found in the drawing is suitable as a pre-processing step to recognition techniques. A compact representation of features —signature— can provide the accuracy and the speed desirable in such cases. A comprehensive review on description techniques for shape representation can be found in [3].

There are two different signature paradigms for symbol description depending on whether it is represented, with pixel based features or with a vectorial representation. Some techniques like those described in [4] use bitmap images to find a signature, named force signature, based on angular information to describe each symbol. The method can handle degradation of the objects to recognize, because force signatures are robust to noise. Also, the method can discriminate very similar objects, but user interaction is needed to correct object segmentation. In [5] constraints as length-ratios or angles between two pixels in reference to a third pixel are accumulated in a histogram used as a symbol descriptor. This method gives high performance under diverse drawbacks as degradation, rotation, scaling. However it also needs a pre-segmentation of the symbols and its complexity is very high ($\mathcal{O}(n^3)$) because every triplet of pixels is considered. Since technical drawings are highly structured and composed by simple geometric sub-shapes, it seems more suitable to work with a vectorial representation rather than with pixel based images. In [6] the raster images are approximated by polygons to obtain boundary segments and to extract constraints from this representation as distance, angle, directions, etc. among segments. An indexing scheme is built and the method gives good results locating and recognizing the objects. However, the method can not handle images with noisy edge data. In [7], the image is completely vectorized and the segments are combined to form a set of tokens, as chains of adjacent lines, or even textual features. The frequency of an indexing feature and the document frequency of the indexing features are taken into account to extract the similarity between a document and the query. We have based our work on the method proposed by Dosch and Lladós [8], where the vectorial signatures are built from the occurrences of constraints between segments, as parallelisms, straight angles, overlap-ratios, etc. But, these signatures are calculated only for small rectangular parts of the image. This fixed bucket partition provokes a lack of flexibility, there is no invariance to scale. It can also cause some problems if the symbol to discriminate is not completely inside of one of these windows.

Our work is targeted to vectorized drawings obtained either from scanned printed pages or from hand-drawn sketches. The operation of feature extraction when we are working with a vectorial representation can provide more speed than when we face up to a pixel based approach. The drawback of vectorial data

is that a distorted input or even a different set of vectorization parameters may result in a sensitive variation in the obtained segments. Because of that, the spotting technique must be error tolerant and invariant to rotation, scale and translation. These constraints, which are a problem in bitmap images, are easily solved in vectorial representation if the features taken into account are extracted using comparisons between different segments (ratios or angles).

The features are organized in a signature and then matched against a database of signature models in order to index the different regions of interest of the drawing, as explained in [9]. This kind of indexing-based approaches, together with vectorial signatures, do not look for an exact match with the model symbols but to determine which symbols are probable to be found in these regions by a fast technique. Afterwards, the recognition step can be focused in each region and the well-known recognition techniques can take advantage of the extracted knowledge of the spotting process.

Vectorial signatures are suitable for this kind of problem since the drawings are formed by structured sub-shapes, so, the extracted constraints are focused in geometric-invariant relationships between segments, as parallelisms, straight-angles, segment junctions, length-ratios, distance-ratios, etc. The determination of the spatial relationship between a pair of segments is explained in [10]. These simple constraints can be observed and organized in a hierarchical way in order to represent some expressive structural relations between different segments. This means that it is more representative to find a square than two parallelisms and four straight angles. Our method tries to find these sub-shapes to create the vectorial signature.

The remainder of this paper is organized as follows. Section 2 describes how the vectorial signatures are built and learned. In section 3, we present the construction of the regions of interest which allows to apply the signatures in real drawings. In Section 4, the experimental results are explained and finally, the conclusions are discussed in section 5.

2 Building the Vectorial Signatures

Given a line drawing image, it is first vectorized and represented with an attributed graph. Graph nodes represent segments and graph edges represent structural relationships among segments. Formally, a graph G is defined as follows:

Definition 1. *A* graph *is denoted as* $G = (V, E)$ *where* V *is the set of nodes representing the segments and* E *is the set of edges representing the relationship between them. A* subgraph *of* G *containing the nodes* $s_i, ..., s_j$ *will be denoted as* $G_{\{s_i,...,s_j\}}$. G *is a complete graph.*

Definition 2. *An* attributed graph *is denoted as* $G = (V, E, \mu, \nu)$ *where* Σ_V *and* Σ_E *are a set of* symbolic labels*, and the functions* $\mu : V \rightarrow \Sigma_V$ *and* $\nu : E \rightarrow \Sigma_E$ *assign a label to each node and each edge.* $\Sigma_V = [\theta_{s_i}, \rho_{s_i}]$ *will contain the information of each segment* s_i *according to a polar representation.* $\Sigma_E = \{L, T, P, 1, 0\}$ *will represent the different kind of relationship between each pair of segments. The possible relationships between segments are:*

1. *L represents a straight angle between a pair of adjacent segments.*
2. *T represents a straight angle between a pair of non-adjacent segments.*
3. *P represents two parallel segments.*
4. 1 *represents two adjacent segments.*
5. 0 *represents a non expressive relation between two segments.*

Definition 3. *M_G is the* adjacency matrix *of a graph G and $M_{G\{s_i,\ldots,s_j\}}$ the adjacency matrix of the subgraph $G_{\{s_i,\ldots,s_j\}}$.*

We define a signature in terms of a hierarchical decomposition of symbols. Thus, it counts the occurrences of primitive shapes that are expressive enough in line drawings as squares, triangles, parallelisms, etc. Since the number of vectors of these shapes are not the same for all of them, we work with a combinatorial approach on the number of graph nodes. In Fig. 1, we have got a graph[1] of a simple symbol, we can see the symbol could be decomposed in a square and in a triangle, the subgraph $G_{\{s_2,s_3,s_4\}}$ represents the triangle and the subgraph $G_{\{s_1,s_4,s_5,s_6\}}$ the square.

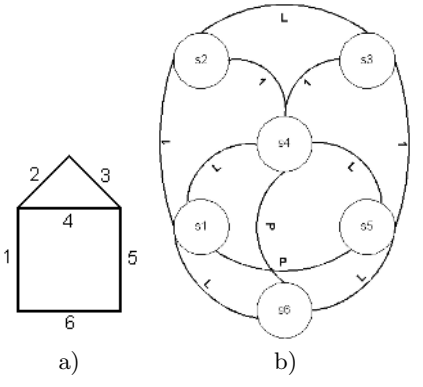


Fig. 1. (a) A simple symbol. (b) Its attributed graph.

For all the segments, all the subgraphs formed by at least two nodes, and a maximum of four nodes are analyzed to search some representative shapes. The equation 1, being n the number of segments, gives the number of subgraphs to analyze.

$$\#G_{\{\ldots\}} = \sum_{k=2}^{4} C_n^k = \sum_{k=2}^{4} \frac{n!}{(n-k)! \times k!} \tag{1}$$

For each subgraph, we work with its adjacency matrix. The matrix M_G is in fact only computed once for all the segments, and then, when we want to focus to a subgraph, a group of rows and columns of this matrix will be selected. Notice that in most cases the relations between segments could be extracted in the

[1] The edges labelled with a 0 are not shown.

vectorization process. In the extraction of these constraints, each comparison has associated a threshold value in order to be more tolerant. For the simple shape of Fig. 1, we can see below (2) its corresponding adjacency matrix M_G.

$$M_G = \begin{pmatrix} s_1 & 1 & 0 & L & P & L \\ 1 & s_2 & L & 1 & 0 & 0 \\ 0 & L & s_3 & 1 & 1 & 0 \\ L & 1 & 1 & s_4 & L & P \\ P & 0 & 1 & L & s_5 & L \\ L & 0 & 0 & P & L & s_6 \end{pmatrix} \tag{2}$$

From this matrix, we examine all the possible combinations of matrices taking four, three and two of the six possible nodes. Hence three different levels are considered. Below, in (3) we can see the resulting sub-matrices considering only the triangle and the square of the shape of Fig. 1.

$$M_{G_{\{s_2,s_3,s_4\}}} = \begin{pmatrix} s_2 & L & 1 \\ L & s_3 & 1 \\ 1 & 1 & s_4 \end{pmatrix} \textit{and}\, M_{G_{\{s_1,s_4,s_5,s_6\}}} = \begin{pmatrix} s_1 & L & P & L \\ L & s_4 & L & P \\ P & L & s_5 & L \\ L & P & L & s_6 \end{pmatrix} \tag{3}$$

For all the sub-matrices representing the subgraphs, normally the analysis of one single row can determine the shape that it encodes.

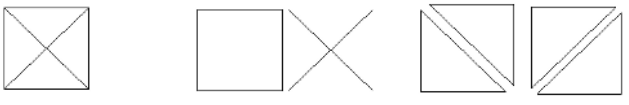
Definition 4. *Let us denote $d(M_{G_{\{s_i,\ldots,s_j\}}})$ a single row of an adjacency matrix. It will be used as a* descriptor *of the expressive sub-shapes. Every descriptor d has associated an equivalence class $L : \{d_1, ..., d_n\}$ of its synonyms.*

Definition 5. *Let S be a vector where in each position we have the number of occurrences of each descriptor $d(M_{G_{\{s_i,\ldots,s_j\}}})$ representing an expressive sub-shape. S is defined as the* vectorial signature *of the analyzed shape.*

A descriptor of a reference segment having two straight angles and a parallelism represent a square $d = [s_iLLP]$. As the information inside the matrix depends on the order of definition of the segments when the vectorization step is done, we must have a dictionary of its synonyms $L : \{d_1, ..., d_n\}$. For instance, a segment having a descriptor $d = [s_iLLL]$ is the same that another segment having a descriptor $d = [s_iPLP]$ because if a segment is perpendicular to three other segments, one of them is parallel to two of them and perpendicular to the last one. We can see some example of synonyms list and some of the expressive shapes taken into account to perform the voting scheme to build the signature in Table 1. It seems redundant to store the information for multiple levels of the subgraph, if in the level of four nodes we find a square, it is obvious we will find two parallelisms and four straight angles in the level of two nodes. But this redundancy helps to detach completely all the multiple shapes in the drawing. For instance, a square with a cross inside can be seen as a square and a straight angle, or it can be seen as a set of triangles (see example in Fig. 2). This redundancy helps to be more error tolerant and to store all the structural information

Table 1. Some possible rows of the sub-matrix, with its synonyms and its representations

Level	Synonyms list	Image representation of the sub-shapes
4 nodes	$[[s_iPPP]]$	
4 nodes	$[[s_iPPL],[s_iPLP],[s_iLPP],[s_iLLL]]$	
4 nodes	$[[s_iPLL],[s_iLPL],[s_iLLP]]$	
3 nodes	$[[s_iPP]]$	
3 nodes	$[[s_iPL],[s_iLP],[s_iLL]]$	
3 nodes	$[[s_i11]]$	
2 nodes	$[[s_iP]]$	
2 nodes	$[[s_iL]]$	
2 nodes	$[[s_i1]]$	



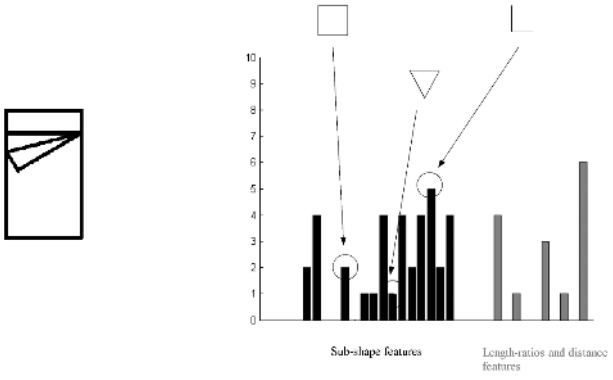
a) b) c)

Fig. 2. (a) Original symbol. (b) Symbol detached at level four and at level two. (c) Symbol detached at level three.

of all the multiples sub-shapes of the drawing. The occurrences of each sub-shape will vote to build the vectorial signature. To have more information, we can add some additional information as the length-ratio and the distance-ratio at the end of the signature. These measure features can take values from 0 to 1; this space is the split into five bins where the votes are accumulated. We can see an example of a signature of a symbol in Fig. 3.

The learning process of the signatures has been made using a selection of the symbol database used in the *GREC 2003* symbol recognition contest [11], containing symbols of both architectural and electronic fields. The symbols with arcs are not taken into account, except the doors which are approximated by a polyline. The signature of the twenty-seven symbols has been calculated, and a mean of signatures has been made with a set of at least ten distorted representations of each symbol. With this symbol database, there is no need to use higher order relationships to discriminate the symbols between them.

Once the database of signatures is constructed, we can compare the obtained signature with this database and associate a probability to each correspondence between the original symbol and all the symbols of the database following a distance function. This comparison process could be done in a hierarchical way,



a) b)

Fig. 3. (a) A symbol of a bed. (b) A graphical representation of its signature divided in the sub-shapes features and the measure features.

in order to discard some non possible solutions, or simply associating weights to each feature of the signature to classify the most relevant features. The used distance function is a simple Euclidean distance which gives acceptable results.

3 Regions of Interest

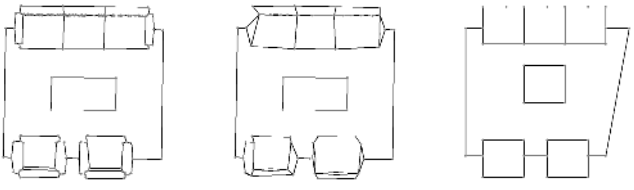
When we work with complete drawings we need to divide the drawing into separate windows framing every symbol. In each zone of interest a voting scheme of structural relations is used. The idea is that every structural relation found in this zone will contribute to form the signature to be compared with the models. Every framing window has its own voting scheme and the regions where the votes reach their maximum will be selected as candidates to contain the queried symbol using a voting approach inspired by the idea of the GHT [12]. These regions are dynamic since they are built depending on the original line drawing, this approach works much better than other segmentation techniques used in technical drawings which only divide the drawing in a fixed bucket partition which loses flexibility.

These regions of interest are computed from the maximum and minimum coordinates of several adjacent segments. So, the size of the regions of interest is variable. Also, a first filter of area and aspect-ratio can be easily implemented in order to delete some non relevant symbols as for example the walls in the architectural field or the wiring connections in electronic diagrams.

For each node n_{s_i} of G (segment in the drawing) we build a list of all the nodes connected to n_{s_i} by an edge. We have a list of all the endpoints of the adjacent segments to reference segment. In this list, we get the maximum and minimum coordinates of the endpoints that will construct a framing window of these segments. As in most cases of technical drawings the symbols have a low eccentricity, its bounding-box are square-shaped and this kind of windows frame

them. But, as the windows are based on the connection of the segments, the efficiency decrease if the symbols are disconnected or overlapped.

But, in the vectorization step, more problems may happen: small vectors can appear due to noise, straight lines can be split into several collinear vectors, the arcs are approximated by polylines, some neighboring lines in the drawings are not adjacent in the vectorial representation because of gaps, dashed lines appear as a set of small segments instead of one unique instance, etc. To solve this kind of problems, the best results are reached when we work with a lower resolution of the drawing to calculate the windows. This sub-sampling step reduces local distortions in the vectorial representation but preserving the most salient geometrical properties.



a) b) c)

Fig. 4. (a) Original drawing. (b) Graph contraction by distance. (c) Low resolution representation.

First, a contraction of the normalized graph is done, merging the adjacent nodes having a lower distance than a threshold thr. Then, applying the equation 4 to each node coordinate we get a lower resolution graph. With this representation with decreased resolution, the problems of the gaps, or the split segments are solved. Every endpoint is sampled for each step of m, so the minor errors are corrected.

$$\begin{aligned} x &= m \times round(x/m) \\ y &= m \times round(y/m) \end{aligned} \tag{4}$$

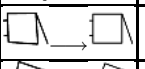
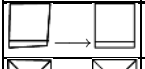
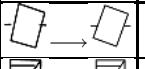
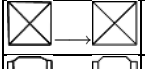
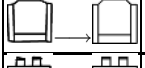
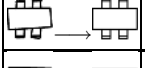
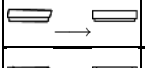
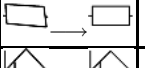
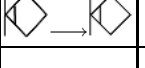
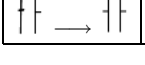
Experimentally, in Fig. 4(a) the graph has 154 nodes because an horizontal line has been split in the vectorization process. When the graph contraction by distance is done (with a threshold value $thr = 0.06$) Fig. 4(b), we get a graph with 52 nodes which lines are crooked due to the node contraction, and with the decreased resolution graph (with $m = 35$) Fig. 4(c) we have to face up to only 33 nodes.

This change of resolution can cause some other errors, for example some lines which are almost horizontal or vertical can be represented with a very different slope. But these errors do not interfere with the obtained windows, since they continue to frame the symbols. Notice that these lowest resolution images will only be used to calculate the regions of interest, not for the spotting process. Since each segment proposes a region of interest, there is no problem if one of the segments of a symbol gives a mistaken window.

4 Experimental Results

Our experimental framework consisted of two scenarios. First we have tested the performance of our approach to classify isolated symbols. Secondly, we have used the method for symbol spotting in real architectural drawings.

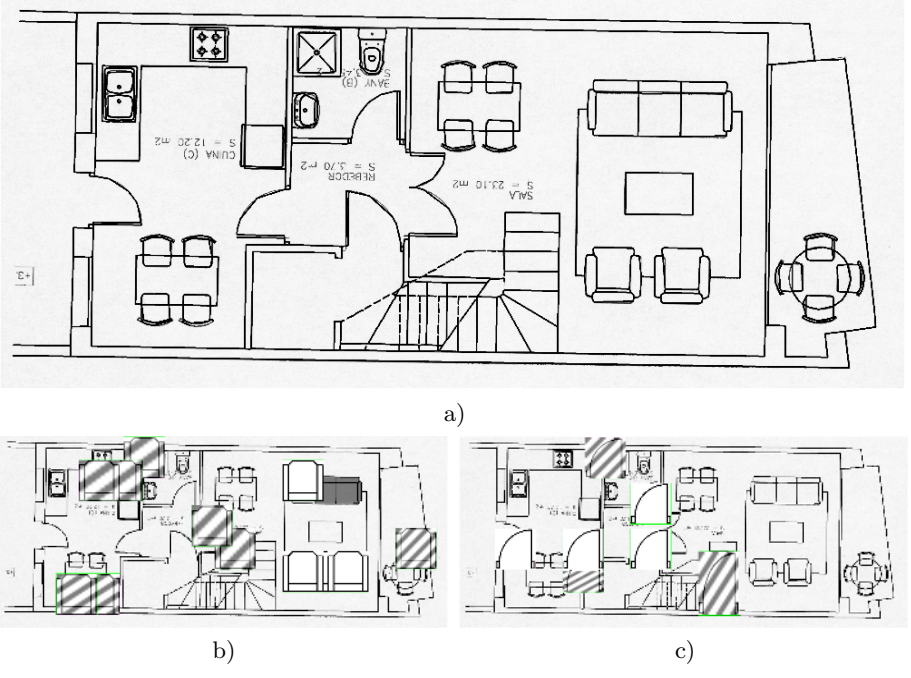
Table 2. Results of the recognition

Symbol	Recognition rate	Symbol	Recognition rate
	36/38		19/19
	39/39		19/20
	15/15		15/15
	27/27		16/16
	15/15		20/20
	7/8		20/20
	36/38		14/14
	11/11	Total	309/315

The first tests were done using the *GREC 2003* database. This database contains, in addition of the models, some synthetical distorted symbols, rotated symbols and symbols at different scales. Working with fifteen different classes of symbols, a set with 315 examples of different levels of distortion has been tested and we achieved a 98% of recognition. We can see the detailed results in Table 2. With 230 examples of rotated and scaled symbols we achieved the 100% of recognition.

In the second test, we tried out the vectorial signatures with real architectural drawings. Allowing a higher error than when we are working with the database, the symbols can be spotted, and they are usually well discriminated. As we can see in Fig. 5, some false positives appear (dashed zones) and one sofa is not spotted (grey zone). False positives appear when a window does not correctly frame a symbol, or when a symbol (like the shower) is not in our signature database. The stairs which consist of a lot of segments give a lot of regions of interest where false positives appear, and the wrong segmentation of the tables makes that the part where the chairs are drawn a sofa is spotted, because their representation is very close. When we use vectorial signatures in real drawings there are two factors that cause the spotting results not to be so good. First of all, the symbols can be adjacent between them or to a wall, or the region of interest could not frame perfectly the symbol, in this case we face up to occlusions and additions of segments. On the other hand, in real drawings, the symbol design

may be different of the learned model, so the learned features of a symbol could not appear in real drawings, in this case it is obvious the symbol can not be spotted, a semantic organization of different design instances for any symbol is necessary.



a)

b) c)

Fig. 5. (a) Original image. (b) Locations with high probability to find a sofa. (c) Locations with high probability to find a single door.

Finally we tried the method with an application of image database retrieval by sketches. A sketch of a symbol is drawn and then vectorized. Its signature is computed and the locations of a drawing with high probability to contain the queried symbol are spotted. We can see the results in Fig. 6. With this kind of applications we have to face up to two important drawbacks, first of all, the signature of the sketch is not the same of the models, because in the vectorization process a lot of small single segments appear. Secondly, as we said before, the symbol design may change from a drawing to another. That is why in this test we must allow a higher error in the spotting process and the constraints to obtain the regions of interest must be much more restrictive, otherwise a lot of false positives appear. This kind of applications, give much better results when they are used in an on-line process in order to have the users feedback when the sketch is drawn to correct the representation errors.

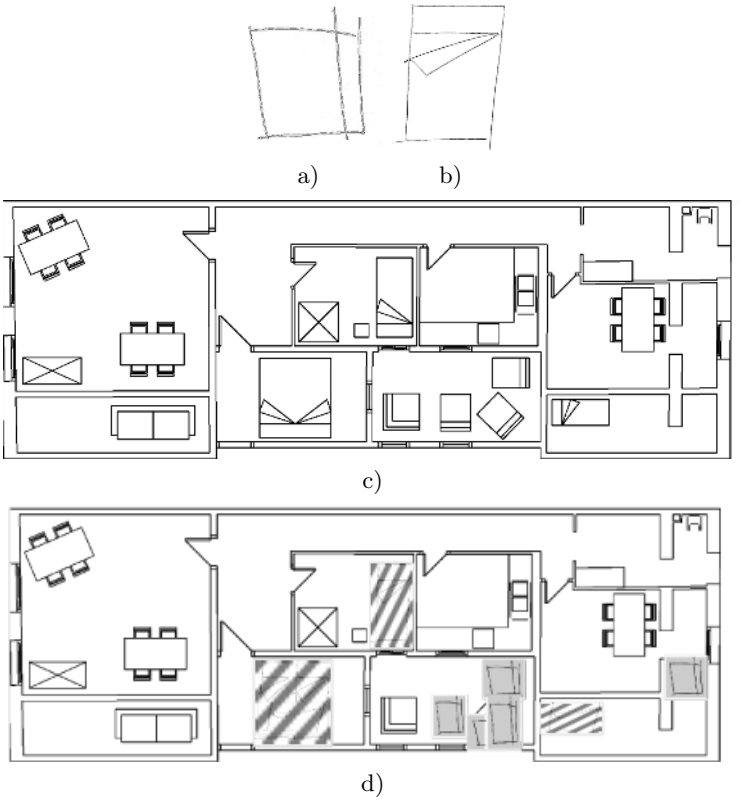


Fig. 6. (a) Sketch of a sofa. (b) Sketch of a bed. (c) Original drawing. (d) Locations with high probability to find a sofa (grey zones) and a bed (dashed zones).

5 Conclusions and Discussion

In this paper we have presented a method to detect the regions of a technical drawing where a symbol is probable to be found. This method is suitable for applications of iconic indexing and retrieval. Our method, starting from a vectorial representation of the line drawing, builds a vectorial signature in each region of interest using a voting scheme. These signatures are formed by the occurrences of some sub-shapes which can appear in line drawings that are expressive enough, and by some additional information as length-ratios, distance-ratios, etc. These signatures are then matched with the database of learned signatures of the models to determine which symbol is probable to be present.

We can see that the symbol discrimination using vectorial signatures gives good results when we are working with the database and with symbols with synthetical distortion, which is a controlled framework. In real scanned architectural drawings, the segmentation and discrimination of symbols are done simultaneously, even if the symbols are usually well spotted, a lot of false positives appear.

In the image database retrieval by sketches test we get acceptable results. As the objective of this technique is not to give a recognition of the symbol but in some way to index the drawing, the false positives problem is not so significant.

When working with vectorial data, one of the main drawbacks is the arc representation. In this paper we have approximated arcs with polylines, but it is necessary to add an arc segmentation algorithm in the vectorization process to consider the arcs and circles as expressive sub-shapes. Moreover the hierarchical organization of the signatures is essential to achieve a fast and accurate indexation of line drawings. Some features are more discriminative than others, and the presence or absence of a feature can cluster the candidate symbols database.

References

1. R.C. Thompson and R. Brooks. Exploiting perceptual grouping for map analysis, understanding and generalization: The case of road and river networks. In *Graphics Recognition, Algorithms and Applications*, pages 148–157. 2002.
2. J. Lladós, E. Valveny, G. Sánchez, and E. Martí. Symbol recognition: Current advances and perspectives. In *Graphics Recognition, Algorithms and Applications*, pages 104–127. 2002.
3. D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.
4. S. Tabbone, L. Wendling, and K. Tombre. Matching of graphical symbols in line-drawing images using angular signature information. *International Journal on Document Analysis and Recognition*, 6(2):115–125, 2003.
5. S. Yang. Symbol recognition via statistical integration of pixel-level constraint histograms: A new descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):278–281, 2005.
6. F. Stein and G Medioni. Structural indexing: Efficient 2-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(12):1198–1204, 1992.
7. O. Lorenz and G. Monagan. A retrieval system for graphical documents. In *Symposium on Document Analysis and Information Retrieval*, pages 291–300, 1995.
8. P. Dosch and J. Lladós. Vectorial signatures for symbol discrimination. In *Graphics Recognition: Recent Advances and Perspectives*, pages 154–165. 2004.
9. H. Wolfson and I. Rigoutsos. Geometric hashing: an overview. *IEEE Computational Science and Engineering*, 4(4):10–21, 1997.
10. A. Etemadi, J.P. Schmidt, G. Matas, J. Illingworth, and J. Kittler. Low-level grouping of straight line segments. In *Proceedings of the British Machine Vision Conference*, pages 118–126, 1991.
11. E. Valveny and P. Dosch. Symbol recognition contest: A synthesis. In *Graphics Recognition: Recent Advances and Perspectives*, pages 368–386. 2004.
12. D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

A Generic Description of the Concept Lattices' Classifier: Application to Symbol Recognition

Stéphanie Guillas, Karell Bertet, and Jean-Marc Ogier

L3I, Université de La Rochelle, av M. Crépeau, 17042 La Rochelle Cedex 1, France
{sguillas, kbertet, jmogier}@univ-lr.fr

Abstract. In this paper, we present the problem of noisy images recognition and in particular the stage of primitives selection in a classification process. We suppose that segmentation and statistical features extraction on documentary images are realized. We describe precisely the use of concept lattice and compare it with a decision tree in a recognition process. From the experimental results, it appears that concept lattice is more adapted to the context of noisy images.

1 Introduction

The work presented in this paper tackles the problem of the automatic reengineering of documents, and proposes a first theoretical approach concerning the use of concept lattices for automatic recognition of graphic objects, under the multi-scale and multi-orientation constraints.

In the field of invariant pattern recognition, there is a consensus about the fact that each stage of the recognition process is important [1, 2]. Furthermore, the review of the literature highlights several difficulties that existing techniques try to tackle more or less partially.

The first difficulty is the adaptation to the notion of context, aimed at trying to find some adequate recognition scenarios to a particular problem, if possible by integrating the capacity of evolution of the system. Another difficulty is related to the problem of combination of recognition schemas, by integrating structural and statistical description of the shapes, without any previous distortion of the recognition schema. At last, the problem concerning the selection of relevant primitives, adapted to a particular context, in adequation with evolutive systems stays an open problem, and is not made explicit. Literature is rich in terms of classification strategy. A lot of references indicate these problems, depending on different techniques : statistical [3] and/or structural approaches [4], parametric and non parametric approaches [5], connexionnist [6], training problems, primitives selection or fusion of classifiers problems [7], ...

In this paper, we propose a first contribution concerning symbols recognition based on concept lattices. Indeed, lattices seem to bring some interesting answers to the previously discussed difficulties, thanks to their natural ability to integrate statistical and structural description, and to their capacity to validate some relevant primitives in regard with a particular context. Moreover, the concept

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 47–60, 2006.
© Springer-Verlag Berlin Heidelberg 2006

lattice presents the advantage of a good readability and thus to be easy to understand. Shape recognition is classically realized in two stages : *learning* on symbols images which is the subject of the next part and *classification* of damaged symbols images presented in section 3.

2 Learning

The learning stage consists in organizing the information extracted from a set of objects by a concept lattice. In our case objects are images of symbols. These symbols are described by same-size numerical signature computed thanks to image processing techniques. Previously, it is necessary to have normalized data so that their representation is equivalent. More precisely, the learning stage can be described by:

Name: Learning
In: a *set of objects* O where each object $p \in O$ is a symbol described by a normalized signature $p = (p_1, \dots p_n)$ and a label of class $c(p)$.
Out: a *concept lattice* $(\beta(C), \leq)$ described by a set of concepts $\beta(C)$ and a relation $\leq$ between its concepts.

The learning involves two stages as shown in Figure 1:

- the *discretization* of signatures: the data are assigned to disjoined intervals. It is possible to find again the initial data by the union of these intervals. Discretization is essential to build the concept lattice. It is parameterized by a *cutting criterion* necessary to the construction of the intervals.
- the *building of concept lattice* from discretized data. This stage does not need any parameter.

2.1 Discretization

The discretization stage consists in organizing the numerical data of the different objects in discrete intervals to obtain a specifical characterization of each class of objects.

Name: Discretization
In: a *set of objects* O where each object (symbol) $p \in O$ is described by a signature which is a *numerical vector* $p = (p_1, \dots p_n)$ where each value is normalized and a *label of class* $c(p)$.
Out:
- the *intervals* organized in sets of intervals $I = I_1 \times I_2 \times \dots \times I_m$ where the intervals of each set I_i are disjoined, and cover the values p_i of the whole objects $p \in O$.
- a *membership relation* R which is defined for each object $p \in O$ and each interval $x \in I$ by: $pRx \Leftrightarrow$ it exists $i = 1 \dots m$ such as $p_i \in x \in I_i$

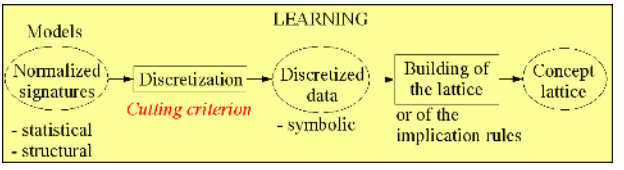


Fig. 1. Schematic description of learning

Description. The discretization is realized on the signatures organized in a table of data (fig. 1). At the beginning, we build for each feature i an interval $x \in I_i$ by gathering the whole values p_i taken by the symbols $p \in O$. Thus, we can initialize the membership relation R which is deduced. After this initialization stage, each set I_i contains one interval, and each symbol $p \in O$ is in relation with each interval $x \in I_i$. Then, we have to select an interval x to cut, and a cutting point in this interval x. To do that, we introduce the following notations:

- For a symbol $p \in O$, we define the set I_p of the intervals in membership relation with p: $I_p = \{x \in I \text{ such as } pRx\}$.
- For an interval $x \in I$, we define the set V_x of the numerical values of the symbols in which it is in relation and sorted by ascending order : $V_x = (p_i \text{ such as } p_i \in x)$ sorted by ascending order so : $V_x = v_1 \leq v_2 \leq \ldots \leq v_n$

Thus we have to select an interval $x \in I$ among the whole set of intervals, and a value $v_j \in V_x$ among the wholes values, and then to cut the interval x in two intervals x' and x'' with $V'_x = v_1 \leq \ldots \leq v_j$ and $V_{x''} = v_{j+1} \leq \ldots \leq v_n$. Each symbol will have a membership relation with one of these two created intervals, that enables to differentiate the two subsets of formed symbols. We repeat this process of cutting the intervals until we can distinguish each class. The selection of interval to cut depends on a *cutting criterion* that have to be defined.

When each class can be characterized by an own set of intervals, we obtain a discretized table involving the whole symbols $p \in O$ and the whole intervals $I = I_1 \times I_2 \times \ldots \times I_m$ where I_i is the set of intervals obtained for each feature $i = 1 \ldots m$. Notice that if a feature k has never been selected to be discretized, it contains only one interval ($|I_k| = 1$) which is in relation with the whole symbols. This feature is not discriminative, and thus can be removed from the discretized table. From this table, it is possible to deduce the membership relation R, and consequently, for an symbol $p = (p_1, p_2, \ldots, p_m) \in O$ where p_i is the value for the feature $i = 1 \ldots m$, to know the set I_p of intervals associated to p.

Example 1. *Table 1 (left) shows normalized data of 10 symbols distributed in 4 classes. The signature characterizing each symbol is composed of 3 features (a, b and c). After discretization by the entropy criterion, we obtain Table 1 (right). Each feature has been selected and cut one time, they consequently are kept.*

Table 1. Signatures of the 10 symbols (left) and discretized table with the entropy criterion (right)

Class	*Ident.*	*Signature*		
		a [0-20]	*b* [0-20]	*c* [0-20]
1	1	1	4	15
	2	0	0	18
2	3	1	12	13
	4	0	16	15
	5	3	12	11
3	6	8	16	15
	7	6	20	20
	8	15	12	15
4	9	18	4	0
	10	20	12	2

Class	*Ident.*	*Intervals*					
		a1 [0-3]	*a2* [6-20]	*b1* [0-4]	*b2* [12-20]	*c1* [0-2]	*c2* [11-20]
1	1	X		X			X
	2	X		X			X
2	3	X			X		X
	4	X			X		X
	5	X			X		X
3	6		X		X		X
	7		X		X		X
	8		X		X		X
4	9		X	X		X	
	10		X		X	X	

Cutting Criterion. A large number of criteria allows to select the interval in order to divide and to determine the cutting point, and the choice of this parameter is decisive in the learning process. It is necessary to search an interval $x \in I$, with the values $V_x = (v_1 \dots v_n)$ sorted by ascending order, that *maximizes* a criterion, for a given value v_j. The interval will be cut between the values v_j and v_{j+1}. We can define a lot of cutting criteria depending or not on the data. Among these criteria, we mention the maximal distance, the entropy and the Hotelling's coefficient:

Maximal distance: $distance(v_j) = v_{j-1} - v_j$

Entropy: $gain_E(v_j) = E(V_x) - (\frac{j}{n}E(v_1 \dots v_j) + \frac{n-j}{n}E(v_{j+1} \dots v_n))$

with $E(V) = -\sum_{k=1}^{|c(V)|} \frac{n_k}{n} log_2(\frac{n_k}{n})$ the measure of entropy of an interval with n values where n_k is the number of symbols of the class k of the interval.

Hotelling's coefficient:

$gain_H(v_j) = H(V_x) - (\frac{j}{n}H(v_1 \dots v_j) + \frac{n-j}{n}H(v_{j+1} \dots v_n))$

with $H(V) = \frac{VarB(V)}{VarW(V)}$ the Hotelling's measure of an interval V of n values, with n_k the number of symbols of the class k, g_k the gravity center of the class k, g the gravity center of V, v_{k_i} the i-th element of the class k, $VarB(V) = \frac{1}{n}\sum_{k=1}^{|c(V)|} n_k(g_k - g)^2$ the measure of between class variance and $VarW(V) = \frac{1}{n}\sum_{k=1}^{|c(V)|} n_k(\sum_{i=1}^{n_k}(v_{k_i} - g_k)^2)$ the measure of within class variance.

Maximal distance method consists in searching the primitive which has the maximal gap between two consecutive values when values are in ascending order. Entropy function is a measure characterizing the degree of mixture of the classes. Hotelling's coefficient takes in consideration the maximization of the distance between classes and the minimization of the dispersion of each class.

Extensions. Note the possibility to integrate symbolic data with the numeric one. This integration consists in computing an extension of the membership relation R which can be done after discretization, to add symbolic data to the building of the lattice. This extension can also be done during the initialization of this relation

R, before the discretization. Thus it is useful to refine the cutting criterion. But it is only interesting when the cutting criterion takes into consideration the indication of class of the symbols, as the entropy criterion.

For some criteria, as the maximal distance one, we can cut the intervals after the stage that enables the characterization of each class. Indeed, instead of stopping when the table is discretized (for a number of discretization stages equals to t_1), we pursue the discretization n times to obtain more intervals for characterizing each class. The discretized table to t_n has got more intervals than the one to t_1, but it enables to refine the description of each class.

2.2 Construction of the Concept Lattice

After the discretization stage comes the building of the concept lattice. This stage is totally determined by the obtained membership relation R. There is no criterion or parameter to be considered for the construction of this graph because it represents the whole possible combinations of relation R between objects and intervals.

Name: Building of the concept lattice
In: a *membership relation* R between a set of objects O and a set of intervals I.
Out: a *concept lattice* $(\beta(C), \leq)$ described by a set of concepts $\beta(C)$ and a relation $\leq$ between its concepts.

Description. Concept lattice has first been studied from a theoretical point of view [8] before being developed in [9] to represent data in *formal concept analysis*. A concept lattice is defined from data organized by a discretized table. More formally, a *concept lattice* is defined as a set of *concepts* ordered by inclusion.

We associate to a set of symbols $A \subseteq O$, the set $f(A)$ of intervals in relation with the symbols of A: $f(A) = \bigcap_{p \in A} I_p = \{x \in I \mid pRx \ \forall \ p \in A\}$ Dually, we associate to a set of intervals $B \subseteq M$, a set $g(B)$ of symbols in relation with the intervals of B: $g(B) = \{p \in O \mid pRx \ \forall \ x \in B\}$ These two functions f and g defined between symbols and intervals establish a *connection of Galois*. Moreover, $g \circ f$ and $f \circ g$ verify the properties of a closure operator. We note $\varphi = g \circ f$ the closure on the set I.

A *formal concept* is a pair symbols-intervals in relation according to R. More formally, a formal concept is a pair (A, B) with $A \subseteq O$, $B \subseteq I$, $f(A) = B$ and $g(B) = A$. The *concept lattice* associated to the relation R is a pair $(\beta(C), \leq)$ where:

- $\beta(C)$ is the set of the whole concepts of C.
- $\leq$ is an order relation on $\beta(C)$ defined for two concepts of $\beta(C)$, (A_1, B_1) and (A_2, B_2) by: $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow \left\| \begin{array}{l} A_2 \subseteq A_1 \\ \text{(equivalent to } B_1 \subseteq B_2\text{)} \end{array} \right.$

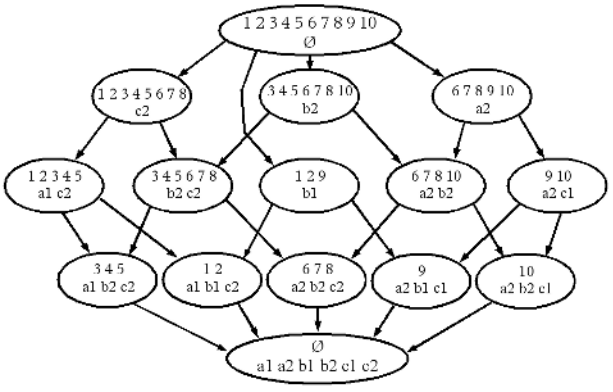


Fig. 2. Concept lattice

The relation $\leq$ is an order relation [1], thus it can be associated to a cover relation noted $\prec$. $(\beta(C), \prec)$ is then the Hasse diagram [2] of the concept lattice $(\beta(C), \leq)$.

The minimal concept to the sense of the relation R contains the whole symbols O. It is the concept $(O, f(O))$. The set $f(O)$, generally empty, corresponds to the intervals shared by the whole symbols. Dually, the maximal concept is $(g(I), I)$.

The representation of the concept lattice of the relation R is uniquely defined and the concepts corresponding to the relations symbols-intervals are ordered by inclusion. There are a lot of algorithms to generate the concept lattice : Bordat [10], Ganter [9], Godin et al. [11] and Nourine et Raynaud [12] which has the best theoretic complexity (quadratic complexity by elements of the produced lattice). To build the lattice, we have to set up the list of its whole concepts. The search of the concepts consists in finding in the discretized table the maximal rectangles, meaning the biggest sets of symbols and intervals in relation. After the generation of the whole concepts, it only remains to order them by inclusion (Figure 2).

Extension. The main limit of the use of concept lattice is its cost in time and space. Indeed, the size of the concept lattice is bounded by $2^{|S|}$ in the worst case, and by $|S|$ in the best case. Consequently the complexity is exponential in time and space in the worst case. It is very difficult to use studies of average complexity because the size of the lattice depends on the data. However notice that its size stays reasonable in practice as stated by the large number of experimentations which have been done.

To limit this exponential complexity, notice the possibility to generate only a *representation* of the lattice. An effective representation is defined by the fact that it is smaller, easily understandable, and that it determines the concept

[1] An order relation is a reflexive, symmetric and transitive relation.

[2] Representation of an order relation without its relations of reflexivity and transitivity.

lattice via efficient algorithms of generation. There are a large number of representations of a lattice proposed in the literature which verify these criteria (representation of a lattice by an order in lattice theory, by a table in formal concept analysis, by a conjunctive normal form in logic, by functional dependencies in databases). We mention the representation by a *system of implicational rules* [13, 14, 15] that we can find in data analysis. Such a representation enables, beyond its property of digest representation of the lattice, to avoid the complete generation of the lattice due to its possibility to use a *on line generation* of the only concepts which are necessary during the recognition stage. It also enables a description of the links between the features on another form, and highlights the links between features of type "The whole symbols which have the features x and y also have the feature z", that is formalized by the implicational rule $\{x, y\} \rightarrow z$ (or simply $x\,y \rightarrow z$).

3 Classification

After the learning stage and the generation of the concept lattice, it is the classification stage. The principle is to determine the class of new representations of the symbols, that is to say, to recognize the class of the symbols which can be more or less noised as shown in Figure 3.

Name: Classification
In:
- the signature $s = (s_1 \ldots s_n)$ of the symbol s to class
- the concept lattice $(\beta(C), \leq)$ comes from the learning stage

Out: a label of class $c(O)$ for s

3.1 Navigation Principle

The concept lattice can be used as a research area in which we can move depending on the validated features. The first step is the *minimal concept* $(O, f(O))$ meaning that each classes of the symbols are candidate to be recognized and any interval is validated. Then the progression to a next step in the concept

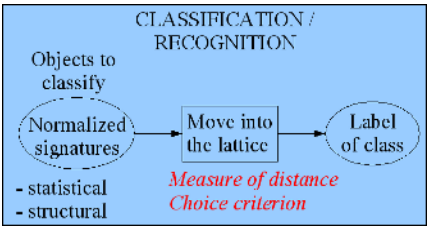


Fig. 3. Schematic description of the classification

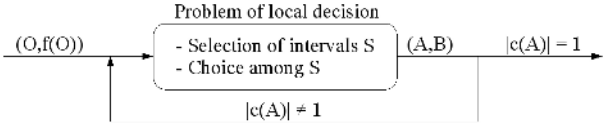


Fig. 4. Progression in the concept lattice

lattice corresponds to the validation of new intervals and consequently to the reduction of the set of symbols. The final step is the *final concept* where the remaining symbols, which are in relation with the whole intervals validated during the progression in the lattice, have the same label. Formally, from the minimal concept $(O, f(O))$, a local decision step is iterated until reaching a final concept (A, B) where $|c(A)| = 1$ (Fig. 4). In each local decision step, we progress in the graph from a *current concept* to one of its successors and a new set of intervals is validated bigger and bigger. It is necessary to define a criterion for the selection of the intervals in a local level. In practice, the progression is done in the Hasse diagram which is the transitive reduction of the concept lattice.

Description of an Elementary Stage of Classification. An elementary stage of classification consists in choosing some intervals in a subset S of intervals selected from the lattice. More precisely, S is deduced from the successors of the current concept in the lattice. Let (A, B) be the current concept, and $(A_1, B_1), \ldots, (A_n, B_n)$ be the n successors of (A, B) in the lattice. Then S is a family of intervals: $S = \bigcup_{i=1}^{n} B_i \backslash B = \{X_1, \ldots, X_n\}$

such that the following properties are satisfied:

- $X_i \bigcap X_j = \emptyset$, $\forall i, j \leq n, i \neq j$
- $|X_i \bigcap I_j| \leq 1$, $\forall i \leq n, \forall j \leq m$, meaning that X_i does not contain 2 intervals from a same feature j.

The computation of the family S of selected intervals is completely defined from the concept lattice. When S is computed, a subset X_i has to be chosen from S, thus the following choice problem: *Choosing* X_i *from* $S = \{X_1, \ldots, X_n\}$.

This choice is a main step of any elementary stage of classification, and therefore of the navigation principle in the lattice whose intend is to provide a class for the symbol s. However, this choice depends on data (and not only on the structure of the lattice as for the computation of S). More precisely, it depends on a *choice criterion* using a *distance measure* between s and an interval x.

Choice Criterion. Any choice from S is described using a *distance measure* according to data, and more precisely a distance between the i^{th} value s_i of the symbol s to be classified, and an interval $x \in I$. We abuse notation and denote such a distance $d(s, x)$ instead of $d(s_i, x)$, thus an extension to a set $X \subseteq I$ of intervals: $d(s, X) = \frac{1}{|X|} \sum_{x \in X} d(s, x)$

We can define many choice criteria depending on data, and sometimes equivalent. Let us propose some simple choice criteria, all of them use the distance d:

1. Choosing i such that $d(s, X_i)$ is minimal.
2. Choosing i such that $|X_i \bigcap I_k| = |\{x \in X_i \bigcap I_k\}|$ is maximal, where I_k is the set of the k first intervals of S sorted according to the distance $d(s, x)$.
3. Choosing i such that $|\{x \in X_i$ such that $d(s, x) < d_c\}|$ is maximal, with d_c a constant.

The second choice uses the principle of the k nearest neighbors [16]. Notice that the third choice is a particular and simplest case of the second choice.

Extensions. It is possible to evaluate the decision risk in each elementary stage of classification (i.e. the confidence degree in a decision) during the navigation in the lattice. For example, a confidence degree for the second choice criterion could be the rate $\frac{X_i \bigcap I_k}{k}$. Such a confidence degree represents another indicator useful for the decision-making. It can be used :

- to try another way of navigation in the lattice from a former explored concept which has given the second best result with the considered choice criterion.
- to compute a more complete signature of the symbol and to make again the whole process.

When we need to search a more accurate signature of the symbol, with new features, and to make again the whole process (discretization and construction of the lattice), it is possible to proceed in an incremental way, meaning without reconstructing the whole lattice, but by a simple addition of the new data.

4 Experimentation

Context. We would like to highlight the links between the decision tree and the concept lattice since both integrate the primitive selection and the classification stages at a time. The decision tree, as the concept lattice, requires a discretization stage and the use of a selection criterion for the feature. Thus it is possible to build the decision tree with the same discretized data. However its construction requires the use of a selection criterion of the feature to be tested at each node of the tree. As a matter of fact, it is possible to obtain a large number of trees with the same data by using different selection criteria. Thus the representation with a decision tree is not only as the one obtained for the lattice.

Figure 5 presents the decision tree (in bold) and the concept lattice associated to the data of the example. The decision tree is built according to a criterion of selection based on a measure of entropy. Notice that the size of the tree is more condensed than the one of the lattice. Indeed, its size is polynomial in the size of the data whereas the size of the concept lattice is exponential in the worst case. Moreover, it is important to notice that each node of the decision tree also is a node in the concept lattice, whatever the selection criterion used for the construction of the tree. Finally, the organization of the structure of the tree forms itself in the lattice, where the tree (in bold) is included in the lattice.

Using a decision tree or a concept lattice, the elementary stage of classification remains the same. Using a tree, the selected intervals S are also defined from

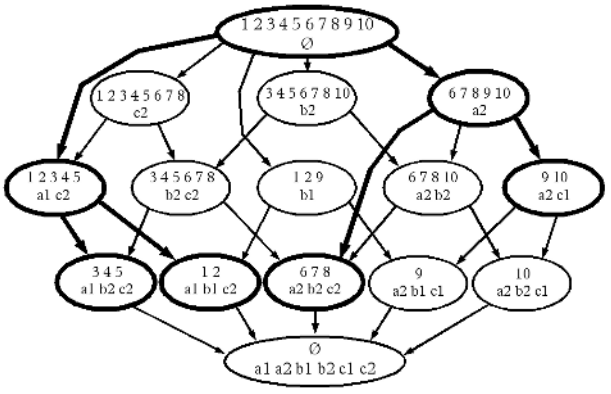


Fig. 5. Inclusion of the decision tree (in bold) in the concept lattice

the successors of a node, where each subset X_i of S is of cardinality 1, and the intervals of all the X_i's correspond exactly to intervals of a same feature j: $S = \{\{x\} : x \in I_j\}$. Therefore the navigation principle is the same with a decision tree and with a concept lattice, depending on a choice criterion defined according to a distance measure. We use the following cutting criteria, distance measure and choice criterion:

Cutting: *maximal distance, entropy or Hotelling's coefficient.*

Distance: $d(s,x) = \frac{\sqrt{(x_m - s)^2}}{\sqrt{(x_m - x_{min})^2}}$ where x_m is the middle of the interval x and x_{min} is the inferior boundary of the interval x. Note that we could replace x_{min} by x_{max} the superior boundary of the interval x and the formula will be the same. This distance is inferior or equal to 1 if the value of the symbol s is in the interval x, and superior to 1 if the value is out of the interval.

Choice: *Choosing i such that $|\{x \in X_i$ such that $d(s,x) < 1\}|$ is maximal. Then in case of multiple choice, choosing i such that $|\{x \in X_i$ such that $d(s,x) < 1,1\}|$ is maximal. Then in case of multiple choice, choosing i such that $d(s,X_i)$ is minimal.*

We compare the recognition rate using these two structures according to: the *signatures* and the *cutting criteria*. This experimentation has been performed with the intention to compare decision tree and concept lattice and not to obtain the best results in terms of recognition. We used symbols of GREC 2003 [17] and characterize them, by several signatures. For each model of symbol, we had 90 symbols noised by the Kanungo et al.'s method [18].

Evaluation of Signatures with the Hotelling's Coefficient Cutting Criterion. We first compare the 6 following signatures : 33 invariants of Fourier-Mellin [19], 50 invariants of Radon transform [20], 24 invariants of Zernike [21] and combination of these 3 signatures : 83 invariants of Fourier-Mellin and Radon combined, 57 invariants of Fourier-Mellin and Zernike combined and 74 invari-

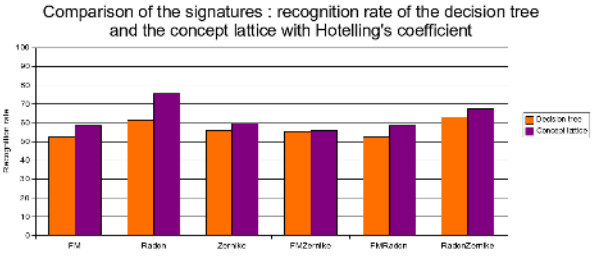


Fig. 6. Evaluation of the recognition with the Hotelling's coefficient

ants of Radon and Zernike combined. We compare these 6 signatures by comparing for each of them the size of the structure and the recognition rate. This experimentation has been realized using the Hotelling cutting criterion.Thus we made a comparison of the size of the structures obtained on the data according to the 6 different signatures. First, with the Hotelling's coefficient as cutting criterion, the number of discretization stages, the number of intervals and the size of the decision tree are almost the same with the whole signatures. However, the size of the concept lattice fluctuates and is smaller for the signature of Radon. Second, the size of the decision tree is really smaller to the one of the concept lattice. Figure 6 shows the recognition rate of the decision tree and the concept lattice according to each signature. The recognition rate is always better for the concept lattice than for the decision tree. Moreover, the Radon signature obtains the best rate of recognition for the concept lattice.

Evaluation of the Cutting Criteria with the Radon Signature. We made a comparison of the size of the structures obtained on this example of data according to the cutting criterion of the discretization stage, i.e. maximal distance, entropy and Hotelling's coefficient. We verify that the entropy and Hotelling's coefficient criteria need a lower number of discretization stages than the maximal distance criterion, because they consider the labels of class of the symbols. The concept lattice size is also smaller with the entropy and Hotelling's coefficient criteria, but it is not the case of the decision tree which has about the same number of nodes with both criteria. The comparative results of the lattice and the tree are shown in Figure 7. This comparative study of efficiency enables the constatation that with the both cutting criteria, the concept lattice improves the recognition results of the noised symbols in comparison with the decision tree. We can add that the best results are obtained with the maximal distance as cutting criterion but it is also the criterion which gives the biggest concept lattice. The Hotelling's coefficient criterion gives almost as good results as the maximal distance one but the size of the concept lattice is really smaller. So, this cutting criterion is the best compromise.

Comparison with Bayesian Classifier and k-NN Classifier. Figure 8 presents the recognition rates of 4 classification methods (Bayesian classifier, k-NN classifier with k=1 and k=3, decision tree and concept lattice) obtained on

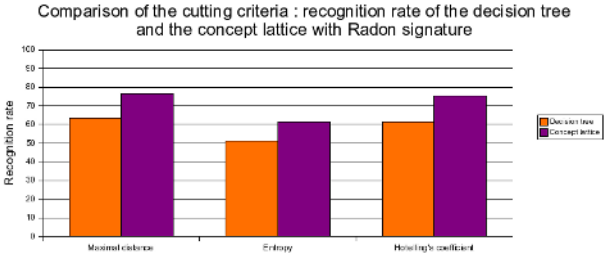


Fig. 7. Evaluation of the recognition with the Radon signature

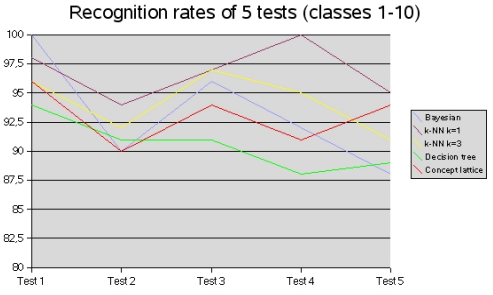


Fig. 8. Recognition rates of the methods obtained for 5 tests

Table 2. Mean recognition rates obtained with 5 tests of the 4 methods (left) and theoretical complexity of the 4 methods (right) with n the size of the learning set, w the number of classes, i the number of values of the signature selected by the cutting criterion, i' the size of the signature where $i \ll i'$, a the number of nodes in the tree and c the number of concepts in the concept lattice where $w \leq a \leq c \leq 2^w$

Mean of recognition rates (%)	*Classes 1-10*	*Classes 11-20*
Bayesian	93,2	94,6
k-NN k=1	96,8	98,4
k-NN k=3	94,2	98,4
Decision tree	90,6	89,8
Concept lattice	93	89,8

Theoretical complexity	*Learning*	*Classification*
Bayesian	$O(wi^2)$	$O(wi^2)$
k-NN		$O(ni)$
Decision tree	$O(ni' + a)$	$O(wi)$
Concept lattice	$O(ni' + c)$	$O(wi)$

two sets of noised symbols of 10 classes (namely classes 1-10 and classes 11-20), with 5 different learning sets computed from each set of data (namely Test 1 to Test 5). Each learning set is composed of 5 symbols per class : 1 model symbol; and 4 noised symbols randomly extracted from the set of noised symbols from which they are removed. Each symbol is described by its Radon signature, restricted to the features selected by the cutting criterion of Hotelling. These selected features are used in entry of each classifier. Notice that recognition rates

really depend on the learning set whatever the method. Table 2 (left) shows the mean results of these 5 tests for classes 1-10. The k-NN classifier gives best results, then the Bayesian classifier, the concept lattice and the decision tree. Table 2 (right) presents a comparison of these methods in terms of complexity of the learning and the classification steps. Notice that best results are obtained by Bayesian and k-NN classifiers directly on numerical data (i.e. without discretization stage). The constraint of these methods are to make an hypothesis on the type of distribution of the data (gaussian, uniform...) for the bayesian classifier and to stock the whole data for the k-NN classifier. Concept lattice and decision tree give lower rates and need discretized data. However, their assets are a good readability, the taking into account of the linked between features in the construction of the graphs and the fact that they don't need hypothesis on the data.

5 Conclusion

The aim of this work is not to reach the best classification results for the moment, but to harmonize a quite un-explored strategy, based on concept lattices, with the well known decision tree method. The size of the decision tree, which is smaller than the concept lattice's one, permits to optimize the processing, but may produce some classification errors, due to the noise. The lattice approach proposes a higher number of classification sequences, and appears to be more adapted to the context of noisy images, to the detriment of a higher dimension. Its other advantage is its good readability. As for the decision tree, a non specialist can easily understand the principle of the progression in the graph by validating intervals. Our future experiments will refer to a comparative study concerning order structures and concept lattices for primitives selection, in the context of an increasing noise, to tackle robustness and scalability problems. Also, our current works deal with the use of concept lattices for a statistical-structural description of the data. Finally, it seems interesting to reduce the construction of the lattice cost, especially through the use of a non exponential but a canonical representation of a lattice, by using a rules system [13, 15], that would permit to generate the lattice on-line, that is to say, to generate the selection stages, if required.

References

1. S. Adam and al., "Multi-scaled and multi oriented character recognition : An original strategy," *ICDAR'99*, pp. 45–48, September 1999.
2. K. Tombre and B. Lamiroy, "Graphics recognition - from re-engineering to retrieval," *Proceedings of 7th ICDAR, Edinburgh (Scotland, UK)*, pp. 148–155, August 2003.
3. S. Canu and A. Smola, "Kernel methods and the exponential family," *National ICT*, 2005.
4. H. Bunke, "Recent developments in graph matching," *15th International Conference on Pattern Recognition*, vol. 2, pp. 117–124, 2000.

5. E. Lefevre, O. Colot, P. Vannoorenberghe, and D. De Brucq, "Contribution des mesures d'information la modlisation crdibiliste de connaissances," *Traitement du Signal*, vol. 17(2), 2000.
6. M. Milgram, *Reconnaissance des formes : mthodes numriques et connexionnistes*. Colin, A., 1993.
7. V. Gunes, M. Menard, and P. Loonis, "A multiple classifier system using ambiguity rejection for clustering-classification cooperation," *IJUFKS, World Scientific*, 2000.
8. G. Birkhoff, *Lattice theory*, vol. 25. American Mathematical Society, 3rd ed., 1967.
9. B. Ganter and R. Wille, *Formal concept analysis, Mathematical foundations*. Springer Verlag, Berlin, 1999.
10. J. Bordat, "Calcul pratique du treillis de Galois d'une correspondance," *Math. Sci. Hum.*, vol. 96, pp. 31–47, 1986.
11. R. Godin and H. Mili, "Building and maintening analysis-level class hierarchies using Galois lattices," *OOPSLA*, pp. 394–410, 1993.
12. L. Nourine and O. Raynaud, "A fast algorithm for building lattices," in *Third International Conference on Orders, Algorithms and Applications*, (Montpellier, France), august 1999.
13. K. Bertet and M. Nebut, "Efficient algorithms on the family associated to an implicationnal system," *DMTCS*, vol. 6, no. 2, pp. 315–338, 2004.
14. R. Taouil and Y. Bastide, "Computing proper implications," *Proceedings of ICCS-2001 Internationnal Workshop on Concept Lattices-Based Theory, Methods and tools for Knowledge Discovery in Databases*, pp. 290–303, 2001.
15. S. Obiedkov and V. Duquenne, "Incremental construction of the canonical implication basis," in *Fourth International Conference Journée de l'Informatique messine*, pp. 15–23, 2003. submitted to Discrete Applied Mathematics.
16. T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
17. GREC, "www.cvc.uab.es/grec2003/symreccontest/index.htm."
18. T. Kanungo and al., "Document degradation models: parameter estimation and model validation," in *IAPR Workshop on machine vision applications, Kawasaki (Japan)*, pp. 552–557, 1994.
19. S. Derrode, M. Daoudi, and F. Ghorbel, "Invariant content-based image retrieval using a complete set of fourier-mellin descriptors," *Int. Conf. on Multimedia Computing and Systems (ICMCS'99)*, pp. 877–881, june 1999.
20. S. Tabbone and L. Wendling, "Recherche d'images par le contenu l'aide de la transforme de radon," *Technique et Science Informatiques*, 2003.
21. M. Teague, "Image analysis via the general theory of moments," *Journal of Optical Society of America (JOSA)*, vol. 70, pp. 920–930, 2003.

An Extended System for Labeling Graphical Documents Using Statistical Language Models

Andrew O'Sullivan[1], Laura Keyes[1], and Adam Winstanley[2]

[1] School of Informatics and Engineering, Institute of Technology Blanchardstown, Dublin 15, Ireland
Andrew.O'Sullivan/Laura.Keyes@itb.ie
[2] Department of Computer Science, NUI Maynooth, Maynooth, Co. Kildare, Ireland
Adam.Winstanley@nuim.ie

Abstract. This paper describes a proposed extended system for the recognition and labeling of graphical objects within architectural and engineering documents that integrates Statistical Language Models (SLMs) with shape classifiers. Traditionally used for Natural Language Processing, SLMS have been successful in such fields as Speech Recognition and Information Retrieval. There exist similarities between natural language and technical graphical data that suggest that adapting SLMs for use with graphical data is a worthwhile approach. Statistical Graphical Language Models (SGLMs) are applied to graphical documents based on associations between different classes of shape in a drawing to automate the structuring and labeling of graphical data. The SGLMs are designed to be combined with other classifiers to improve their recognition performance. SGLMs perform best when the graphical domain being examined has an underlying semantic system, that is; graphical objects have not been placed randomly within the data. A system which combines a Shape Classifier with SGLMS is described.

1 Introduction

This paper describes a graphical object recognition framework that applies statistical models to graphical notation based on associations between different classes of object in a drawing to automate the structuring of graphical data. Graphics recognition comprises the recognition and structuring of geometry such as points, lines, text, symbols on graphical documents into meaningful objects for use in graphical information systems for example, Computer Aided Design (CAD), Geographical Information Systems (GIS) and multimedia systems. All of these systems need to capture, store, access and manipulate large volumes of graphical data. For semantic capture of paper/digital data, not only the geometry but also attribute data describing the nature of the objects depicted must be stored, thus representing the graphical data in a high-level object-oriented format for description and semantic analysis. This structuring into composite objects and the addition of labeling attributes is typically a manual, labour intensive, expensive and error-prone process. The automatic structuring and labeling of graphical data is desirable.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 61 – 75, 2006.
© Springer-Verlag Berlin Heidelberg 2006

This semantic capture and analysis of graphical data is difficult to automate. Graphical object recognition is a sub-field of pattern recognition and includes classification and recognition of graphical data based on shape description of primitive components, structure matching of composite objects and semantic analysis of whole documents. Previous work by authors and colleagues devised and evaluated a graphics recognition system for labeling of objects and components on drawings and plans based on their shape [1]. Shape description has proved successful in distinguishing graphical objects, with classification confidence up to 80% depending on the domain, however, no one shape method provides an optimal solution to the problem. Automation of the structuring and recognition of objects through statistical modeling for efficient and complete input into graphical information systems can form a solution to this complex problem. That is, treating the graphical document as a language, statistical language modeling is applied through a statistical graphical language model framework.

Statistical Language Models (SLMs) are successful methods used in Natural Language Processing (NLP) for recognising textual data. SLMs estimate the probability distributions of letters, words, sentences and whole documents within text data. They have been used for, among other tasks, Speech Recognition [2] and Information Retrieval [3]. This work investigates the use and adaptation of SLM techniques that is, Statistical Graphical Language Models (SGLMs) to aid in the semantic analysis of graphical data on graphical documents. The proposed framework will apply statistical models to graphical languages (CAD data) based on the associations between different classes of shape in a drawing to automate the structuring of graphical data and to determine if SLMs have applicibility to improve the classification of graphical objects as they do for NLP applications. A SGLM module to extend the system for labeling and semantic analysis of graphical documents to improve performance is applied.

In this paper, SGLMs for graphics recognition is presented. Section 2 describes SLMs as a method used in natural langauge processing and their application to graphical data. It outlines the similiarities between natural language and the language characterised by graphical data that support the application of SLM to graphical notation and shows how N-gram models, a widely used SLM technique, can be used to build SGLMs for the recognition of unknown objects within CAD drawings for engineering plans. Section 3 depicts the graphical recognition system used and the application of the SGLM module to extend the system for labeling and semantic analysis of graphical documents. Section 4 describes the background to this work, the experimental work carried out and discusses the results. Section 5 concludes and outlines future work.

2 SLMs and Graphical Object Recognition

Statistical Language Models (SLMs) are estimates of probability distributions, usually over natural language phenomena such as sequences of letters, words, sentences or whole documents. First used by Andrei A. Markov at the beginning of the 20th century to model letter sequences in Russian literature [4], they were then developed as a general statistical tool, primarily for NLP. Automatic Speech Recognition is arguably the area that has benefited the most from SLMs [2] but they have also been used in many

other fields including machine translation, optical character recognition, handwriting recognition, information retrieval and augmentative communication systems [5].

There are different types of SLMs that can be used. These include *Decision Tree* models [6], which assign probabilities to each of a number of choices based on the context of decisions. Some SLM techniques are derived from grammars commonly used by linguists. For example Sjilman et al. [7] use a declarative grammar to generate a language model in order to recognise hand-sketched digital ink. Other methods include *Exponential* models and *Adaptive* models. Rosenfeld [8] suggests that some other SLM techniques such as *Dependency* models, *Dimensionality* reduction and *Whole Sentence* models show significant promise. However this research will focus on the most powerful of these models, *N-grams* and their variants.

2.1 N-gram Models for Predicting Unknown Words in NLP

N-gram models are the most widely used SLM technique. In NLP N-grams are used to predict words based on their N-1 preceding words. The most commonly used N-grams are the bigram model, where N=2 and the trigram model, where N=3. That is, a bigram model uses the previous word to predict the current word and a trigram model uses the two previous words. These probabilities are estimated by using the relative frequencies of words and their co-occurrences within a training corpus of natural language examples.

For bigram models, the corpus of data is analysed for the relative frequencies of pairs of words that occur together. For instance if the last sentence was analysed the following pairs would be recorded: “For bigram”, “bigram models”, “models the” and so on. The same applies for trigram models, except the corpus is analysed for triples, not pairs, of words that occur together. Bigram tables and trigram tables store these frequencies, which are then used to predict unknown words. These probabilities can be estimated using the equations (1) and (2), respectively.

$$P(w_i \mid w_{i-1}) = C(w_{i-1}\, w_i) / C(w_{i-1}) \tag{1}$$

$$P(w_i \mid w_{i-1},\, w_{i-2}) = C(w_{i-2}\, w_{i-1}\, w_i) / C(w_{i-2}\, w_{i-1}) \tag{2}$$

where C represents the frequency of words occurring together. The right hand sides of equations (1) and (2) are computed from the bigram and trigram tables, correspondingly. The w_i that results in the highest frequency and hence the highest probability is judged to be the next word in the sentence. The corpora required for this process are usually extremely large and contain a wide range of examples of natural language. For example the Brown Corpus [4] contains one million words taken from fifteen different sources such as legal text, scientific text and press reportage. It should be noted however that corpora can be constructed to just include a particular subset of language, if so required for a particular task.

2.2 SLMs for Labeling Graphical Documents

SLMs have previously been used almost exclusively for NLP. There are sufficient similarities between natural language and graphical notations that suggest that adapting SLMs to become SGLMs is a worthwhile approach [9].

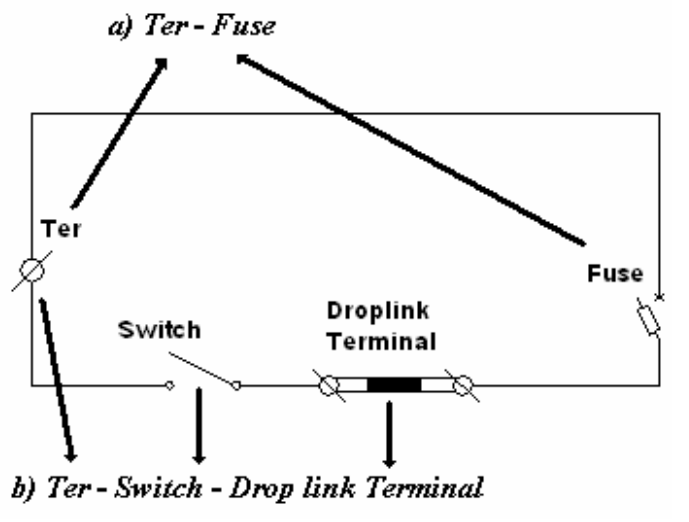


Fig. 1. Sample electrical circuit and phrases constructed

Recent work applied SLMs to the automatic structuring of topographic data [10] for Geographical Information Systems (GIS). In their work Winstanley and Salaik characterise the similarities that can be drawn between topographic data and natural language. Both consist of discrete objects (words, graphical objects) and these objects:

- have a physical form (for example spelling, object shape);
- have a semantic component (meaning, graphical object label);
- are classified according to function (part of speech, object class) and
- are also formed into larger components (sentences/paragraphs, diagrams/ documents).

A similar analogy can be used for natural language and graphical data found on architectural or engineering plans. By considering the graphical data as a language with its own syntax and vocabulary the analogy becomes:

- Word – particular object
- Spelling – configuration of graphic components of object (shape)
- Part-of-speech – type of object (relay, resistor)
- Phrase – connected sequence of objects

Using this framework N-gram models can be constructed that build phrases representing graphical data on drawings and plans. Figure 1 shows a sample circuit and phrases that can be constructed for a graphical language.

2.3 SGLMs for Labeling Graphical Documents

As in NLP, a corpus of training data is needed for SGLM. This training data must contain examples of graphical objects in their contextual use that is, actual real world documents. N-gram tables must be built which contain the relative frequencies of

co-occurrences of the graphical objects. This requires the counting of occurrences of phrases of objects within the corpus. It is here that one of the major differences between natural language and graphical notations is noted. Natural language is a one-dimensional sequence of symbols, whereas graphics are inherently multi-dimensional. This difference is significant in relation to N-grams as the one-dimensionality of natural language makes the choice of which words to use for phrase construction and counting an easy one that is, the preceding words of the unknown word. With graphical notation however, there can be numerous other objects neighbouring the unknown object. This makes the choice of which of these neighbouring objects to use to construct object phrases a harder one. One approach to dealing with this is to use adjacency relationships between objects on a document.

2.4 Object Adjacencies

In SGLMs, neighbouring objects are used to form object phrases. How the term *neighbouring* is defined will govern how the object construction process works. Object *adjacencies* are used for this purpose, with the adjacencies defining how objects relate to each other. Once an adjacency is defined for a particular domain or diagram all the objects within that data that are adjacent to one another can be used to form object phrases, for storage in the *N*-gram tables. Defining the object adjacency rules that will govern how the object phrases are constructed is an important decision in designing SGLMs. There are several ways to define *adjacent* in this context. Experimental work undertaken so far has used a corpus of graphical documents consisting of electrical circuits. Objects are defined as being *adjacent* to one another if they are connected by a wire. For example in Figure 1 two examples of phrases of objects that can be constructed using this adjacency definition are shown. Part a) shows the constructed bigram phrase "*Ter – Fuse"* and part b) shows a trigram phrase "*Ter – Switch – Droplink Terminal"*.

This method of defining adjacency does not take into account higher-level information about electrical circuit diagrams. For example objects within circuits can occur in series or parallel with each other. Objects in series relate differently to other objects than if they were in parallel. This suggests that the object *adjacencies* should

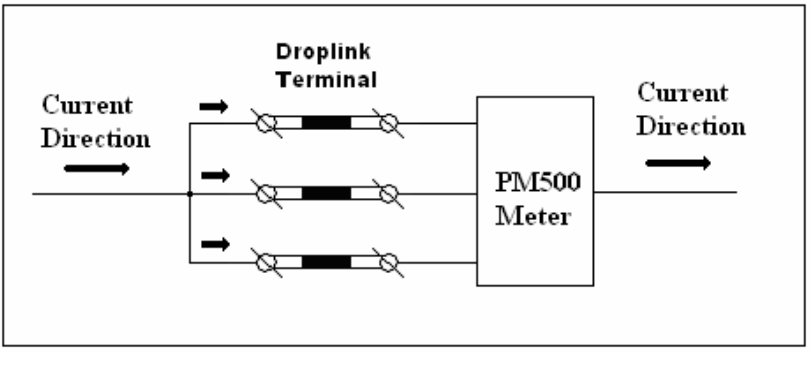


Fig. 2. Electrical Circuit example with current direction indicated by black arrows

attempt to model these differences, perhaps by having more than one type of adjacency. This however would introduce more complexity into the process and for the experimentation conducted so far objects in parallel were not treated differently to objects in series. Possible solutions to this problem involve determining different ways of defining object neighbours and counting phrases. Options include the use of direction in the adjacency definition. For example object phrases could only be formed in the direction of the current. So in Figure 2 below, the Droplink Terminal Objects would only form phrases with the PM500 Meter object and not each other. This is ongoing work being investigated by authors in current experiments.

2.5 N-gram Models for Predicting Unknown Graphical Objects

All of the phrases extracted from the corpus are used to build bigram and trigram frequency tables. The frequencies of phrases are known but the relative frequencies must be obtained as they estimate the probabilities. The relative frequencies of N-gram phrases are computed by dividing the frequency of a phrase by the total frequency of that phrase. Relative frequencies for bigram and trigram phrases are computed using equations (1) and (2) in section 2.1. The resulting bigram and trigram tables are used to predict unknown objects.

One problem associated with N-grams is the data sparseness problem. This means that there are some events within the N-gram tables that have a probability of zero. This is because those events did not occur in the training data so they have a frequency and hence probability of zero. These events therefore will not be considered in any future prediction process, even though the events may actually occur in the future. The data set used in this work is limited in terms of size so such zero-probabilities were expected. However, even with extremely large datasets, zero-probabilities occur. A solution to this problem is *Smoothing*. *Smoothing* attempts to give probability values to events with zero probability. There are several *Smoothing* techniques available but here *Add-One Smoothing* is used. This is a simple technique where the value '1' is added to all the entries in the bigram and trigram frequency tables. So any event, which previously had a zero frequency, will now have a frequency and a probability.

3 Graphics Recognition System with SGLM

This work investigates the use and adaptation of SLM techniques i.e. Statistical Graphical Language Model (SGLMs) to aid in the semantic analysis for structuring and labeling graphical data on technical documents for the purposes of recognition, indexing and retrieval. An earlier system has been developed for the recognition and labeling of graphical objects where the underlying classifier is based on shape recognition [1]. Shape methods are applied to object boundaries extracted from drawings represented as vector descriptions.

3.1 Shape Classifier

To assess the capability of the SGLM to improve the performance of other classifiers, a classifier was implemented which is based on simple set of shape descriptors. The

shape classifier, implemented in Matlab, uses the following six descriptors to classify the graphical entities:

- Bounding Box width to height ratio. The bounding box is the smallest rectangle to enclose the symbol.
- Minor Axis Length to Major Axis Length ratio. The length of the minor and major axis' of the ellipse that has the same second-moments as the region.
- Eccentricity. The ratio of the distance between the foci of the ellipse and its major axis length.
- Euler Number. The number of objects in the symbol minus the number of holes in those objects.
- Solidity. The proportion of the pixels in the convex hull that are also in the symbol.
- Extent. The proportion of the pixels in the bounding box that are also in the symbol.

The output obtained by the description methods provides a measurement of shape that characterises the object type. These shape descriptors provides a list of candidate classes of each object. Extending this system with SGLM is envisaged as a possible means of improving the performance of the overall graphical object recognition system. The SGLM model is combined with the score produced by shape classifier to improve the likelihood that the classification is correct or re-classify incorrect or misclassified features. Figure 3 shows the configuration of the recognition system and the role of SGLM within this system.

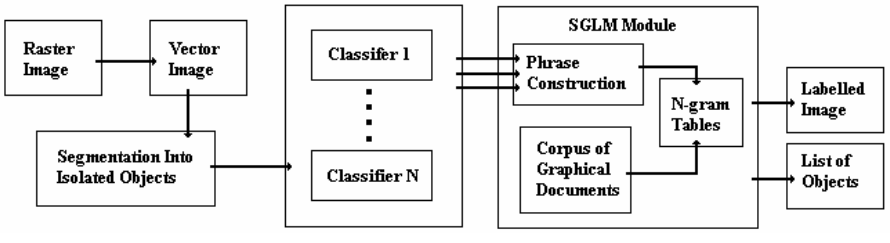


Fig. 3. Extended Recognition System

3.2 Combining N-grams with Shape Classifier

It is suggested by this research that the main benefit of adapting N-grams to work for graphical notations is in improving the performance of other classification technique. It is proposed to use the developed N-grams to improve the performance of a shape classifier. In a document for each unknown object the shape classifier produces a candidate list of possible identities. These possible identities and the identities of the object's neighbours are then used to construct object phrases. The N-gram tables are then consulted to find the most probable of these phrases and hence find the most probable identity for the unknown object. For example, Figure 4 shows the same circuit as in Figure 1, except that the switch's identity is unknown. There are two possible trigram phrases involving the unknown object:

- "*Fuse – Ter – Unknown Object*"
- "*Fuse – Droplink Terminal – Unknown Object*"

Table 1. Sample shape classification of unknown object

Classification	Probability
Isolator	0.4536
Switch	0.3241
Ter	0.1532
ELU	0.0072

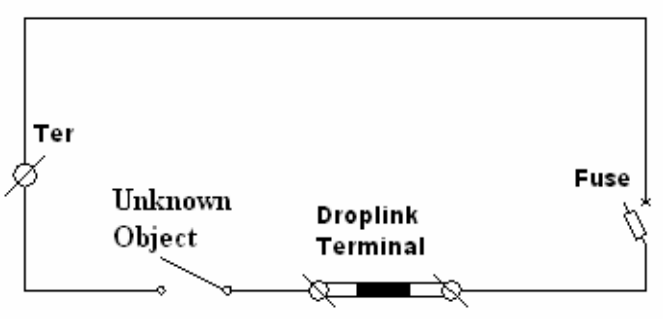


Fig. 4. Sample of circuit with unknown object

Table 1 shows sample results of shape classification for the unknown object. Combining these with the two trigram phrases taken from Figure 4 gives eight candidate phrases:

- *"Fuse – Droplink Terminal – Isolator"*
- *"Fuse – Droplink Terminal – Switch"*
- *"Fuse – Droplink Terminal – Ter"*
- *"Fuse – Droplink Terminal – ELU"*
- "*Fuse – Ter – Isolator*"
- *"Fuse – Ter – Switch"*
- *"Fuse – Ter – Ter"*
- *"Fuse – Ter – ELU"*

The trigram table can now be checked to see which of the eight phrases is the most frequent and hence which identity to assign the unknown object.

The combination of shape classification and N-grams described, form a major part of this projects work. Another major part will develop Part-of-Speech (POS) tagging for use with the graphical notation. POS tagging is a technique that is used in NLP to assign tags to words. Examples of these tags are noun, verb adjective and pronoun. Tags can be assigned to graphical objects by using the equation:

$$P(\text{object shape} \mid \text{tag}) * P(\text{tag} \mid \text{neighbouring k tags}_i) \quad (3)$$

This is the probability of an object belonging to a particular class combined with the likelihood that that class would have the observed neighbouring class (of neighbours up to *k* deep).

Part of this research will be to ascertain the best way to define tags for graphical objects. For example, with the electrical data, tags could be based on hierarchical classes e.g. an object is identified as a Resistor and its tags are the various types of Resistor such as 10 Ohm, 20 Ohm etc. Another approach could be to define tags based on the object's function within the circuit e.g. an object could be a Meter, a Relay or a User.

4 Experimental Work

Experimental work to determine the applicability of SLMs to graphical data was carried out in two phases. Firstly SLMs were applied on their own to the data. This initial step was used to establish the feasibility of applying SLMs in the form of SGLMs to deal with a graphical language. Secondly the SGLMs were combined with a set of simple shape classifiers and the results evaluated. The following sections outline each experiment and discuss the results obtained. Figure 5 shows sample vocabulary of graphical language used in this work. The amount of data available for use in this work was limited at the time of these experiments; however, this work and corpus of data constructed are used to determine the viability of this novel approach to graphical object recognition.

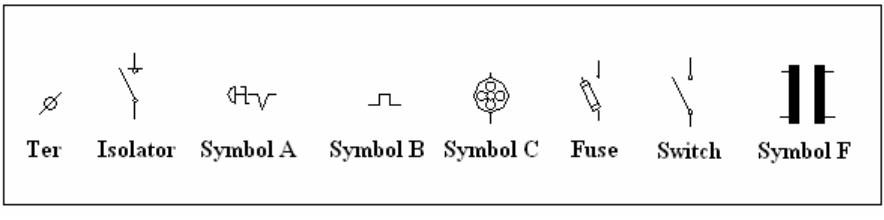


Fig. 5. Sample vocabulary used in experiment

4.1 Effectiveness of SGLMs on CAD Data

This works forms part of a project to develop an online Operation and Maintenance information system. The O&M System allows a user to select an example object (simple or composite) and the software finds similar objects in the same or other drawings. The tool generates data structures that can be used to build multimedia linkages between objects, drawings and related information. The information is accessed through a standard web browser interface including navigation through hot-links and keyword search facilities. CAD drawings showing the location of utilities and services also act as browser navigational maps. The system can be implemented for all sizes of installations but comes particularly suited for the infrastructure management of large industrial or service sites. Current use relevant to this paper is electrical data for business park sewage pumping station.

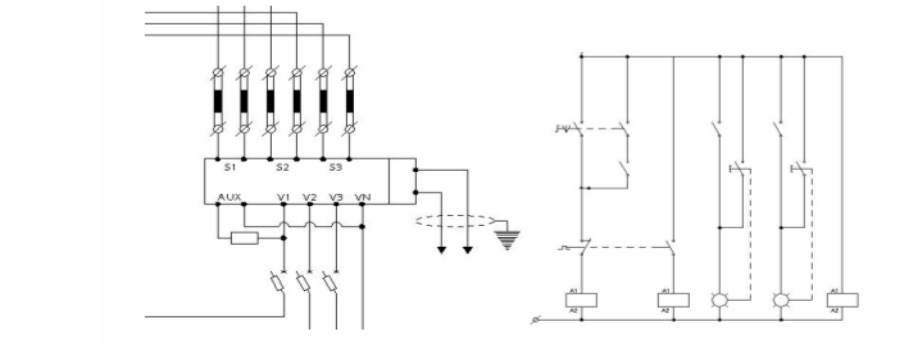


Fig. 6. Samples of the electrical circuits used in the work

Figure 6 shows examples of electrical data used in this experiment. The graphical data used for this work consists of electrical circuits. In total 18 electrical diagrams were used. The diagrams contain 738 graphical objects (excluding wire connection lines) and there are 24 different objects types. A bigram model and a trigram model were implemented on the graphical notation. For the bigram model phrases of pairs of objects which occurred together within the data were counted. Likewise for the trigram model triples of co-occurring objects were counted. The bigram phrases were stored in a 2-dimensional array, where the index *(i, j)* corresponds to the number of times the Object_i occurred with the Object_j. The trigram phrases were stored in a similar 3-dimensional table.

The N-gram tables were tested on two unseen electrical diagrams. The first diagram tested contained 39 objects and 8 object types. The second diagram contained 30 objects and 6 object types. Each object was treated as an unknown object and its adjacent objects were used to construct bigram and trigram phrases. The probabilities of these phrases were then combined into one final prediction for each object by three different voting combination methods: Majority Vote, Sum Rule and Maximum [11]. Table 2 shows the performance results of the bigram and trigram models in terms of the percentage of objects they classified correctly. Table 3 shows a more detailed breakdown of the trigram model's performance with the first diagram.

4.1.1 SGLMs Results Discussion

These experiments were used to determine the applicability of applying SLMs to graphical CAD data. N-grams are not primarily designed to work on their own so the low percentage rates of objects correctly predicted are not unusual. The small size of the test data is also an obvious factor in the results. As the project continues and the test data is enlarged with more object types and contextual use examples added, the performance of the N-grams should improve.

Table 2. Bigram and trigram performance results

N-gram:	**Bigram**			**Trigram**		
Combination Method:	**Majority**	**Sum**	**Max**	**Majority**	**Sum**	**Max**
Drawing 1:	33%	30%	13%	44%	49%	44%
Drawing 2:	30%	30%	17%	37%	37%	17%

Table 3. Detailed trigram performance results for Drawing 1

Object Type	Total Number of Objects	Amount Predicted Correctly			Percentage Predicted Correctly		
		Majority	Sum	Max	Majority	Sum	Max
Switch	18	11	11	11	61	61	61
Symbol A	2	0	0	0	0	0	0
Symbol B	3	0	0	0	0	0	0
Symbol D	1	0	0	0	0	0	0
Symbol E	3	0	0	0	0	0	0
Ter	6	0	2	0	0	33	0
ELU	5	5	5	5	100	100	100
HOA	1	1	1	1	100	100	100

When the N-grams were used on their own on the electrical diagrams they displayed typical behavior. N-grams are highly sensitive to their test data, with objects or events with high frequencies within the test data predicted with large frequency during classification processes. For example many objects were misclassified as *Switch* during the testing as *Switch* is one of the most frequent objects within the data. Likewise, as Table 3 shows, none of the entities of object type *Symbol A, Symbol B, Symbol D or Symbol E* were correctly predicted. This is due to their low frequency within the test data. The objects of type *ELU* however, which have high frequency values, were 100% correctly predicted.

The trigrams performed better than the bigrams, again this was expected as bigrams use less information than trigrams who use two neighbouring objects to form a phrase. If N was increased to four, to form a Quadgram table, the performance could be improved further. The complexity of the process however would be increased significantly.

4.2 Shape Classifier Combined with SGLMs

The SGLMs combined with the shape description approach were tested on two electrical diagrams, consisting of 43 electrical symbols and 14 symbol types. For each symbol the shape classifier produces a ranked list of possible symbol types. The candidates are ranked based on the distance between the unknown symbol's descriptor values and the ground truth values of the symbol types. In this classification if the top 2 ranked scores are within 5% the classification is deemed to be uncertain. And the SGLMs employed. The unknown symbol's neighbours are used to create symbol

phrases. The symbol, which in combination with the symbol phrases has the highest frequency value within the training data, is judged to be the identity of the unknown symbol.

Table 4. Recognition perfromance results

		Recognition Performance: % recognised correctly			
Object Type	**Amount in Test Data**	**Shape**	**Shape + Trigram**	**Shape + Bigram**	**Shape + (Trigram When Shape Uncertain)**
Ter	8	25	62.5	0	62.5
Mi-crologic	3	10	66.67	0	100
ELU	8	87.5	100	0	87.5
Symbol A	2	100	0	0	100
Symbol B	1	0	0	0	0
Symbol C	1	100	0	0	100
Symbol E	1	100	0	0	100
Symbol F	3	100	0	0	100
Switch	9	66.67	66.67	66.67	88.89
Fuse	2	100	100	0	100
Droplink	1	100	100	100	100
PM500	1	100	100	0	100
Isolator	1	100	100	100	100
ASP	1	100	0	0	100
HOA	1	100	100	0	100
Total	**43**	**74.4**	**62.79**	**18.6**	**86**

4.2.1 Combined Approaches Results Discussion

The shape classifier recognised 32 of the 43 symbols correctly, a rate of 74.4%. When the Trigram SGLM module is used in combination with the shape classifier on every symbol, 27 symbols are recognised correctly, which at a rate of 62.79%, is a decrease of 11.61%. There is a decrease in recognition performance because the SGLM module typically fails to recognise symbols that have low frequency within the training data. For example, Symbols A, B, C E and F occur relatively infrequently within the training data and as seen in Table 4 the SGLM failed to recognise them within the test data. When the bigram SGLM is used the recognition rate falls severely to 18.6%. This is to be expected as bigram models typically perform worse than trigram models as they make use of less information. In this case the information in question is the identities of the neighbouring symbols. The low bigram recognition rate can be

viewed as proof that by using more of the neighbouring symbols the performance of the SGLM improves.

When the trigram SGLM module is used in combination with the shape classifier only when the shape classifier is uncertain about classification, 37 of the symbols are recognised. This is a recognition rate of 86%, which is a 11.6% increase in recognition from the original rate of 74.4%. By using the SGLM only when the shape classifier is uncertain, symbols that the SGLM might fail to recognise have the chance to be recognised by the shape classifier. Likewise, when the shape classifier is uncertain, the symbols in question have a chance to be recognised by the SGLM. This method of using both recognition techniques has resulted in an increase in recognition performance, which shows promise for the use of SGLM.

It should be noted that in this test, it is assumed that when the SGLM module is used to classify a symbol, the identities of the neighbouring symbols are known. This of course might not be the case so an option is to use the shape classifier to temporarily classify any unidentified neighbouring objects and use these temporary identities for use with the SGLM for the current symbol. Further tests will assess the performance of this approach.

Another factor of interest is the number of neighbouring symbols to use. At present the bigram and trigram models have been used, which use one and two neighbouring symbols respectively. An increase in this number could result in improved results, as more information is used. A Quadgram for example would form three-symbol phrases from the neighbouring symbols. This increase however would result in an increase in computational expense. One problem with forming symbol phrases within the electrical domain that is the focus of these tests is the number of wire connections within the electrical circuits can result in a large number of phrases being created. An increase in the number of symbols used could result in an even larger number of phrases created, which increases the computational expense.

5 Conclusion and Future Work

This paper has proposed the adaptation of Statistical Language Models for recognition and labeling of graphical objects within architectural and engineering documents. Previously used for Natural Language Processing there exists similarities between natural language and technical graphical data that suggest that Statistical Graphical Language Models is a worthwhile approach. Digitised CAD drawings for electrical data are processed to extract their component objects. SLM are applied and N-gram phrases are constructed. Initial experiments apply SGLM without the combination of other classifiers to determine their applicability and effectiveness at classifying graphical objects. Results show classification rates of less than 50% for bigram model and 61% and 100% for certain instances of graphical objects using trigram model. The size of data used in the experiment is a factor in results. However, it is envisaged that with bigger amounts of data for training and testing and increased frequencies of graphical objects in data, the performance of SGLM will improve.

The SGLMs are designed to be combined with other classifiers to extend previous recognition system. Using this approach SGLMs are applied based on associations between different classes of ‘shape’ in a drawing to automate the structuring and

labeling of graphical data. Digitised CAD drawings are processed to extract their component objects from which shape descriptions are built. These feed into several description and matching algorithms, each of which produces one or more candidate categories to which each object may belong. An overall consensus decision gives a ranked list of candidate types. The SGLM module can then be used to improve the performance of the recognisers.

Combination of scores can take the form of voting methods such as majority vote or borda count. An extension of N-grams used in NLP is to count the part-of-speech of the word (noun, verb and so on) rather than the word itself. This n-gram part-of-speech tagging model can be used with shape for graphical data, where the tag is some descriptive classification of the graphical object. It s envisaged that tagging will provide a effective means of combining SGLM module with the existing graphical recognition system.

The experiments conducted so far to evaluate SGLMs have been conducted on a limited dataset. Training corpora used in Natural Language Processing however, can contain millions of words. The next stage in evaluating SGLMs is to undertake a large-scale experiment, with a significantly larger number of graphical diagrams and objects used. The authors are currently undertaking this experiment with electrical circuit diagrams. There is a vastly increased vocabulary of graphical objects being considered and as such the number of circuit diagrams needed is also immensely increased. Whereas the previous experiments involved 18 diagrams, the present research involves thousands.

Different approaches to the adoption and application of SGLM will be carried out. Other possibilities include different ways of defining the *adjacency* of objects, Different vote combination methods such as Borda Count, Minimum and Median will be computed to find the optimal method. Part-of-Speech tagging as a way of combining modules will be exhaustively tested. A final SGLM module can be used to extend and improve the performance of system for the labeling and semantic modeling of graphical documents.

References

1. Keyes, L., Winstanley, A., "Shape Description for Automatically Structuring Graphical Data", in Josep Llados, Y.B. Kwon (eds), Graphics Recognition – Recent Advances and Perspectives, LNCS 3088, 353-262, Springer-Verlag, 2004
2. Jelinek, F., Statistical Methods for Speech Recognition. MIT Press 1997
3. Ponte, J.M., Croft, W.B., "A Language Modeling Approach to Information Retrieval", Proceedings of SIGIR'98, 1998, 276-281
4. Manning, C.D., and Schutz, H., Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, 2001.
5. Jurafsky, D. and .Martin, J.H., Speech and Language Processing, Prentice-Hall, 2000.
6. Bahl, L R., Brown, P. F., Peter V. de Souza and R. L. Mercer., "A Tree-based Statistical Language Model for Natural Language Speech Recognition." IEEE Transactions on Acoustics, Speech and Signal Processing, 37:1001-1008, July 1989.
7. Shilman, M., Pasula, H., Russell, S. and Newton, R., "Statistical Visual Language Models for Ink Parsing." AAAI Spring 2002 Symposium on Sketch Understanding, 2002.

8. Rosenfeld, R., "Two Decades of Statistical Language Modeling: Where Do We Go From Here?", Proceedings of the IEEE, 88 (8), pp 1270-1278, 2000.
9. Andrews, J.H., Maps and Language, A Metaphor Extended, *Cartographic Journal*, 27, 1–19, 1990.
10. Winstanley, A., B. Salaik, L. Keyes: "Statistical Language Models For Topographic Data Recognition", IEEE International Geoscience and Remote Sensing Symposium (IGARSS'03), July 2003.
11. J. Kittler, M. Hatef., R.P.W. Duin and J. Matas, "On Combing Classifiers" IEEE Transaction on Pattern Analysis and Machine Intelligence, 20 (3), 226-239, 1998.

Symbol Recognition Combining Vectorial and Statistical Features

Hervé Locteau, Sébastien Adam, Éric Trupin, Jacques Labiche, and Pierre Héroux

LITIS, Université de Rouen,
F-76800 Saint-Etienne du Rouvray, France
Herve.Locteau@univ-rouen.fr

Abstract. In this paper, we investigates symbol representation introducing a new hybrid approach. Using a combination of statistical and structural descriptors, we overcome deficiencies of each method taken alone. Indeed, a Region Adjacency Graph of loops is associated with a graph of vectorial primitives. Thus, a loop is both representend in terms of its boundaries and its content. Some preliminary results are provided thanks to the evaluation protocol established for the GREC 2003 workshop. Experiments have shown that the existing system does not really suffer from errors but needs to take advantage of vectorial primitives which are not involved in the definition of loops.

Keywords: Graph Matching, Symbol representation, Symbol Recognition, Vectorisation, Moment Invariants.

1 Introduction

Managing huge amount of digital image of documents implies an effective knowledge extraction process and a suitable representation. Thus some systems have been designed taking advantage of well defined rules regarding the involved domains. Nevertheless, this knowledge - parameters, scenarii - may appear scattered in the code. In this way, a new case of study may lead to start from scratch the development of a new system. That is why new systems have to be designed putting out as much as possible the knowledge relative to image treatment and to the application domain.

As each domain introduces its own graphic notation, which is not really standardized in some cases (e.g. architecture), the automatic interpretation of corresponding images of documents with a generic approach relies on the ability to discover and learn the corresponding notation. Designing an automatic interpretation system is more or less ambitious and researches have been mainly devoted to automatic conversion of graphic documents into a readable format for CAD systems, that it to say, a set of vectorial primitives. The raster-to-vector conversion is a generic step but systems have to provide information closest to the domain. They have to enable the interpretation of the drawing, to give meaning in a semantic way. In this way, the very first studies to get the high level

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 76–87, 2006.
© Springer-Verlag Berlin Heidelberg 2006

representation of a drawing have been designed based on structural or syntactic representations consisting of vectorial primitives ([1]).

As [2] stated, typical applications in the field of graphical recognition, for example, backward conversion from raster images to CAD, hand-drawn based user interfaces for design systems or retrieving by content in graphical document databases, involve symbol recognition processes. We investigate in this study the symbol recognition step. Symbol recognition consists in locating and identifying the symbols on a graphical document.

Many types of documents contain symbols that appear connected to other graphic components, making symbol extraction a difficult task [3]. Among the application domains and the corresponding specificities for the process of symbol recognition we focus on, we can distinguish three main classes of documents: technical drawings; maps; musical scores and mathematical formulas. Regarding technical drawings, symbols are mainly embedded in a net of lines, without any predefined orientation or scale. A huge number of informations can be drawn in a restricted region of the document. Thus, the problem of symbol recognition depends on the crowded context in which they are located.

Moreover, images can be degraded since the documents are themselves degraded or since the acquisition process is unreliable. From this point of view, document image analysis is accomplished by building a hierarchical perception of the document from raster to high level objects belonging to the domain ([1]). Symbol recognition must be designed regarding obvious constraints among which invariance to affine transforms, segmentation, degradation and scalability. The GREC 2003 symbol recognition contest has proposed a performance evaluation framework to compare various works on this topic [4].

We can find a classical classification of the existing modelisation: structural approaches and statistical approaches.

As we remind it before, many structural models involve vectorial primitives that are embedded in a structure where relations are geometric constraints ([5, 6, 7]) among which *interconnection*, *tangency*, *intersection*, *parallelism*. Nevertheless, other geometric primitives have been used (loops or contours for example). Matching consists in finding a subgraph isomorphism between the input graph and the prototype graph. Error-tolerant subgraph isomorphism has then became an important field of research to decrease the computational complexity [8, 9, 10]. Structural approaches can be very sensitive in presence of noise or deformation but the rotation and scale-invariance can be easily acquired.

In statistical pattern recognition, each entity is being assigned a feature vector extracted from a portion of the image. Classification is then achieved using a partition technique of the feature space. Among features, we can cite studies on geometric features, moment invariants or image transformations ([11, 12, 13, 14, 15], [16, 17]). Since statistical approaches work at the pixel-level, they do not rely on a thinning and a vectorisation process. However, some signatures suffer from scale and rotation estimator when trying to define the invariance. At least, statistical approaches have to be defined on a well defined region of interest which is not an easy task dealing with symbols embedded in a net of lines.

The remainder of the paper is organized as follows. We present in section 2 an exhaustive graph representation for symbols based on regions in terms of theirs boundary and content. In section 3, we propose a system that illustrates advantage of such a model while section 4 deals with the use of statistical features. We introduce in section 5 a generic preprocessing method to correct segmentation errors on technical documents. Experiments are discussed in section 6 and, finally, section 7 is devoted to the conclusion and further works.

2 Proposed Symbol Representation

As low-level stages and the image itself may introduce error and noise, the encoding of an instance of an object using an attributed graph may differ from an ideal model of this object. Indeed, the resulting graph may appear distorted, have more or less nodes and edges and get different labels. Therefore, the recognition step has to take into account errors to make an object identified as a distorted instance of the involved model.

We propose an exhaustive graph representation combining vectorial primitives and statistical features. A Region Adjacency Graph (RAG) is built using two formalisms for the nodes. A node is associated to a loop and is mapped to a structure consisting of both:

- region boundaries in terms of vectorial primitives,
- region content in terms of moments invariant (Zernike invariant).

To get a rotation and scale invariant representation, the extremities of each vectorial primitive are encoded using relative location taken into account polar coordinates with respect to the centroid, the orientation of the loop and the size of the whole object. Edges involves the shared vectorial primitives and relative properties such as location of the centroid, area and orientation.

One may note that RAG are not suitable to encode some patterns (e.g. in Figure 1(a)). Indeed, vectorial primitives that do not appear in any boundary definition are not reported using this formalism.

All the vectorial primitives, those involved in the region boundaries and the others, are embedded in a graph using 'interconnection' links. Using such a representation enables to define precisely the regions of interest in an unknown digital document where entities are connected in a network. For further developments, we plan to insert additional geometric constraints such as ([5, 6, 7]).

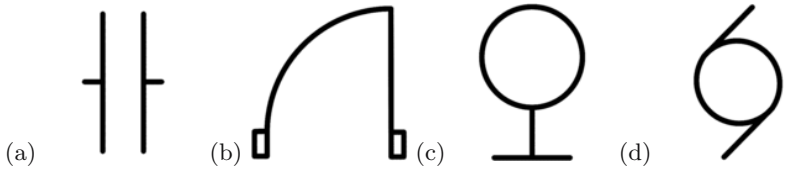


Fig. 1. Some symbols that can not be well defined using RAG with loops as regions (a): Two isolated components without any region (b): Two isolated regions (c-d): two distinct symbols represented by a disc

2.1 Relative Properties

Ignoring labels, a structural representation is clearly independent from any scale, position or orientation. We precise in this section how invariance to an affine transform can be obtained regarding the involved labels. We encode in the edge mapped structure the centroid's location, area and orientation with respect to the source node mapped structure. We detail below how are estimated the affine transform parameters.

Orientation. The orientation ϕ of a shape can be evaluated ($\pm pi$) from the Hu moments as:

$$\phi = \frac{1}{2}\arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right)$$

where

$$m_{p,q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)x^p y^q \mathrm{d}x\mathrm{d}y$$

and $(p,q) \in \mathbb{N}^{+^2}$ Nevertheless, if we want to get a directed orientation estimator, we must consider Zernike moments.

The Zernike moments have complex kernel functions based on Zernike polynomials, and are often defined with respect to a polar coordinate representation of the image intensity function $f(\rho,\theta)$ as:

$$A_{n,l} = \frac{n+1}{\pi}\int_0^{2\pi}\int_0^1 V^{\star}_{n,l}(\rho,\theta)f(\rho,\theta)\rho\mathrm{d}\rho\mathrm{d}\theta$$

where $\rho \leq 1$, the function $V_{n,l}(\rho,\theta)$ denotes a Zernike polynomial of order n and repetition l, and $\star$ denotes complex conjugate. In the above equation n is a non-negative integer, and l is an integer such that $n - |l|$ is even, and $|l| \leq n$. The Zernike polynomials are defined as:

$$V_{n,l}(\rho,\theta) = R_{n,l}(\rho)\exp(il\theta)$$

where $i^2 = -1$, and $R_{n,l}()$ is the real-valued Zernike radial polynomial.

$$R_{n,l}(\rho) = \sum_{s=0}^{(n-|l|)/2} \frac{(-1)^s \rho^{n-2s}(n-s)!}{s!\left(\frac{n+|l|}{2}-s\right)!\left(\frac{n-|l|}{2}-s\right)!}$$

As Teague has demonstrated it, given an image and its rotated image with an angle θ, the Zernike moments of the second image can be expressed using the one of the first image:

$$A^{\phi}_{n,l} = A^{0}_{n,l}\exp(-il\phi)$$

Thus, choosing a suitable (n,l), we can remove the uncertaincy of the orientation estimator when trying to map a node of the model with a node of an unknown pattern testing whether $A^{pattern}_{n,l} = A^{model}_{n,l}\exp(-il\phi)$ or $A^{pattern}_{n,l} = A^{model}_{n,l}\exp(-il(\phi+\pi))$.

We define relative location of target node's centroid with respect to the source node's centroid using:

$$gc_{target} = gc_{source} + d_{source,target} \exp(i\alpha_{source,target} + \alpha_{source})$$

where $\alpha_{source,target}$ denotes the relative orientation of the segment gc_{source} gc_{target} with respect to the orientation α_{source} of the source node and $d_{source,target}$ the distance between gc_{source} and gc_{target}.

Scale. Relative area and distance properties are strongly dependent to scale property. Width and height of the fitting ellipse can be evaluated using the singular values of the covariance matrix (or the Hu Moments up to 2^{nd} order):

$$width = \sqrt{\frac{\mu_{2,0} + \mu_{0,2} + \sqrt{(\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2}}{\mu_{0,0}/2}}$$

$$height = \sqrt{\frac{\mu_{2,0} + \mu_{0,2} - \sqrt{(\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2}}{\mu_{0,0}/2}}$$

At first sight, from a given reference, the scale of a pattern can be approximated using the ratio:

$$scale_{pattern} = \frac{width_{pattern} \times height_{pattern}}{width_{reference} \times height_{reference}}$$

Nevertheless, scale has to be evaluated taking into account variations of the layout thickness since noise can make the layout thicker. From the previous observation, width and height are in fact evaluated using all points within the shape instead of all *white* points within the shape.

2.2 Example

We report below a glimpse of the represented structure of a saving file associated to the symbol of the figure 2. One may reads for example that the segment P0 is interconnected to the arc P7 at pixel (438,193); one of the vectorial paths in the graph - identified PP6 - is made of segments P2, P5, P4, P3; L3 get an internal boundary (PP6) since L3 includes L0; ...

```
<?xml version="1.0"?>
<symbol name="Symbol9">
<list-prim>
<prim id="P0"><line x1="70" y1="193" x2="438" y2="193" thickness="17"/>
</prim>...
<prim id="P7"><arc cx="254" cy="260" r="196" from="69.9919" to="109.083"
thickness="17"/></prim>
</list-prim>
```

<list-link-prim>
<link-prim source="P0" target="P7">x="438" y="193"</link-prim>
...

</list-link-prim>
<list-path-prim>
<path id="PP0">"P0"</path>
<path id="PP1">"P1"</path>
<path id="PP2">"P7"</path>
<path id="PP4">"P9"</path>
<path id="PP6">"P2,P5,P4,P3"</path>
...
</list-path-prim>
<list-loop>

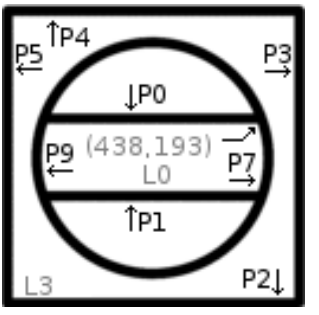


Fig. 2. A symbol and its representation model

<loop id="L0" area="41942" width="212.482" height="65.3871" xc="254" yc="259">
... </loop>...
<loop id="L3" area="95784" width="350.066" height="348.734" xc="258" yc="255">
<orientation angle="1.52869" refAngle="-2.45722" confidence="0.0491451" />
<features name="ANLN"><area size="20">"0.399,0.969,1.790,...,0.000"</area>
</features>
<list-border>
<border id="B0" ref="PP6" kind="outer">"a=223.356 d=0.0104297,a=134.167
d=0.0103775, ..., a=134.167 d=0.0103775,a=43.4713 d=0.0101896"</border>
<border id="B1" ref="PP2" kind="inner">"a=69.4653 d=0.006,a=109.339 d=0.006"
</border> ...
<border id="B4" ref="PP1" kind="inner">"a=109.339 d=0.006,a=248.415 d=0.006"
</border>
</list-border>
</loop>
</list-loop>
<list-link-loop>
<link-loop id="LL4" source="L0" target="L3" r-area="8.786" r-angle="5.498"
r-dist="0.000" border="PP0, PP1,PP2,PP4" />
<link-loop id="LL5" source="L3" target="L0" r-area="0.113" r-angle="0.827"
r-dist="0.000" border="PP0, PP1,PP2,PP4" /> ...
</list-link-loop>
</symbol>

Once having defined such a model, next section shows the system which take advantage of it.

3 Knowledge Operationalization

In the litterature, symbols can be recognized using either a bottom-up or a top-down strategy. The first family of approaches tries to make a symbol appearing among an ascending hiearchy of representations. Thus, the recognition is not guided by any pre-acquired knowledge. In the other hand, the second family of approaches is based on a query, systems try to fit a symbol's model into the data. The matching is then performed verifying the presence of the different components defined in the model. Contextual knowledge (the image to be analysed) and constraints (the candidate model to be found) are used to formulate and then to verify interpretation hypothesis. In practice, recognition systems alternate bottom-up and top-down strategies. The detection of an object enables the system to generate a set of search actions from objects specified within a model in its neighborhood. From this overview, a structural representation of the entity's components seems to be a relevant choice when we can define - for a given domain - a set of symbols to be matched in an unknown document where symbols are embedded in a network.

A recognition hypothesis is triggered identifying a white connected-component ($loop_i$) within an image as a region of a symbol's RAG (see algorithm 1 [1]).

Algorithm 1. [1]Trigger symbol recognition

Input: $Image$ to be analysed
for all $loop_i \in Image$ **do**
  for all $label_j \in \mathcal{L}(loop_i)$ **do**
    $CG(j) \leftarrow \{G_m \in G \mid \exists\ node_m^n \in Node(G_m), label_j \in \mathcal{L}(node_m^n)\}$
    for all $G_k \in CG(j)$ **do**
      for all $node_k^l \in Node(G_k), label_j \in \mathcal{L}(node_k^l)$ **do**
        $RAG\_Matching(G_k, node_k^l, loop_i)$
      end for
    end for
  end for
end for

The interpretation hypothese leads to an affine transform $\lambda()$ and a mapping is created between the seed loop and the node of the candidate RAG model. From

[1] We recall here the involved notations:

- $G = \{G_k\}_{k\in[1,K]}$ is the database of the K symbols models,
- $Node(G_k) = \{node_k^n\}$ and $Edge(G_k) = \{edge_k^e\}$ are respectively the set of nodes and edges of G_k,
- $\mathcal{L}()$ is the label function for a region,
- Λ is the set of parameters of the affine transform function $\lambda()$,
- $X \Leftrightarrow Y$ denotes a valid bijective mapping between X and Y,
- $CG(j)$ is the set of candidate symbols' model, the graphes one node of which has label label_j

edges specified within the model's graph, the next step consists in evaluating whether we can find in the neighborhood of the seed loop regions, possibly adjacent, from which the Λ is nearly constant and relative properties (scale, orientation and location of the centroid) are respected. A hierarchy of regions to be found permits to break the current process as soon as possible giving higher priority in the matching to biggest regions. This strategy is illustrated by the algorithm 2^1: it is a greedy algorithm commanded by candidate models.

Algorithm 2. ^{1}RAG_Matching

Inputs: G_k the candidate model to be found
$seedNode$ a node of G_k
$seedLoop$ a loop such as $\mathcal{L}(seedLoop) \cap \mathcal{L}(seedNode) \neq \emptyset$
$Matching \leftarrow \{seedNode\}$
evaluate Λ from $(seedNode, seedLoop)$
$Node(G) \leftarrow \{seedLoop\}$
repeat
$AN(Matching) \leftarrow \{node_k^l \in Node(G_k) \mid \exists\ edge_m^e \in Edge(G_m),$
$edge_m^e(source) \in Matching, edge_m^e(target) \in Node(G_k) \setminus Matching\}$
$node_k^A \leftarrow \arg\max_{n \in AN(Matching)}\{Area(n)\}$
if $find(loop_i \in Image \mid \mathcal{L}(loop_i) \cap \mathcal{L}(node_k^A) \neq \emptyset, loop_i \Leftrightarrow \lambda(node_k^A))$ **then**
$Matching \leftarrow Matching \cup \{node_k^A\}$
$Node(G) \leftarrow Node(G) \cup \{loop_i\}$
else
return **false**
end if
update Λ
until $G \Leftrightarrow G_k$
return **true**

As the presented work is equivalent to a subgraph isomorphism searh, the interpretation hypothesis among the models $G = \{G_k\}_{k \in [1,K]}$ are initially ordered to avoid any hasty recognition of a G_i instead of a G_j if $G_i \subset G_j$.

4 Learning and Classificition

One of the main problem of the classification phase in pattern recognition relies on the lack of samples for the learning step. Using the method proposed by [18], we can generate noisy images that may have been obtained by operations like printing, photocopying, or scanning processes. For a given model and from a set of images obtained using such an approach, we apply for each loop of the model a mask in order to extract on each pretreated noisy image one Zernike moments' vector corresponding to the involved loop(s). As we do not want to introduce any specific knowledge, each loop of each model lead to a distinct label of the alphabet of nodes. It means that we get initially as many labels as we have loops in the database. Once a database of symbols has been defined,

a Principal Component Analysis (PCA) is performed in order to reduce the feature space dimension with an unsupervised linear feature extraction method. Each cloud is then approximated using a multivariate Gaussian distribution. Finaly, the empirical Bayes decision rule is used for the classification phase. From a validation dataset, we assign a common label to classifiers from their confusion matrix.

5 Proposed Generic Preprocessing Method

Noise can make a recognition method inacurrate. In this section, we present a generic approach we have developped to correct segmentation errors occuring when a loop is fragmented (e.g. on figure 3). Obviously, looking for regularities and singularities on an image are parts of the first interpretation steps of human vision. Concerning technical drawings, one can note that symbols are mainly drawn with a constant thickness. Though, before recognizing a symbol on a real document, we are able to remove part of the noise. Indeed, we can consider that thinner lines may result from a degradation process of the document. The implemented preprocessing method is defined as follow. From a graph representation of the skeleton, we construct a region adjacency graph of cycles $G_C(N_C, E_C)$ where a node defines a geometrical constrained cycle, i.e. a cycle within the skeleton graph enclosing a single loop, and an edge defines a neighborhood relation for which we associate a *brothers* or *child of* label. For each couple of adjacent regions, we extract two features on the depth map along the common connected skeleton branches. Each edge $e_i \in E_C$ is been assigned a minimum depth - denoted as $min_d(e_i)$ - and an extremum depth - denoted as $ext_d(e_i) = min\{depth(firstpoint(e_i), depth(lastpoint(e_i)\}$ - from the endpoints of the frontier. A global mean depth $\overline{d}$ and a standard deviation depth σ_d are evaluated as contextual measures. From $\{min_d(e_i)\}_{e_i \in E_C}$, we decide to remove skeleton parts, i.e. to merge cycles, according to the following rule :

$$IF((min_d(e_i) < ext_d(e_i)/2)\ OR\ (min_d(e_i) < \overline{d} - max\{\overline{d}/2, \sigma_d\}))THEN$$
$$MERGE(e_{i \cdot source}, e_{i \cdot target})$$

The first alternative enables to detect a singular deviation of the thickness along a line (a factor of 1/2 is used as we do not correct displacement of the junction point). The second alternative enables to introduce contextual information since the extremum depth may be itself unreliable. This preprocessing method does not require any parameter and yields good results on technical drawings characterized by a constant thickness. The remaining errors from a recognition point of view - if we plan to use a unique prototype per loop - concern:

- thin cycles that may be split anywhere (see figures 4(b) and 4(c)),
- closed *father and son* cycles for which the father may be split (see figure 4(a)).

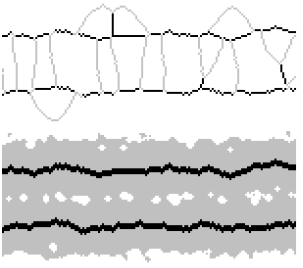


Fig. 3. Skeleton cycles: the initial cycles and the preprocessing results to extract the relevant loop boundaries

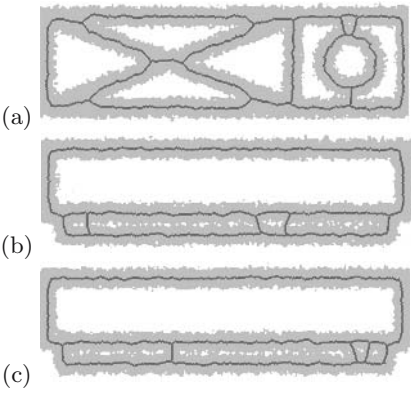


Fig. 4. Remaining segmentation errors after our preprocessing method - (a) Closed *father and son* cycles: the father is split because of the growing child - (b) and (c): Thin cycles can be split anywhere

6 Experimental Results

Experiments have been achieved using the datasets of the GREC 2003 symbol recognition contest to evaluate the scalability of our approach according to the number of symbol models. Evaluation is performed for images with scaling and binary degradation. As our system just take advantage of the RAG to trigger symbol recognition, we do not report result for Level 7-9 which mainly lead to images with thinner lines than original symbol and make region split with one another. For our experiments, we use the pseudo Zernike invariants up to the 6th order on images generated using the first 6 degradation models of the GREC 2003 symbol recognition contest. From 10 images per degradation model, we get 30 images both for learning and validation step. The Principal Component Analysis make the feature space dimension decreasing from 24 to 8 while the study of the confusion matrix make the set of 120 classifiers sharing 82 labels on the third dataset. As pointed out in section (2), a RAG of loop does not allow to represent some symbols. Though, we get a lack of 1, 2 and 3 models respectively in the dataset 1, 2 and 3. Nevertheless, all images specified during the GREC 2003 contest are submited to our recognition process. Moreover, a single model is assigned to some restricted set of symbols (see figure 5(a-c)). The experimental results are summarized in the table 1. One may note that the presented performances are to be compared to 100% for the dataset 1; to 90% for the dataset 2; to 94% for the dataset 3. Finally, as similar symbols are to be distinguished, the results of the presented greedy algorithm are to be compared to 82% for the dataset 3. The reported recognition rates with respect to the previous comments show that the current system is mainly able to reject unknown patterns.

Table 1. Recognition rates (%) for the 3 datasets (each symbol appearing 5 times)

Models	Ideal images	Scaled images	Degraded images					
			Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
5	100	72	96	100	96	72	92	96
20	80	67	85	80	76	71	72	77
50	72	64.4	68.4	67.6	62.4	62.8	52.8	65.2

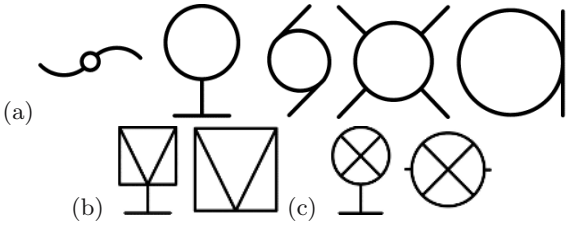


Fig. 5. Sets of symbols the recognition system can not distinguish

7 Conclusion and Future Works

Designing of a symbol recognition system mainly relies on the choice of (i) preprocessing, (ii) data representation and (iii) decision making. In this study, we have focus our attention on the first two points and have proposed an hybrid representation of symbols combining vectorial primitives and statistical features. A Region Adjacency Graph based on loops is firstly built and associated with a graph of vectorial primitives. Therefore, this representation put into relief the definition of regions using two points of view: (i) region boundaries and (ii) region content. For further development, we intend to trigger recognition using either a string edit distance as [19] or a statistical recognition based on moment invariants. Having such a combination will enables to formulate and validate hypothesis with a higher confidence rate while processing an unsegmented document. Moreover, it gives higher discriminative capabilities than RAG. As we select only one model per symbol, we should take advantage of adjacent parts of loops within a model while building a solution. Indeed, since thin shapes can be split even after preprocessing and new search actions scheme have to be introduced. The presented results must be taken as a preliminary study in the way of building an exhaustive representation for symbols.

References

1. Joseph, S., Pridmore, T.: Knowledge-directed interpretation of mechanical engineering drawings. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(9) (1992) 928–940
2. Lladós, J., Valveny, E., Sánchez, G.: A case study of pattern recognition: Symbol recognition in graphic documents. In Ogier, J.M., Trupin, É., eds.: Pattern Recognition in Information Systems, Proceedings of the 3rd International Workshop on Pattern Recognition in Information Systems, ICEIS Press (2003) 1–13

3. Cordella, L.P., Vento, M.: Symbol recognition in documents: a collection of techniques? International Journal on Document Analysis and Recognition **3**(2) (2000) 73–88
4. Valveny, E., Dosch, P.: Symbol recognition contest: A synthesis. In Lladós, J., Kwon, Y.B., eds.: Selected Papers of the 5th IAPR International Workshop on Graphics Recognition. Volume 3088 of Lecture Notes in Computer Science. Springer-Verlag (2004) 368–385
5. Yan, L., Huang, G., Yin, L., Wenyin, L.: A novel constraint-based approach to online graphics recognition. In: Structural, Syntactic, and Statistical Pattern Recognition. Volume 3138 of Lecture Notes in Computer Science. (2004) 104–113
6. Peng, B., Liu, Y., Liu, W., Huang, G.: Sketch recognition based on topological spatial relationship. In: Structural, Syntactic, and Statistical Pattern Recognition. (2004) 434–443
7. Ramel, J.Y., Vincent, N.: Strategy for line drawing understanding. In: Fifth IAPR International Workshop on Graphics Recognition, Barcelona, Spain (2003) 1–12
8. Messmer, B.T., Bunke, H.: A decision tree approach to graph and subgraph isomorphism detection. Pattern Recognition **32** (1999) 1979–1998
9. Lopresti, D.P., Wilfong, G.T.: Evaluating document analysis results via graph probing. In: 6th International Conference on Document Analysis and Recognition. (2001) 116–120
10. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: A (sub)graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(10) (2004)
11. Khotanzad, A., Hong, Y.: Invariant image recognition by zernike moments. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(5) (1990) 489–497
12. Adam, S., Ogier, J., Cariou, C., Gardes, J.: A scale and rotation parameters estimator application to technical document interpretation. In D. Blostein, Y.B.K.E., ed.: Graphics Recognition. Algortihms and Applications: 4th International Workshop, GREC 2001, Kingston, Ontario, Springer-Verlag Heidelberg (2001) 266–272
13. Mukundan, R., Ong, S., Lee, P.: Image analysis by tchebichef moments. IEEE Transactions on image Processing **10**(9) (2001) 1357–1364
14. Ramos, O., Valveny, E.: Radon transform for lineal symbol representation. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition, Edinburgh, Scotland (2003) 195–199
15. Tabbone, S., Wendling, L., Tombre, K.: Matching of graphical symbols in line-drawing images using angular signature information. International Journal on Document Analysis and Recognition **6**(2) (2003) 115–125
16. Chong, C.W., Raveendran, P., Mukundan, R.: Translation and scale invariants of legendre moments. Pattern Recognition **37** (2004) 119–129
17. Yang, S.: Symbol recognition via statistical integration of pixel-level constraint histograms: A new descriptor. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(2) (2005) 278–281
18. Kanungo, T., Haralick, R.M., Baird, H.S., Stuezle, W., Madigan, D.: Document degradation models: Parameter estimation and model validation. IAPR Workshop on Machine Vision Applications (1994) 552–557
19. Lladós, J., Marti, E., Villanueva, J.: Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(10) (2001) 1137–1143

Segmentation and Retrieval of Ancient Graphic Documents

Surapong Uttama, Pierre Loonis, Mathieu Delalandre, and Jean-Marc Ogier

Faculty of Sciences and Technology, L3i Research Laboratory, Université de La Rochelle, Avenue Michel Crépeau, 17042 La Rochelle Cédex 1, France

Abstract. The restoration and preservation of ancient documents is becoming an interesting application in document image analysis. This paper introduces a novel approach aimed at segmenting the graphical part in historical heritage called *lettrine* and extracting its signatures in order to develop a Content-Based Image Retrieval (CBIR) system. The research principle is established on the concept of invariant texture analysis (Co-occurrence and Run-length matrices, Autocorrelation function and Wold decomposition) and signature extraction (Mininum Spanning Tree and Pairwise Geometric Attributes). The experimental results are presented by highlighting difficulties related to the nature of strokes and textures in *lettrine*. The signatures extracted from segmented areas of interest are informative enough to gain a reliable CBIR system.

1 Introduction

The cultural and scientific heritage is the public and unique resource that represents the collective memory of different societies. The international communities (governments, organizations, etc.) have been recognizing an increasing requirement concerning the safeguard and the accessibility of this heritage. This paper deals with a project, called MADONNE[1], aimed at managing various resources of international inheritance especially books, images collections and iconographic documents. These numerous documents contain huge amount of data and decay gradually. One of the goals of MADONNE project is to develop a set of tools allowing to extract all information as automatically as possible from digitized ancient images and to index them in order to develop Content-Based Image Retrieval (CBIR) system. In this paper, we present our contribution concerning the segmentation of ancient graphical drop cap namely *lettrine* in French as well as the extraction of its signatures for constructing CBIR system.

This paper is organized as follows: section 2 gives the definition of lettrine. Section 3 refers to related works in texture analysis and segmentation approaches. Section 4 focuses on lettrine segmentation and section 5 deals with lettrine retrieval. Section 6 demonstrates the experimental results. The conclusion and perspective works are presented in section 7.

[1] MAsse de DOnnées issues de la Numérisation du patrimoiNE, available at http://l3iexp.univ-lr.fr/madonne/

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 88–98, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Definition of Lettrine

From historical point of view, a lettrine (cf Fig.1) is a printed graphic document impressed by a handmade carving wood block. Normally, this block is used during the entire printing process and sometimes reused in other publications. As a result, a wood block is more and more deteriorated after multiple stampings and then results in noticeable changes in details of those printed lettrines.

In view of image analysis, a lettrine is a line-drawing image that is generally handmade in old documents. Usually a lettrine is composed of two main parts: a letter (a first character of the first word in the chapter or section) and a drawing painted in background, dealing with a scene with a semantic or only illustrative motif.

This drawing is specific because it is a line drawing with crosshatch or flat tint, used to draw a scene in order to model its volume. Moreover, the objects drawn in the scene have no closed boundary i.e. the objects are represented by groups of parallel lines, reunited in homogeneous texture zones without exact border. The segmentation of such image is very specific and the standard segmentation tools such as region-based segmentation are not applicable.

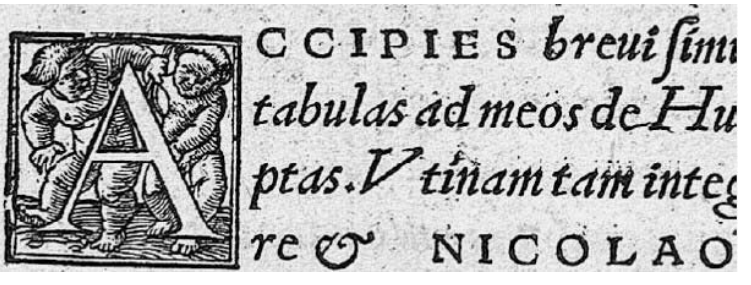


Fig. 1. Lettrine in Context

3 Related Works and Research Strategy

Segmentation of ancient hand-drawing image like lettrine is not widely concerned as one can see from few related works. Normally, from the nature of a lettrine, it contains different strokes and patterns that can be considered as texture. The pattern of strokes in lettrine could be regarded as Oriented-Defined Texture (ODT) and some researchers proposed the segmentation concept called Oriented-Based Texture Segmentation (OBTS) [1]. However, most of the work in segmentation of ODT is conceptual and either experimented on synthetic textures or natural textures normally from Brodatz album [2].

In order to segment lettrine properly, the selection of texture analysis approaches is an essential task. Texture analysis is mainly divided to four directions [3] ; structural, statistical, model-based and transform-based methods and it is reported that a Gray-Level Co-occurrence Matrix (GLCM) [4] is widely used due

to its performance and efficiency. In model-based method, Wold decomposition [5] is an interesting choice to interpret a texture as a combination of multiple signals. In addition, it can be used to model autocorrelation function to describe the fineness and coarseness of texture.

Concerning CBIR, nowadays there exists a lot of image retrieval systems. Unfortunately, there are few literatures relating to the retrieval of ancient graphic document [6, 7].

In this article, we introduce a novel lettrine segmentation method based on different texture analysis approaches. Our proposed method is a combination of GLCM, Run-length matrix [8], Autocorrelation function and Wold decomposition. For lettrine retrieval process, we propose a novel system that indexes lettrines by computing signatures of segmented areas of interest from previous step.

4 Segmentation of Lettrine

Our approach is inspired by the classical segmentation techniques by computing image features on a sliding window at any spatial point of an image i.e. Block Processing Operation(BPO). We divide our segmentation process into two steps: global segmentation and local segmentation.

4.1 Global Segmentation

For a specific image as a lettrine, it is observed that we can roughly partition a lettrine into homogeneous regions (zones with nearly the same contrast without any patterns) and texture regions (zones with strokes or patterns). Therefore, it is not necessary to apply BPO for entire lettrine. In other words, the BPO on homogeneous regions will be redundant because we can use other simple features such as contrast for segmentation. Then the primary task in segmentation process becomes to differentiate between homogeneous and texture regions. To separate two regions from each other, we propose to use GLCM and the average of its uniformity (U) which is defined as:

$$U = \frac{1}{4} \sum_{i=1}^{G} P_{d\theta}(i, i) \tag{1}$$

where $P_{d\theta}(i, i)$ is normalized diagonal members of GLCM with one-unit displacement ($d = 1$) in four directions ($\theta = 0, \pi/4, \pi/2, 3\pi/2$) and G is number of gray level (empirically 4). This process is performed by sliding a small window (empirically 4 by 4) throughout the quantized lettrine (4-bit grey level). In each window, we calculate GLCM as well as its uniformity and store it as a value of center pixel. Finally, after applying binary threshold, the regions with high uniformity will present homogenous zones while low-uniformity regions will refer to texture zones.

4.2 Local Segmentation

After available to partition the homogeneous area and its complement, the next step is to prepare necessary information for segmentation process. Concerning the uniform region, it is not necessary to gain supplementary information. The segmentation of this zone is possible by measuring some parameters such as its contrast. Conversely, for texture region, we need to perform BPO. Therefore we demand more data i.e. how large the block's size should be so that it can present unique texture characteristics and which texture descriptors are suitable for segmentation.

Indeed, the appropriate block's size is very crucial to the segmentation process. For example, if the texture motif is quite large (coarse) but the block's size is too small, we will lose some significant information. In contrast, if the texture motif is quite small (fine) but the block's size is too large, our block may contain multiple texture regions and finally provides bad segmentation results.

To answer this question, we acknowledge that Autocorrelation Function (ACF) can reveal the coarseness and fineness of texture which is related to the suitable block's size. The work in [5] models ACF in one dimension by using Wold decomposition as follow:

$$\widetilde{ACF}(r) = e^{-\alpha r} + \gamma e^{-\beta r} \cos(2\pi f r + \phi) + \delta + \epsilon(r) \tag{2}$$

where $\widetilde{ACF}(r)$ is modeled ACF, and $\alpha, \gamma, \beta, f, \phi$ and δ are model's parameters and $\epsilon(r)$ is error of modelling.

From 2 we observe that the term $\gamma e^{-\beta r} \cos(2\pi f r + \phi)$ represents the periodic behavior of texture. As a result, if a texture motif is periodic (e.g. parallel lines), the unique texture size can be presented by the frequency (f). In turn, if a texture motif is non-periodic or random, the term $\gamma e^{-\beta r} \cos(2\pi f r + \phi)$ should be very small and ACF will be dominated by the term $e^{-\alpha r} + \delta$. Therefore, in this case, the unique texture size should be the value of ACF when δ is nearly zero.

In order to get these parameters (f and δ), it is required to calculate real ACF and optimize with modeled ACF ($\widetilde{ACF}$). Theoretically, real ACF can be computed in space domain from

$$ACF(k,l) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} I(i,j)I(i+k,j+l)}{\sum_{i=1}^{M}\sum_{j=1}^{N} I^2(i,j)} \tag{3}$$

However, in this paper, we calculate real ACF in frequency domain by using inverse of power spectrum of image which is more faster in computation. Nonetheless, the real ACF ($ACF(k,l)$) here is in two dimension. We need to interpolate it to one dimension in order to optimize with 2. The work of [9] suggests to interpolate two-dimensional ACF by using a circle $\hat{C}$ (cf. Fig.2) as follow.

$$ACF(r) = \frac{1}{2\pi r} \sum_{(k,l)\in\hat{C}} \widehat{ACF}(k,l) \tag{4}$$

where,

$$\widehat{ACF}(k,l) = \begin{cases} ACF(k,l) & \text{if } (k,l) \in N \\ \frac{\sum_{i=1}^{4} d_i ACF(k,l)}{\sum_{i=1}^{4} d_i} & \text{otherwise,} \end{cases} \quad (5)$$

d_i is the distance between point (k,l) and surrounding points (P_1 to P_4) and r is the radius of circle $\hat{C}$ used to interpolate ACF.

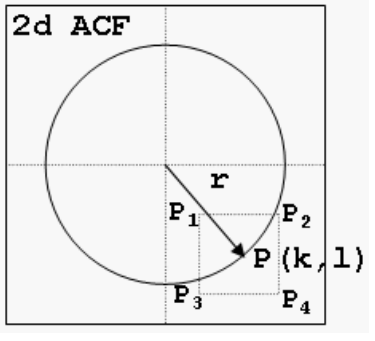


Fig. 2. A circle $\hat{C}$ used for the interpolation of ACF

We can explain the interpolation method from 4 and 5 and Fig. 2. Starting from the two-dimensional ACF, we locate its center. Then construct a circle with radius r ($r \geq 0$). At any point on a perimeter of that circle, if its coordinate is integer the ACF does not change, otherwise interpolate ACF from the four nearest neighbors. Finally find summation of ACF around the perimeter and normalize.

Finally, by optimizing 2 and 4 we can define the block's size of interested texture. The next process is to choose the texture descriptor. Here, we experimentally select eight descriptors from GLCM and three descriptors from Run-length matrix as shown in Table 1.

Table 1. Texture Descriptors for Lettrine Segmentation

GLCM (8 descriptors)	**Run-length matrix** (3 descriptors)
contrast, entropy, correlation, uniformity, sum average, sum variance, information measure of correlation 1 and 2.	long-run emphasis grey-level distribution run-length percentage

The segmentation process starts by initially dividing the entire image into small blocks (empirically 20% of image's size). In each divided block, the ACF is calculated and optimized to identify the sub-block's size. This sub-block is used to compute texture descriptors and its size will be constant in one block

but adaptive for other blocks according to the ACF. This adaptive block's size allows us to obtain more realistic texture descriptors during our segmentation process. After computing texture descriptors, a feature vector of 11 dimensions is created for any spatial pixel of input image. Finally the clustering technique (currently k-means) is implemented to provide final segmentation by assuming the numbers of clusters are known in advance.

5 Lettrine Retrieval

The objective of lettrine retrieval is to permit users to input a query lettrine and retrieve the lettrines which are most similar to the query from the database. In this context, we are interested only content-based retrieval. That means when trying to match query and lettrine in database, we do not concern a letter of lettrine. This is because matching by using a letter is quite plain and simple in this case. To compare lettrines, we derive the benefit from the segmented areas of interest available from the previous section. For each lettrine, the three segmented layers i.e. homogeneous, texture and contour layers are considered. The morphological operation is applied to segmented regions of each layer to reduce noises and very small regions. Then those areas are labeled according to connected components before signature extraction. The overall process of CBIR of lettrine is illustrated in Fig. 3. In this paper, we pick out two types of signature based on Minimum Spanning Tree (MST) and Pairwise Geometric Attributes (PGA).

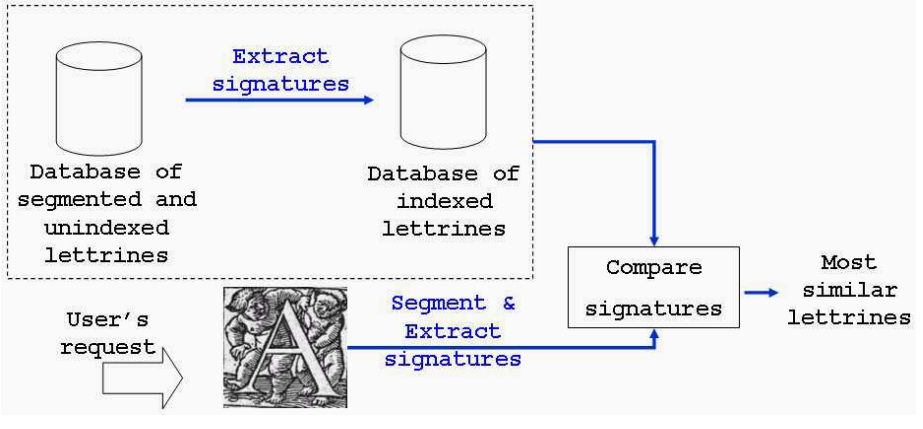


Fig. 3. Lettrine Retrieval System

5.1 MST-Based Signature

A MST-based signature is founded on the spatial organization of segmented areas and is derived from the tree structure of graph. A MST is defined as

a spanning tree that connects all the vertices together with weight less than or equal to the weight of every other spanning tree. Unfortunately, the construction of MST is costly operation. Therefore, in order to reduce its complexity, we try to minimize number of vertices of graph by using centers of gravity to represent labeled regions. By using three segmented layers, thus, each lettrine in database provides a vector of 1×3.

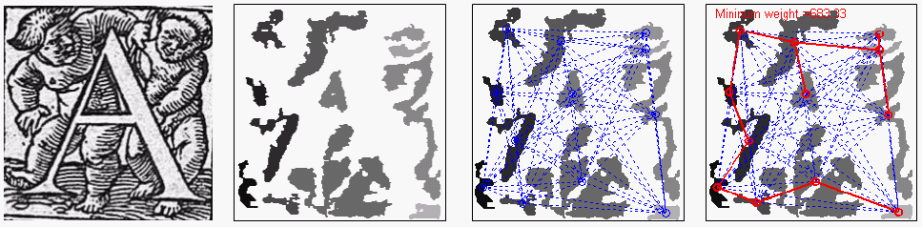


Fig. 4. Lettrine, Segmented regions, Graph and MST

We acknowledge that the extraction of MST-based signature is not a bijective operation that guarantees the unique manner to describe an image. As a consequence, we decide to experiment with another approach, based on a relevant signature namely Pairwise Geometric Attributes(PGA).

5.2 PGA-Based Signature

To avoid a problem found in MST-based signature, we consider PGA which concerns not only the spatial organization but also the shape of segmented regions. In fact, PGA is a combination of relative angle and length between regions. In the original paper proposed by [10], PGA are derived from edges of neighborhood graph of line pattern of image. In our case, PGA are computed from major axis of segmented and labeled regions.

Suppose that x_{ab} and x_{cd} are vectors that represent the major axes of segmented areas of interest (cf Fig. 5). Then the PGA are defined as the relative angle (α) and the length ratio (ϑ):

$$\alpha_{ab,cd} = \arccos\left[\frac{x_{ab} \cdot x_{cd}}{|x_{ab}| \cdot |x_{cd}|}\right] \tag{6}$$

$$\vartheta_{ab,cd} = \frac{1}{\frac{1}{2} + \frac{D_{ib}}{D_{ab}}} \tag{7}$$

To store PGA-based signature of each lettrine, relational histogram is built according to the following conditions:

$$H(I,J) = \begin{cases} H(I,J)+1 & \text{if } \alpha_{ij} \in A_I \text{ and } \vartheta_{ij} \in R_J \\ H(I,J) & \text{otherwise,} \end{cases} \tag{8}$$

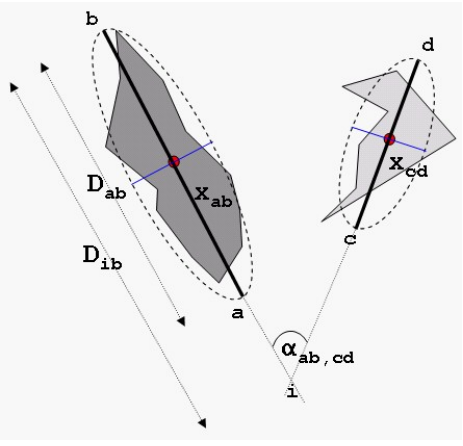


Fig. 5. Pairwise Geometric Attributes between two regions

where A_I is the range of relative angle at I^{th} span (6 bins, $-\pi$ to π) and R_J is the range of length ratio at J^{th} span (6 bins, 0 to ∞, not uniformly distributed). From the fact that we use three segmented layers for a lettrine, therefore, each lettrine will produce three relational histograms.

In order to retrieve lettrines from the database by using query lettrine, we consider employing Bhattacharyya distance. Suppose that H_Q is the normalized histogram for a query lettrine and H_D represents the corresponding normalized histogram contents of one of the histograms contained within the database. The Bhattacharyya distance between two histograms is equal to

$$B(Q, M) = -\ln \sum_{I=1}^{n_A} \sum_{J=1}^{n_R} \sqrt{H_Q(I, J) \times H_D(I, J)} \tag{9}$$

The Bhattacharyya distance uses the correlation between the bin contents as a measure of histogram similarity. This means that the most match pair of images will result in the minimum distance (the most negative).

6 Experimental Result

The experiment was performed by taking into account the lettrines from *Centre d'Etudes Supérieures de la Rennaissance de Tours*[2]. The database of lettrine contains grey-level 344 images but not all distinct. According to the processes introduced in previous section, they provide the outcomes as follow.

6.1 Segmentation Result

In global segmentation, the original lettrine is partitioned by the property of GLCM called uniformity. A sample of lettrine and its partitioned result are

[2] Avialable at http://www.cesr.univ-tours.fr/

shown in Fig. 6. At this point there are obvious two zones; the white region refers to the high uniformity or the homogeneous zone while the black area represents the low uniformity or the texture zone. This image is intended to use as a mask for applying different segmentation techniques for each region. An example of

Fig. 6. A lettrine and its homogeneous and texture regions

segmentation results after local analysis is shown in Fig. 7. From our experiment, it is observed that a lettrine can be segmented to at least three different layers: homogeneous layer, texture layer and contour layer. For texture layer, we try to segment further but it is not effective because our algorithm depends on invariant texture analysis and does not concern the semantic of textured pattern. In addition, the measurement of segmentaion rate is currently suspended because the lack of ground truth of lettrines. We are working through this problem and going to present the results soon. Nevertheless, the three segmented layers are informative enough to use in the next process i.e. image retrieval.

Fig. 7. Segmentation layers of a lettrine in Fig.6

6.2 Retrieval Result

The development of lettrine retrieval system is currently in progress. The system allows user to request the indexed lettrines from the database by comparing to the query image. Users have choices to use MST, PGA or both of them as signatures in retrieval process. An example of preliminary lettrine retrieval system is shown in Fig. 8 By performing experiments on our lettrine retrieval

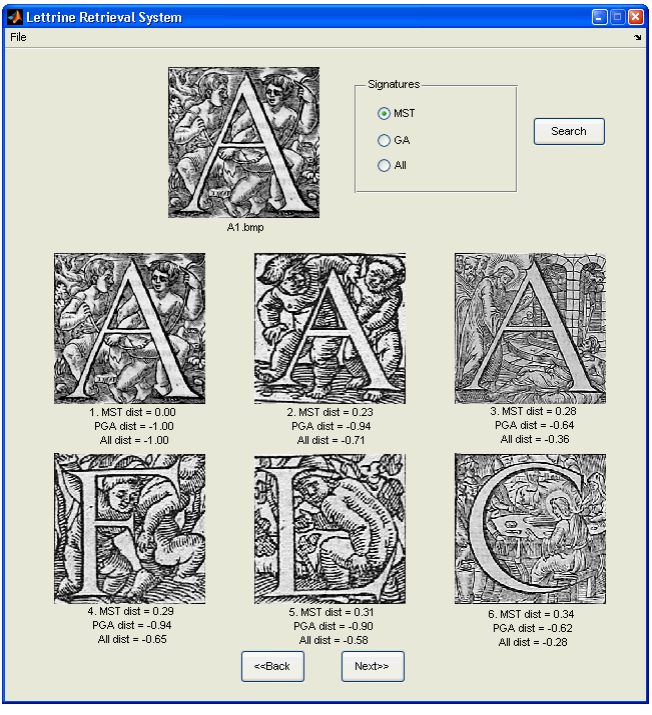


Fig. 8. Preliminary Lettrine Retrieval System

system for multiple times, we found that the first retrieved image is the same as the query image all time. The other retrieved images seem to be relevant to the query image but we need to develop criteria to measure such relevance in order to evaluate our system in terms of precision and recall.

7 Conclusion and Perspectives

Ancient graphics such as lettrine have special and unique characteristic i.e. the oriented-defined texture without definite boundary. In addition, with the dense parallel or crossing lines belonging to texture region, it is hardly possible to implement the normal region-based segmentation method. In this paper, we propose a novel segmentation scheme by adapting the classical segmentation technique such as GLCM and Run-length matrix. We propose an adaptive block's size which is changeable according to the texture's characteristic. The block's size is determined by optimizing the theoretical ACF to the modeled ACF derived from Wold decomposition. The segmentation results are encouraging and informative for further study. Considering lettrine retrieval, we introduce a novel CBIR system that uses signatures extracted from segmentation results in previous stage. The extraction of signatures is performed on three segmentation layers. In this

work, we experiment by using signatures derived from MST and PGA. From our test, the first retrieved image is exactly the same as the query image and the other images seem relevant. The general retrieval system is pleasing.

For prospective works, we try to apply and combine different segmentation algorithms to measure the segmentation performance. We are also working to define the ground truth of lettrine for measuring lettrine segmentation rate. Concerning the retrieval system, we are looking for the algorithms to classify database of lettrines into different classes in order to define their relevances. The results from this process will help us evaluate our CBIR system by determining its recall and precision.

References

1. O. Ben-Shahar and S. W. Zucker, Sensitivity to curvatures in orientation-based texture segmentation, Vision Research, Vol 44(3), 2004, pp. 257-277.
2. P. Brodatz, Texture-A Photographic Album for Artists and Designers, Dover, New York, 1966.
3. M. H. Bharati1, J. J. Liu and J. F. MacGregor, Image texture analysis: methods and comparisons, Chemometrics and Intelligent Laboratory Systems, Vol. 72, 2004, pp.57-71
4. R.M. Haralick, K. Hanmugan, and I. Dinstein, Textural features for image classification, IEEE transactions on Systems, Man, and Cybernetics, vol. 3, 1973, pp. 610-621.
5. R. Sriram, J. M. Francos, and W. A. Pearlman, Texture Coding Using a Wold Decomposition Based Model, IEEE transactions on Image Processing, vol. 5, September 1996, pp. 1382-1386.
6. J. Bigün, S. K. Bhattacharjee, and S. Michel, Orientation Radiograms for Image Retrieval: An Alternative to Segmentation, In Proc. 13th Int. Conf. Pattern Recognition, Vienna, Austria, August 25-30, 1996
7. E. Baudrier, Comparaison d'images binaires reposant sur une mesure locale des dissimilarités Application à la classification, Ph.D. Thesis, UFR des Sciences Exactes et Naturelles, Université de Reims Champagne-Ardenne, 2005, 176 p.
8. X. TANG, Texture Information in Run-Length Matrices, IEEE transactions on image processing, vol. 7, pp. 1602-1609, 1998.
9. C. Rosenberger, Mise en Oeuvre d'un système Adaptative de Segmentation d'Images, Ph.D. Thesis.: Rennes, 1999, 172 p.
10. B. Huet and E.R. Hancock, Line Pattern Retrieval Using Relational Histograms, IEEE transactions on Pattern Analysis and Machine Intelligence, Vol. 21(12), December 1999.

A Method for 2D Bar Code Recognition by Using Rectangle Features to Allocate Vertexes

Yan Heping, Zhiyan Wang, and Sen Guo

School of Computer Science & Engineering, South China University of Technology, Guangzhou 510640, China
gzyhp@126.com, wzhyan@ieee.org

Abstract. This paper describes a method of image processing for the 2D bar code image recognition, which is capable of processing images extremely rapidly and achieving high recognition rate. This method includes three steps. The first step is to find out the four vertexes of ROI (Regions Of Interest); the second is a geometric transform to form an upright image of ROI; the third is to restore a bilevel image of the upright image. This work is distinguished by a key contribution, which is used to find the four vertexes of ROI by using an integrated feature. The integrated feature is composed of simple rectangle features, which are selected by the AdaBoost algorithm. To calculate these simple rectangle features rapidly, the image representation called "Integral Image" is used.

1 Introduction

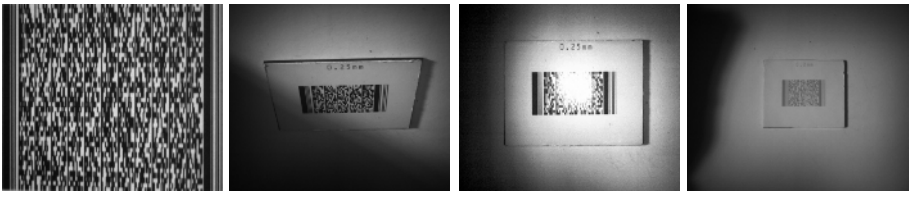
Recognition of the 2D code bar image in normal condition is not a complex problem; It is based on the image processing approach, and is mainly composed of four steps: regions of interest (ROI) detection, code location, code segmentation, decoding [1]. We can get good recognition effect by using normal process methods to get a bilevel image of ROI and then decoding it. But, the recognition rate will be degraded rapidly when the recognition algorithm is embedded in an embedded environment, such as a mobile phone which has a camera. This is because the captured image is not 'normal'; see Fig. 1. The most important reason lead to this problem is the heterogeneity lighting, polarized light, highlight spots and low contradistinction in this image set, see Figure 1(b)-(h). It will lead to bad effect in locating ROI.

The second reason is that the 2D plane of ROI is skewed due to the rotation of three degrees of freedom in the imaging process; see Fig. 1(c)(d). In these cases, the original upright shape (see Fig. 1(a)) is projected into a skew shape. Because of that the coding is based on rectangle units, all of these will influence the recognition rate greatly. Some other reasons, such as noise and speckles, will also influence the recognition rate.

These cases are not popular in general applications. But, when the Corporation wanted to embed the bar code recognition algorithm into their mobile telephone, which have a camera, they found this cases are very popular and the recognition algorithm almost can not work normally; It is because of that the imaging conditions are

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 99 – 107, 2006.
© Springer-Verlag Berlin Heidelberg 2006

almost not 'normal' when we use a mobile phone to capture images. Then, the X Corporation provided an image test set (X set) to call for recognition algorithm. This image set includes almost all abnormal imaging conditions may occur in practical applications. To this test set, the recognition rate of our former recognition algorithm, which works well to normal images, can only reach 37%.



(a) The idea image (c) Skew projection, and polarized light (e) Highlight spot (g) Low contradistinction

(b) Highlight spots (d) Skew projection (f) Low contradistinction and highlight spots (h) Heterogeneity lighting

Fig. 1. 2D bar code images in various imaging condition (encoded in PDF417 format)

To this test set, experiment result shows that it is difficult to locate ROI correctly; and, the threshold image is vary different with its original image even by using complex local threshold algorithms such as Otsu [2].

Popularly, an image representation in term of magnitude and phase of image gradient is used to detect ROI [1]. Other methods such as texture analysis [3] are also used in some complex cases. But these methods do not work well to this X set.

This paper brings forward a method to recognize the 2D bar code image. The first step is to find out the four vertexes of ROI, in which AdaBoost algorithm is used to select the vertex features. The second step is to adjust this quadrangle image into an upright rectangle image by using a geometric translation. The third step is to form a bilevel image of the upright image by restoring the binary value of each rectangle unit, which is based on the priori knowledge of a code bar image. Among of them, the key of this method is to find out the vertexes accurately.

This paper is organized as follows. In Section 2, we introduce the features. In Section 3, we present the feature selection method. In Section 4, we present post-processing briefly, which includes geometric translation and restoring bilevel image. Preliminary results are given in Section 5. A conclusion is given in Section 6.

2 Features

Observing from the sample images, features of the vertexes (here the word 'vertex' means the vertex of the ROI) are obvious. Divide the sub-window centered by a vertex into some rectangle blocks, which are in the same size. We can find that the brightness is uniform within a block; and neighbor blocks are in the uniform brightness or obvious different brightness; see Fig. 2. We can use this kind of features (we call them rectangle features) to find out the vertexes.

But, the vertexes of the code units may have the same features; see red rectangle in Fig. 2. In fact, features in the two kinds of vertexes can be distinguished by the size of the rectangle and the brightness difference between the neighbor rectangles. For example, a coherent feature in one rectangle size between the two kinds of vertexes may be very different in another size.

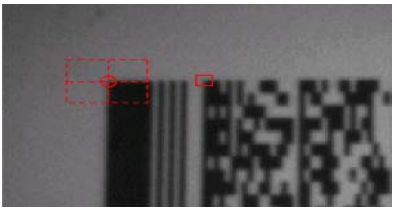


Fig. 2. Features in the sub-window centered by a vertex. Red circle: Vertex of the valid code bar area; Red rectangle: Vertex of a code unit.

But, it is not easy to find out these features that can distinguish the two kinds of vertexes by using the observing method. There is too much this kind of simple features due to the change of rectangle size. Mainly for this reason, we use the AdaBoost algorithm [4] to select effective features from the feature set. In our method, the feature set includes three kinds of features; see Fig.3.

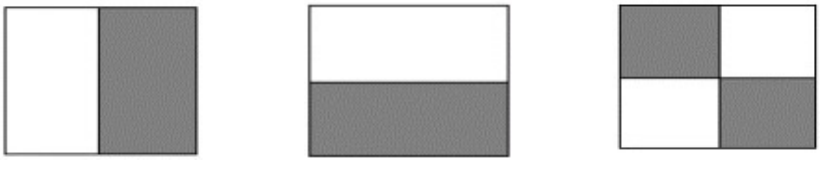


Fig. 3. Three kinds of rectangle features. The sums of the pixels, which lie within the white rectangles, are subtracted from the sum of pixels in the gray rectangles. Two-rectangle features are shown in (a) and (b). And (c) shows a four-rectangle feature.

2.1 Integral Image

Rectangle features can be computed very rapidly by using an intermediate representation for the image, which is called the integral image [4]. The integral image at location *x, y* contains the sum of the pixels above and to the left of *x, y*, inclusive:

$$ii(x, y) = \sum_{x' \le x, y' \le y} i(x', y') \tag{1}$$

Where *ii(x,y)* is the integral image and *i(x,y)* is the original image. Using the following pair of recurrences:

$$\begin{aligned} s(x, y) &= s(x, y-1) + i(x, y) \\ ii(x, y) &= ii(x-1, y) + s(x, y) \end{aligned} \tag{2}$$

Where *s(x,y)* is the cumulative row sum, *s(x,-1)=0*, and *ii(-1,y)=0*, the integral image can be computed in one pass over the original image. Using the integral image any rectangle sum can be computed in four references; see Fig. 4.

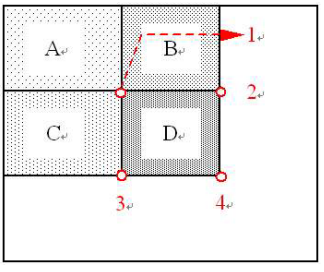


Fig. 4. The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A+B, at location 3 is A+C, and at location 4 is A+B+C+D. The sum within D can be computed as 4+1-(2+3).

All example sub-windows used for training should be variance normalized to minimize the effect of different lighting conditions. Normalization is therefore necessary during detection as well. The variance of an image sub-window can be computed quickly using a pair of integral images.

Recall that $\sigma^2 = m^2 - \frac{1}{N}\sum x^2$, where σ is the standard deviation, m is the mean, and x is the pixel value within the sub-window. The mean of a sub-window can be computed using the integral image. The sum of squared pixels is computed using an integral image of the image squared (i.e. two integral images are used in the scanning process). During scanning the effect of image normalization can be achieved by post-multiplying the feature values rather than pre-multiplying the pixels.

2.2 Feature Discussion

The features in the vertexes are also obvious in the projection curves of the horizontal and vertical directions; see Fig 5. The eight points marked by '+', which are obvious in the curves, are the eight coordinate value of the four vertexes. But, because of the existence of noise especially the highlight spots, which are marked by 'O' in these

curves, it is difficult to allocate the eight points. One method to resolve this problem is to remove the highlight spots. To do this, we should detect the highlight spots and then compensate all the pixels within them. Obviously, it is not easy and especially much time consumption.

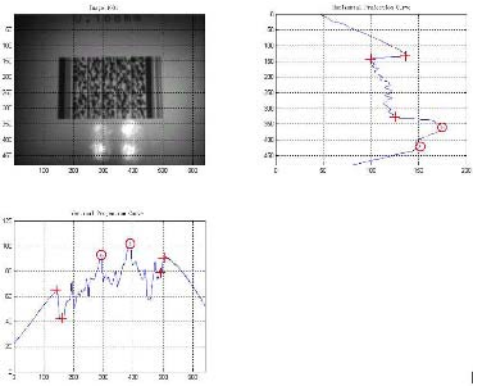


Fig. 5. The horizontal and vertical projection curve. '+': Vertex coordinate; 'O': highlight spots.

3 Feature Selection

There are too much features in the feature set described above. But only a little part of them can distinguish the vertexes. The challenge is to find out them. Paul Viola and Michael Jones [4] have used the AdaBoost algorithm to search important human face features from millions of such features.

AdaBoost algorithm is a kind of classifier algorithm, which brought forward by Yoav Freund and Robert E.Schapire [5] in 1995. Its fundamental idea is to build a strong classifier by boosting a mount of simple classifier, which have general class ability, called week classifier. Freund and Schapire proved that the training error of the strong classifier approaches zero exponentially in the number of rounds. More importantly a number of results were later proved about generalization performance. The key insight is that generalization performance is related to the margin of the examples, and that AdaBoost achieves large margins rapidly. Paul Viola and Michael Jones use this method to detect face objects in sequence images based on the simple rectangle features reminiscent of Haar basis function, which have been used by Papageorgiou et al [6]. The key idea in their method is to select important features from a large feature set. They made success in detection rate and in real-time usage.

We use this idea to select the rectangle features that best separate the positive and negative examples (the vertex and the other). The weak classifier $h_j(x)$ consists of a feature f_j, a threshold θ_j and a polarity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1 & if \ \ p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Here x is the positive or negative sample, and $f_j(x)$ is feature value of the jth rectangle feature. See Algorithm 1 for a summary of the boosting feature selection process.

Algorithm 1. The boosting feature selection algorithm

- Give example images $(x_1, y_1), \ldots (x_n, y_n)$ where y_i=0,1 for negative and positive examples respectively.
- Initialize weights $w_{1,i}=1/2m, 1/2l$ for y_i=0,1 respectively, where m and l are the number of negatives and positives respectively.
- For $t=1,\ldots,T$:

  1 Normalize the weight, $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$, So that w_i is a probability distribution.

  2 For each feature, j, train a classifier h_j which is restricted to use a single feature. The error is evaluated with respect to w_t, $\varepsilon_j=\sum_i w_i|h_j(x_i)-y_i|$.

  3 Choose the classifier, h_t, with the lowest error ε_t, which is corresponding to the feature f_j.

  4 Update the weights: $w_{t+1,i}=w_{t,i}\beta_t^{1-\varepsilon_i}$, where $\varepsilon_i = 0$ if example x_i is classified correctly, ε_i=1 otherwise, and $\beta_t=\varepsilon_t/(1-\varepsilon_t)$
- Output the T most important features: $f_{j1}, \ldots, f_{jT}$.

In this algorithm, a training round select a single feature, f_j. After this round, the parameter θ_j and p_j are adjusted to an optimized value.

After selected the most important features by the boosting algorithm, we do not use the strong classifier to detect vertexes in the scanning process. It is because of that we can combine the features into an integrated feature to search vertexes:

$$f_{Int} = \sum_{i=1}^{T} C_i f_{ji} \tag{4}$$

Here T and f_{ji} are come from the AdaBoost module, and C_i is weight determined by the feature type. We just need to find out the pixels with the maximum feature value in a demisemi image. It is more effective than do it by a strong classifier; and the strong classifier method may find out more vertexes than four.

Due to different importance of these features, the weights $C_i(i=1,...,T)$ are not equal. If the f_i is more important, then c_i is bigger. In our experiments, we used the first 5 most important features to search vertexes. We select c_i by the following rules:

$C_1+C_2+\ldots+C_T=1$

$C_1>C_2>\ldots>C_T>0$

In our experiments, $T=5$ and $[C_1\ C_2\ C_3\ C_4\ C_5]$=[0.5 0.2 0.1 0.1 0.1].

4 Post-processing

4.1 Geometry Adjusting of ROI

After the four vertexes are found out, we can calculate the geometric translation according to the four pairs of vertexes; the other four vertexes of the output rectangle image are defined by us reasonably. This geometric transform can be approximated by a bilinear transform for which four pairs of corresponding points are sufficient to find the transformation coefficients [7]:

$$\begin{aligned} x' &= a_0 + a_1 x + a_2 y + a_3 xy \\ y' &= b_0 + b_1 x + b_2 y + b_3 xy \end{aligned} \tag{5}$$

By using this transformation, we can map all pixels into an upright rectangle image. The brightness of pixel in the target image, which has no mapping, is usually computed as an interpolation of the brightness of several pixels in its neighborhood. Fig.6. shows an example.

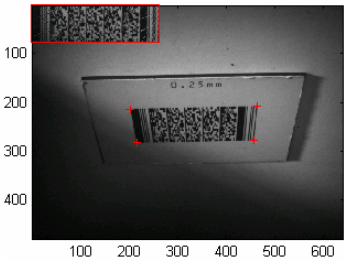


Fig. 6. The vertexes and geometric transform of ROI Red '+': vertexes of ROI; Red rectangle: adjusted area;

4.2 Restore Bilevel Image

To get the bilevel image of the adjusted upright rectangle image, we first divide the upright image into sub code units, according to the prior knowledge of coding. Then the binary value of any unit can be determined by its three preceding units, the left-top unit, the left unit and the top unit; see Fig. 7.

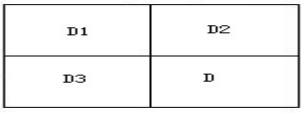


Fig. 7. Threshold of unit D according to D1, D2 and D3

To any unit D, we set its value same as the unit (or units), which is most consistent with it. The consistency is determined by the difference of two unit sums, which also calculated by the integral image. For the first unit, D_i=0(i=*1,2,3*); For units in the first

row, D_2=0; And for units in the first column, D_3=0. We can restore the binary value of all the upright rectangle units by one pass.

At last, we send this binary image into the decoding module and get the recognition result.

5 Results

This research is based on the X set, which includes 406 images of various cases. We divided them into two sets evenly, a training set and a test set. To the training sample set, we label the vertexes and segment the positive samples, which is in 40×40 pixels. The negative samples are also segmented from them and not include any vertexes, but may include the code unit vertexes, also in 40×40 pixels.

The experiments proved that the first five features selected by the AdaBoost module could perform satisfying effect. To the test set, this method can find out vertexes correctly to 194 samples, and the last recognition rates can reach 74%.

In the other 19 samples, which cannot find out the vertexes correctly, four samples don't include vertexes at all. And the other 15 samples have a too low contrast between vertexes and the vertexes of the code units. It seems that the variance normalization method is not strong enough to minimize the effect of the different lighting conditions. More research work should be done in this problem.

The main reason, which affects the last recognition rate, is the existence of large block highlight spots; see Fig. 1(e). In these circumstances, the great mass of information in the code area is destroyed.

Due to the integral image method, the time consumption of all the three process procedure is all $O(N)$ (N is the number of pixels in the original image). So the whole time consumption is $O(N)$.

6 Conclusion

A method for the 2D bar code image process for recognition is brought forward in this paper. The key of this method is to find out the four vertexes by using features, which are selected by the AdaBoost algorithm. Due to the using of the integral image, it is capable of processing images extremely rapidly; it is very important in an embedded application environment. The experiments on the test set have proved its validity.

Acknowledgment

This research work is supported by GuangDong Provincial Natural Science Funding Project B6-109-497.

References

1. Ouaviani, E., Pavan, A., Bottazzi, M., Brunelli, E., Caselli, F., Guerrero, M.: A Common Image Processing Framework for 2D Barcode Reading. Proc. of the Seventh International Conference on Image Processing and Its Applications, 1999 (Conf. Publ. No. 465), Volume 2, 13-15, pp. 652-655.

2. Otsu N.: A Threshold Selection Method from Gray Level Histogram. IEEE Transactions on SMC, 1979(9).
3. Normand, N., Viard-Gaudin, C.: A Two-Dimensional Bar Code Reader. Proceedings of the 12th International Conference on Pattern Recognition, 1994, Vol. 3 (Signal Processing), pp. 201-203.
4. Viola, P. Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. Proc. CVPR, 2001.
5. Freund, Y. and Schapire, R.E.: A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. Journal of Computer and System Sciences, 1997.
6. Papageorgious, C., Oren, M., and Poggio, T.: A General Framework for Object Detection. Proceedings of International Conference on Computer Vision, 1998.
7. Sonka, M., Hlavac, V. and Boyle R.: Image Processing, Analysis, and Machine Vision (2nd Edition), Brochs/Colepp, 62-65, 2000.

Region-Based Pattern Generation Scheme for DMD Based Maskless Lithography

Manseung Seo[1], Jaesung Song[1], and Changgeun An[2]

[1] Tongmyong University, Busan 608-711, Korea
{sms, jssong}@tu.ac.kr
[2] LG Electronics, Gyunggido 451-713, Korea
cgan@lge.com

Abstract. We focus our attention on complex lithographic pattern generation on a huge substrate with no manipulation of the light source shape for Digital Micromirror Device (DMD) based maskless lithography. To overcome the limitations of existing pattern generation methods developed upon the assessment of lithographic paths of the reflected beam spots rather than the recognition of patterns, we place our primary concern on the pattern. We consider pattern generation for maskless lithography using the DMD as a graphic recognition field problem. The pattern generation process is conceptualized as dual pattern recognition in two contrary views, which are the substrate's view and the DMD's view. For pattern recognition in the DMD's view, a unique criterion, the area ratio, is devised for approval of the on/off reflection of the DMD mirror. The Region-based Pattern Generation (RPG) scheme based upon the area ratio is proposed. For verification, a prototype RPG system is implemented, and lithography using the system is performed to fabricate an actual Flat Panel Display (FPD) glass. The results verify that the RPG scheme is robust enough to generate lithographic patterns in any possible lithographic configuration and the RPG system is precise enough to attain the lithographic quality required by the FPD manufacturer.

1 Introduction

Lithography using a light, *i.e.*, photolithography which means light-stone-writing in Greek, was adopted into semiconductor fabrication to transfer geometric shapes of circuit wires onto the surface of a silicon wafer coated with a light sensitive polymer called a photoresist. In conventional lithography for semiconductor fabrication, the main processes are: photoresist coating and prebaking, exposure through an aligned mask, developing, etching, and photoresist stripping. The major lithography equipment may be enumerated as: an optical device including a light source for exposing/writing, a mask for transferring geometric shapes of the pattern, a stage for substrate/wafer moving, and so on. Conventional lithography using masks has been the workhorse till now, in spite of problems caused by masks such as expense and time in fabricating the masks, contamination by masks, disposal of masks, and the alignment of masks.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 108–119, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Recently, research of maskless lithography was initiated and it is growing rapidly upon innovation in digital light processing technology using the Digital Micromirror Device (DMD) by Texas Instruments Inc. (TI) [1]. The DMD appears to be the most successful Micro Electronic Mechanical System (MEMS) solution in the field of microdisplays, and especially for semiconductor lithography [2], [3]. Nowadays, many new DMD application fields have emerged. One of them is lithography for Flat Panel Display (FPD) fabrication [4]. In comparison with other maskless lithographic technologies, the DMD based maskless lithographic technology possesses superior features. These may be enumerated as sufficient throughput for highly customized patterns, higher but precise resolution, fine lithographic quality, efficiency in cost and time, and so on. However, these are feasible if and only if each system developer could set up an excellent optic unit and an accurate lithographic pattern generation unit.

Several lithographic pattern generation methods for DMD based maskless lithography can be found in the literature [5], [6], [7]. But, those were limited to the generation of typically structured patterns being composed of lines rather than arcs. Those have faltered in the generation of unusual patterns with the square shaped light beam such as the one from a mirror pixel in DMD. Besides the fact that their methods require extra optic devices for grating, the application fields may be limited to small sized patterns, such as semiconductor and printed circuit board patterns.

Due to the fast growth of the FPD market, the size of FPD panels has increased in dimension to being 2 m by 2 m and over, and the complexity of the lithographic pattern structure has also increased. Therefore, it may be impossible to generate a lithographic pattern for FPD panels using pattern generation schemes available at this moment. The failure of pre-existing methods in generating unusual lithographic patterns with the square shaped light beam is expected, since the methods were developed upon the assessment of lithographic paths of the reflected beam spots, rather than the recognition of patterns, under the familiarity with lithography using masks. Those never worked whenever an unpredictable pattern appeared.

In this study, we focus our attention on pattern recognition. Then, we devised a method for generating a complex pattern on a huge substrate with no manipulation of the light beam shape. The Region-based Pattern Generation (RPG) scheme is proposed. To verify the RPG scheme, a prototype RPG system is implemented and actual FPD panels are fabricated using the system. The results prove that the RPG scheme is robust and flexible enough to generate a lithographic pattern in any possible lithographic configuration and the RPG system is precise enough to attain the lithographic quality required by FPD manufacturers.

2 DMD Based Maskless Lithography

In the DMD based maskless lithography system, the micromirror array works as a virtual mask to write patterns directly onto FPD glass substrates. To generate

lithographic patterns, millions of micromirrors in the DMD need to be addressed and adjusted, individually and instantaneously. The task of providing a stream of lithographic pattern signals proper to the relative movement of the substrate for the DMD controller is neither simple nor easy.

A diagrammatic lithographic pattern generation by conventional lithography using an ordinary mask is shown in Fig.1(a) and one by maskless lithography using micromirrors in the DMD is shown in Fig.1(b). In conventional lithography, the light beam penetrates through openings in the mask where the patterns were engraved, and then it is irradiated onto the photoresist coated substrate to expose pattern regions identical to the openings. Thus, throughout conventional lithography using a mask, the pattern recognition is performed only once during mask fabrication.

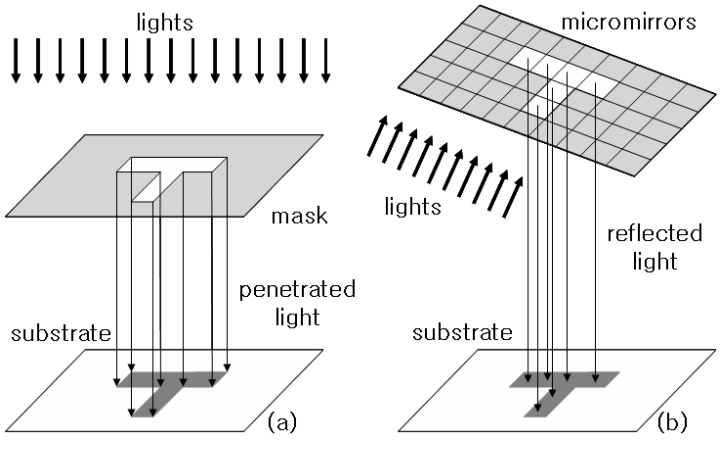


Fig. 1. Diagrammatic lithographic pattern generation

In maskless lithography, the light beam is irradiated onto micromirrors in the DMD, and it is divided into millions of square-shaped beams corresponding to every mirror. For each and all mirrors, on/off reflection of each beam is selected upon the pattern. The proper beams reflected off the selected mirrors are irradiated onto the photoresist coated substrate to expose pattern regions. The micromirror may not recognize the pattern as we would. Moreover, it is not easy to construct pattern regions using millions of square-shaped beams on a moving substrate. Thus, throughout maskless lithography using the DMD, the binary pattern recognition as an on/off voltage is needed in the micromirror's view, instantaneously and properly to the substrate scrolling, along with the conventional pattern recognition as we would.

The major difference between conventional lithography using a mask and maskless lithography using micromirrors is the way of pattern transfer. In conventional lithography, patterns are transferred through the mask made by recognition of vector patterns once. In the lithography in concern, patterns are transferred through the recognition of vector patterns, the conversion of vector

patterns into raster patterns, and the recognition of raster patterns, instantaneously, and over and over millions of times. The other processes not much differ from each other.

Concerning the pattern primarily, we conceptualize the lithographic pattern generation process as dual pattern recognition in two contrary views. The first pattern recognition is considered to be the one performed in the substrate's view, viewing the pattern as we would. The second pattern recognition is considered to be the one performed in the DMD mirror's view, viewing the pattern as a machine would. The distinction between a sector and a square is clear in the substrate's view but it is not in the DMD's view. Thus, we look for a criterion by which the DMD can recognize the pattern as accurately as we do. We devise a unique criterion for the conversion of vector patterns recognized in the substrate's view into raster patterns recognized in the DMD's view. In other words, on/off reflection off micromirrors in the DMD is determined by the area ratio, the area occupied by the pattern per unit DMD mirror. Eventually, in this study, pattern generation for maskless lithography using the DMD is considered as a graphic recognition field problem, i.e., dual pattern recognition in two contrary views upon the area ratio.

3 DMD Based Maskless Lithography Equipment

In order to transfer patterns without a mask, DMD based maskless lithography equipment consists of three major devices. The first is the radiation device for radiating a light including a light source and lenses. The second is the exposure device for irradiating a light and transferring patterns including the DMD, the DMD controller, focusing optics, photoresist coated glass substrates, and the base stage assembly and its controller. The last is the dynamic pattern control device for instantaneous recognition of raster patterns in the mirror's view that is proper to the irradiated energy and substrate moving, and it is composed of the lithographic pattern generation system, the radiation control unit, and the stage control unit.

A schematic diagram of the DMD based maskless lithography equipment is shown in Fig.2. The eXtended Graphic Array (XGA) 1024×768 DMD manufactured by TI has $13.68\mu m$ of mirror (pixel) pitch and it is enlarged to $30\mu m$ of the Field Of View (FOV) in the present work. As shown in Fig.2, micromirrors of the DMD are exposed to incoming radiation released from the ultraviolet light source. The reflection off the micromirrors is determined upon the signal from the lithographic pattern generation system to the DMD controller. Then, the light reflected off the micromirrors is projected through focusing optics onto the photoresist coated glass substrate laid on the x-y scrolling base stage.

Throughout the DMD based maskless lithography in concern, all the DMD controller does is the digital control of the light reflection off the micromirrors, *i.e.*, it gives the approval of reflection as on or off. Therefore, the operation of maskless lithography equipment might be thought of as simple. However, in reality, it is not. Recognizing vector patterns, converting vector patterns into raster

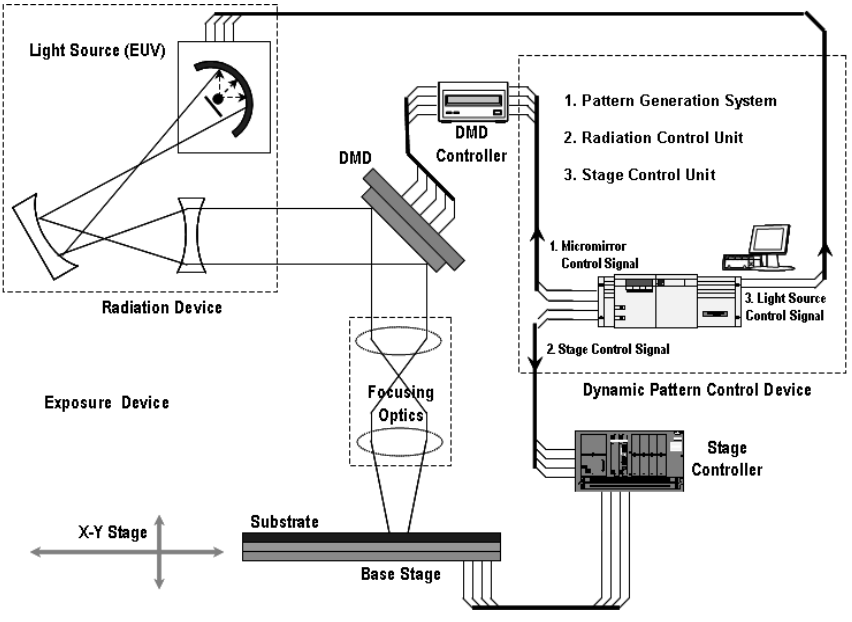


Fig. 2. DMD based maskless lithography equipment

patterns proper for each and every micromirror, and sending the raster patterns to the DMD controller for the approval of on/off reflection off micromirrors is more complicated than thought. These are especially harder when reflection by a rotated DMD frame is unavoidable. It is even more troublesome to instantaneously construct the proper lithographic region, when projection onto a scrolling object is required.

Unfortunately through the maskless lithography in concern, reflection by rotated DMD frame and projection onto a scrolling object are both imposed to satisfy the specifications required by the FPD manufacturer, such as the conservation of the line center. To improve lithographic quality, the DMD frame is rotated counterclockwise at a small angle relative to the longitudinal axis assigned as substrate scrolling direction.

Therefore, devising a customized pattern generation system, entirely from the recognition of vector patterns up to delivering raster patterns to the DMD controller, under constraints such as DMD rotation and substrate scrolling is inevitable.

4 Lithographic Pattern Generation

In this section, the region-based lithographic pattern generation process, which consists of four major procedures, is discussed following the pattern generation flow shown in Fig.3 in conjunction with the illustrated pattern generation process upon the real FPD pattern data shown in Fig.4.

4.1 Pattern Data Loading

The first procedure is loading the Computer Aided Design (CAD) data written in Drawing eXchange Format (DXF). Through parsing of the real vector pattern shown in Fig.4(a), geometric entities are considered to be lines, arcs, and circles. Then, each of the parsed geometric entities is reconstructed as an abstract entity for a polygonal region.

4.2 Recognition of Pattern upon Substrate

The second procedure is recognition of the pattern in the substrate's view. The boundary of the pattern is extracted by the conversion of geometric entities with open loops into polygonal entities forming closed loops. The boundary of the pattern extracted from the DXF formatted CAD data in Fig.4(a) is shown in Fig.4(b).

The construction of the region-based pattern proceeds as follows. Each enclosed area extracted from the pattern boundary is considered as a polygonal entity. In the specific case of the annulus shown in Fig.4(b), the polygons A_1 and A_2 are extracted from the pattern boundaries B_1 and B_2. To manipulate overlapping polygons, set operations on polygons upon computational geometry are performed. For this specific example, set operations on polygons may be written as:

$$\begin{aligned} P_{12} &= (A_1 - B_1) + (A_2 - B_2) - 2\,(A_1 - B_1) \bigcap (A_2 - B_2) + B_1 + B_2 \\ A_1 &= \left\{ x, y | (x - x_1)^2 + (y - y_1)^2 \le {R_1}^2 \right\} \\ A_2 &= \left\{ x, y | (x - x_2)^2 + (y - y_2)^2 \le {R_2}^2 \right\} \\ B_1 &= \left\{ x, y | (x - x_1)^2 + (y - y_1)^2 = {R_1}^2 \right\} \\ B_2 &= \left\{ x, y | (x - x_2)^2 + (y - y_2)^2 = {R_2}^2 \right\} \end{aligned} \tag{1}$$

where, R_1 and R_2 are the radii of the circle A_1 and A_2, and (x_1, y_1) and (x_2, y_2) are the centers of the circle A_1 and A_2. Then, the valid interior regions such as P_{12} in Fig.4(c) are kept as the pattern region for lithography.

4.3 Recognition of Pattern upon DMD

The third procedure is recognition of the pattern in the DMD's view. As mentioned earlier, the DMD frame is rotated counterclockwise at a small angle relative to the longitudinal axis that is assigned as substrate scrolling direction. To account for the counterclockwise rotation of the DMD frame, the pattern region is considered to be rotated clockwise from the longitudinal axis. Then, the coordinate transformation for pattern recognition upon DMD's view, used in the present study, relevant to DMD rotation and substrate misalignment may be written as:

$$\begin{bmatrix} {z_1}^* \\ {z_2}^* \end{bmatrix} = \begin{bmatrix} \cos\theta^* & \sin\theta^* \\ -\sin\theta^* & \cos\theta^* \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} -z_{10}\cos\theta^* & -z_{20}\sin\theta^* \\ z_{10}\sin\theta^* & -z_{20}\cos\theta^* \end{bmatrix} \tag{2}$$

where, $\boldsymbol{z}$ is the reference coordinate vector upon CAD data loading, $\boldsymbol{z}^*$ is the floating coordinate vector relevant to micromirror rotation and substrate misalignment, (z_{10}, z_{20}) is the reference coordinate of the floating origin relevant to micromirror rotation and substrate scrolling, and θ^* is the floating angle which is the sum of the micromirror rotational angle and the substrate misalignment angle.

The rotated pattern region is then projected onto the DMD frame. The part of the region mapped onto the DMD frame array is extracted from the pattern region. By rotating the extracted part of the pattern region counterclockwise back to its original position, the region under the DMD frame is confirmed as the instantaneous pattern region upon DMD rotation at each scrolling phase as shown in Fig.4(d).

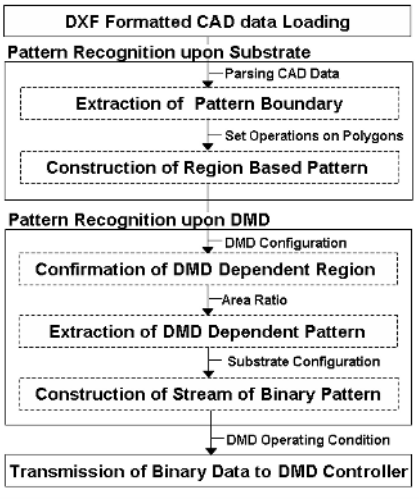


Fig. 3. Pattern generation flow

The distinction between a sector and a square is clear in the substrate's view but it may not be in the DMD's view. We need a criterion by which the pattern recognition in the DMD's view for the on/off determination for the mirrors reflection is accomplished properly. A unique criterion to recognize raster patterns properly in the DMD's view for the approval of the on/off reflection off the DMD mirror is devised to be the area ratio, the area occupied by the pattern per unit DMD mirror. The on/off reflection for each mirror is determined by comparing the area occupied by the pattern per unit DMD mirror with the user specified area ratio. In case when the user specified area ratio is assigned as 0.5, the result of the on or off reflection is shown in Fig.4(e). Micromirrors selected for on reflection are shown as the shaded mirrors, ones for off are shown as the clear mirrors. The regions constructed by the mirrors selected for on reflection may be considered as raster patterns recognized in the DMD's view, as shown

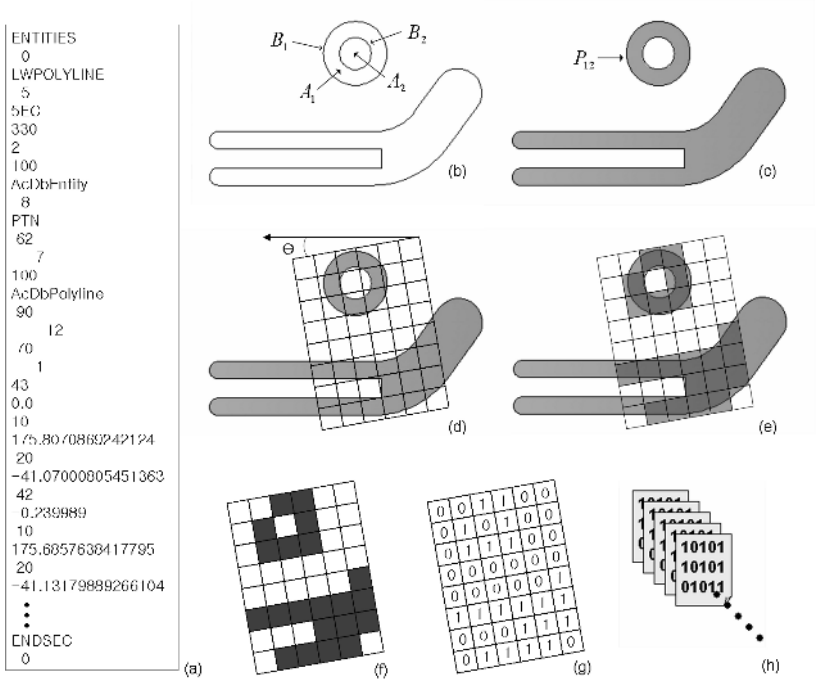


Fig. 4. Pattern generation process

in Fig.4(f). Then, the raster patterns are converted into binary data as shown in Fig.4(g).

The construction of the stream of the binary data in accordance with the substrate configuration is not hard in the proposed method, since the on/off reflection information is contained in the pattern. In the sequence of substrate scrolling step, the stream of binary data is generated accumulating the on/off reflection information contained in the pattern at every substrate location, as shown in Fig.4(h).

4.4 Transmission of Binary Data to DMD Controller

The last procedure is the transmission of the binary coded pattern to the DMD controller in accordance with DMD performance. In this study, an electronic board (DMD Discovery) with a data transit speed of 2500 frames per second is selected to play the role of deliverer.

5 Implementation

To attain our goal of generating a region-based pattern feasible for an actual FPD lithography using the RPG scheme, a prototype RPG system is implemented.

The implemented system is mainly composed of the lithographic pattern generation module discussed in section 4, the signal interchange module that handles the real time communication with the hardware components of the radiation control unit and the stage control unit, and the Graphical User Interface (GUI) that enables the lithography equipment operator to view and control various operations of the system.

The main window of the GUI for the lithographic pattern generation system is shown in Fig.5. The management toolbar, the exposure control sub-window, the pattern display sub-window, and the processing message sub-window are shown on the top, on the left, on the right middle, and on the right bottom, respectively. As shown in Fig.5, the user specified inputs to the implemented system are the origin of the reference/floating coordination system, angle of DMD rotation, angle of substrate misalignment, two- directional DMD resolution, exposure accuracy or pitch upon scrolling step, area ratio for the DMD dependent pattern extraction, and the selection of normal/flip/mirror conversion of the CAD data. Therefore, the system is robust enough to handle any possible user specified mandate and even substrate misalignment.

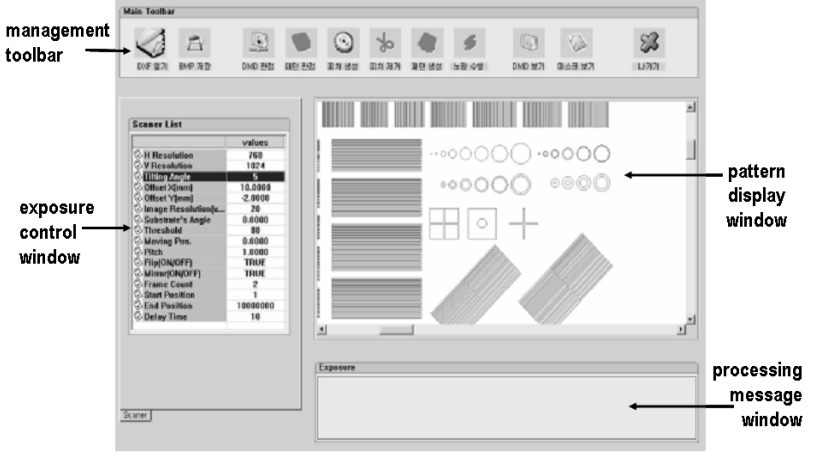


Fig. 5. Lithographic pattern generation system

To examine the capabilities of the devised system, the system is then applied to the generation of a lithographic pattern region for FPD glass fabrication. The FPD glass lithography process upon our system is shown in Fig.6. Throughout the FPD pattern generation, the pitch is assigned to be 5 with $30\mu m$ of enlarged FOV to have irradiation at every $6\mu m$ scrolling. The area ratio for the DMD dependent pattern extraction is held at 0.8. A total number of 16000 frames requiring exposure are used to irradiate the total area of the FPD pattern. The CAD data of the pattern for the FPD glass, with the minimum of $140\mu m$ line space is shown in Fig.6(a). An illustrated example showing the accumulation

of 16000 frames appears in Fig.6(b). A portion of confirmed lithographic region for the FPD pattern is in Fig.6(c), and the two marked sections are enlarged in Fig.6(d) to show the predicted lithographic pattern region in detail. The obtained FPD pattern region shows that the implemented system based on the region-based pattern generation method is capable of producing an actual pattern region for DMD based maskless lithography.

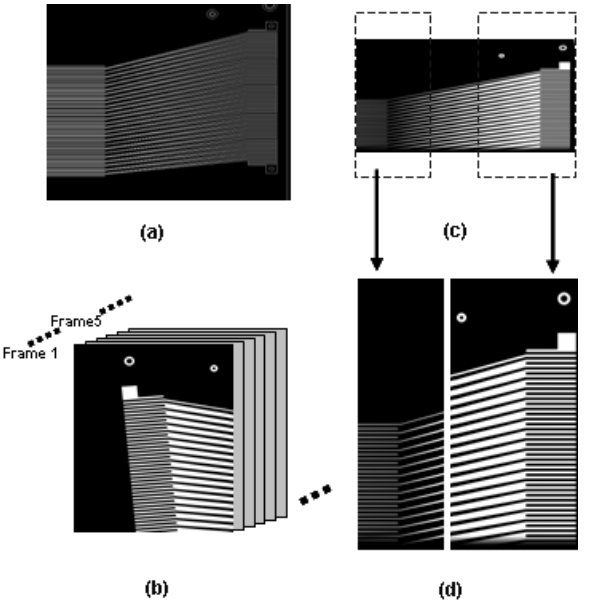


Fig. 6. Actual FPD glass lithography process

6 Results and Discussion

For the validation of the devised RPG system, DMD based maskless lithography is carried out to fabricate an actual FPD glass using evaluation versions of the patterns and actual patterns. An evaluation version of CAD data in DXF format is shown in Fig.7(a), 2.25^o tilted lithographic pattern simulated by the RPG system is shown in Fig.7(b), the electron microscope image from the actual FPD lithography result by the RPG system is shown in Fig.7(c), an enlarged portion of the electron microscope image of the actual FPD pattern by the RPG system is shown in Fig.7(d), and the electron microscope image of the company logo of the LG Electronics Co. by the RPG system is shown in Fig.7(e). All of the electron microscope images shown in Fig.7 are in the condition before etching. The boundaries of the final patterns appear to be clear enough, proving the accuracy of the devised system. Moreover, no unacceptable manifestation of discrepancies between the input from the CAD data and the output from the actual lithography is found. With $30\mu m$ of the FOV, the error in $80\mu m$ line is

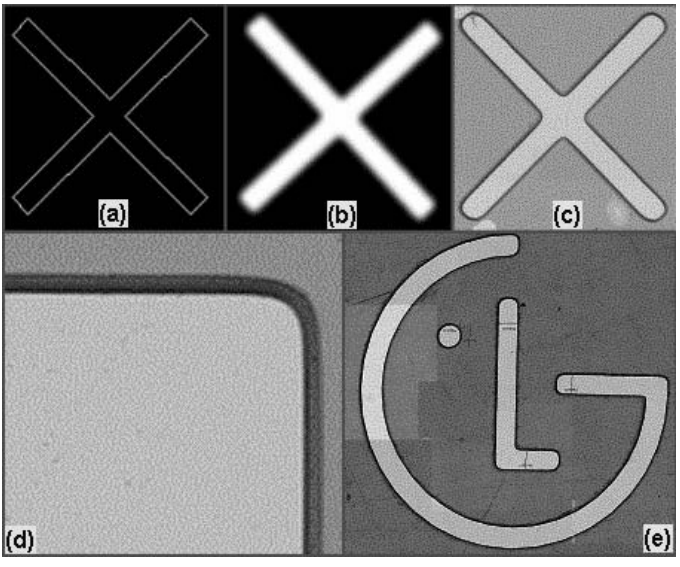


Fig. 7. Actual FPD glass fabrication by RPG system

found to be near zero and the one in $30\mu m$ lines is found to be $0.5\% - 3\%$ insisting that $30\mu m$ FOV is relatively large for $30\mu m$ patterns. The error is also in the tolerable range. Overall, the results of the actual FPD lithography verify that the implemented system is capable of generating lithographic patterns precise enough to acquire content from FPD manufacturers.

7 Conclusions

To make the lithographic pattern generation scheme robust and flexible, we place our primary concern on the pattern and consider pattern generation for maskless lithography using the DMD as a graphic recognition field problem. Then, we conceptualize the lithographic pattern generation process as dual pattern recognition in the substrate's view and the DMD's view. For the conversion of vector patterns recognized in the substrate's view into raster patterns recognized in the DMD's view, the unique criterion, *i.e.*, the area ratio is devised upon dual pattern recognition. The accuracy of the devised pattern generation scheme through dual pattern recognition is verified by the results of the actual FPD lithography. Thus, the devised scheme is robust, flexible, accurate, and even free from restrictions due to pattern structure, pattern size, and light source shape.

Acknowledgments

This work was supported by LG Electronics Co. under Grant to IAMTEN Laboratory, Tongmyong University. We thank the LG Electronics Co. for letting us share the confidential contents.

References

1. Dudley, D., Duncan, W., Slaughter, J.: Emerging Digital Micromirror Device (DMD) Applications, Proceedings of The International Society for Optical Engineering (2003) 4985
2. Mei W.: Point array maskless Lithography, U.S. Patent No.6,473,237 B2 (2002)
3. Hofling, R., Ahl, E.: ALP : universal DMD controller for metrology and testing, Proceedings of The International Society for Optical Engineering (2003) 5289
4. Jung, S.H.: Maskless Lithography Device, Korea Patent pending, Application Pub. No. 2003- 0059705 written in Korean (2002)
5. Mei W., Kanatake T., Ishikawa, A.: Moving exposure system and method for maskless lithography system, U.S. Patent No. 6,379,867 B1 (2002)
6. Chan, K.F., Feng, Z., Yang, R., Ishikawa, A., Mei, W.: High-resolution maskless lithography, Journal of Microlithography, Microfabrication, and Microsystems **2** (2003) 331–339
7. Kanatake, T.: High Resolution point array, U.S. Patent No.6870604 B2 (2005)

Global Discrimination of Graphic Styles

Rudolf Pareti and Nicole Vincent

Laboratoire Crip5-Sip, Université Paris 5, 45, rue des Saints-Pères, 75270 Paris
rudolf@pareti.org
nicole.vincent@math-info.univ-paris5.fr

Abstract. Discrimination between graphical drawings is a difficult problem. It can be considered at different levels according to the applications, details can be observed or more globally what could be called the style. Here we are concerned with a global view of initial letters extracted from early renaissance printed documents. We are going to present a new method to index and classify ornamental letters in ancient books. We show how the Zipf law, originally used in mono-dimensional domains can be adapted to the image domain. We use it as a model to characterize the distribution of patterns occurring in these special drawings that are initial letters. Based on this model some new features are extracted and we show their efficiency for style discrimination.

1 Introduction

Graphical documents can be considered as images organized with lines and repeated small drawings. An architectural plan or a map contain a lot of symbols and we can determinate from them a style. Indeed they can be indexed and discriminated by the author or the map editor. The style has to be defined in a proper way, either from an objective point of view or form an expert view. Here we consider the second way.

We can consider that initial letters in ancient documents and more especially those that are printed [1] as in the Renaissance period are graphics composed with lines and replicated symbols as illuminations.

Most of the books in the middle age period were reproductions of ancient and religious texts, realized in monasteries in which the copyists worked under dictation. The invention of printing led to the multiplication of the books. Nevertheless they tried to attain the quality and the beauty of the ornamented presentation of previous period manuscripts. Books were rich, well ornamented with ornamental initial letters, illuminations, ornate headings and various pictures. Our pictures of the ornaments are in grey levels because of the technical problems. In fact the artists working in the editors shops respected a lot of already existing constraints:

- the page setting
- the requirements of the clients
- and all the representative conventions (saints drawn with their attributes, decorative stylization…)

Some artistic schools were created that had specific aspects depending on the location of the shop or on the people working there. Some copies were made by people in

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 120 – 130, 2006.
© Springer-Verlag Berlin Heidelberg 2006

order to usurp the origin of a book. Some illuminators copy the others and it was not rare that an illuminator realized almost the same illumination as some drawings already realized in other places. Nevertheless the graphics differed in some details from the original one and a close look at the image is needed to determine whether some initial letters come from the same printer or not.

The ornamental initial letter has different functions:

- a religious connotation
- a visual reference to understand quickly the content of the book
- a wealth mark

Since a book contains several media and comprises both a support and content, the study of the text interests many experts in different domains. The text in itself does not contain the same information as can provide the study of the illuminations and particularly the ornamental initial letters. Indexing these pictures will enable us to know whether a book has been made by the same artistic school, or if someone has cribbed from another artist. Even better, the books are often damaged and we cannot know who the author is, when it has been written, if it is an original version of the book or some copy, what edition is being read. To answer these questions and many others, the fonts used in the book can be observed, the figures can be used too. Here we are concerned with the classification of the initial letters. They could tell us a lot of information erased by the time effect. In the Renaissance period, the period we are interested in this study, the initial letters were some little black and white graphical drawings with illumination. A lot of methods exist to classify them, the most simple relies on the use of the histograms, others rely on pattern matching techniques [2-3], others rely on indexing methods linked to image indexing and retrieval techniques. We have tried to create a more robust method than the other existing methods taking into account the many aspects contained in these small elements, lines, dashed lines or zones, symbols, letters. Statistical approach seems necessary to solve the problem. Indeed the structural approaches are still not mature enough to handle small differences we cannot model well. So we have chosen an approach based on elementary windows, which would use a mathematical model to characterize the drawing. Of course, the model depends on parameters. Zipf law seems to hold in many observed 1D phenomena, then we though to apply it to this type of images.

In a first part we are going to recall Zipf law, as developed in the study of 1D signals, then we will show how to transpose this text law for the purpose of image analysis and more specially for the specific graphical drawings we are studying. Finally results will be presented.

Our process can be divided into two steps, the learning phase and the application phase. In the first step, the training set comprises, for each style already known a small number of ornamental initial letters that experts have classified.

Then in the second step, we are going to take any ornamental initial letter at disposal and we are going to classify it according to the classifier developed. Therefore the output of the system is a classification of the ornamental initial letters according to their style. A verification method can be developed too.

2 The Zipf Law

Zipf law is an empirical law expressed fifty years ago [4]. It relies on a power law. The law states that in phenomena figured by a set of topologically organized symbols, the distribution of the occurrence numbers of n-tuples named patterns is not organized in a random way. It can be observed that the apparition frequency of the patterns M_1, M_2… M_n, we note N_1, N_2... N_n, are in relation with rank of these symbols, if sorted with respect to their decreasing occurrence frequency. The following relation can be observed:

$$N_{\sigma(i)} = k \times i^a \tag{1}$$

$N_{\sigma(i)}$ represents the occurrence number of pattern with rank i. k and a are constants. This power law is characterized by the value of the exponent a. k is more linked to the length of the symbol sequence studied. The relation is not linear but a simple logarithmical transform leads to a linear relation between the logarithm of N and the logarithm of the rank.

Then, the value of exponent a can be estimated by the leading coefficient of the regression line approximating the experimental points of the 2D graph ($\log_{10}$ (i), $\log_{10}$ ($N_{\sigma(i)}$)) with *i=1 to n*). Further, the graph is called Zipf graph. One way to achieve the approximation is to use the least square method. As points are not regularly spaced, the empirical points of the graph have to be completed by rescaled points along the horizontal axis.

The validity of this law has been studied and observed first in linguistic study of texts, later it has been observed in other domains [5-6] but always for mono dimensional signals.

In order to study graphical drawings, we are going to try and apply Zipf law to images and therefore to adapt the concepts introduced in the statement of Zipf law to two dimensional data.

3 Application to Images

The ornamental initial letters we are studying have been scanned as grey level images where each pixel is encoded with 8 bits (256 different levels). The intensity is the information encoded. In spite of the black and white property of the drawings, we have preferred grey level information to achieve the study. The noisy paper and the deformation due to ancient paper need to have this precision, the relative grey levels are more significant than the absolute values. The method we are proposing is invariant under the geometrical transform that leave invariant the shape of the mask. The invariance to change of scale that can occur when images are scanned in different conditions is intrinsically linked to the method itself.

In the case of the mono dimensional data, the mask was limited to successive characters. When images are concerned, the mask has to respect the topology in the 2D space the data is imbedded in. We have chosen to use a 3x3 mask because they define the most often considered neighborhood of a pixel in a 2D space.

Then the principle remains the same, the mask scans the whole picture and the numbers of occurrences of each pattern are computed.

Several problems have to be considered. How to describe the patterns? That is to say how the drawing has to be encoded and what are the properties we want to make more evident?

3.1 Coding of the Patterns

We have considered a 3x3 size mask and 256 symbols are used to code pixels. Therefore, there are theoretically 256^9 different patterns. This number is much larger than the number of pixels in an image. Indeed, if patterns are not frequent enough, the model that is deduced from Zipf model cannot be reliable, the statistics loose there significance.

For example a 640x480 image can contain only 304964 different patterns. Then, it is necessary to restrict the number of possible patterns to give sense to Zipf model. The coding is decisive in the matter. According to the coding process chosen, the Zipf curve general shape can vary a lot. When it differs from a straight line, the model of a power law is not suited for the phenomenon. If several straight segment appear we can conclude several phenomena are mixed and several models are involved. Besides, different codings allow to make more evident different properties of an image.

Some studies have shown Zipf law was holding in the case of images with different encoding processes [7]. We are looking for coding process that gives models apt at discriminating the graphical drawings we are studying. in our case, this qualifies as effective a coding process. Two drawings that look alike from a style point of view should verify similar power laws. We are going to see different coding methods we have tested.

According to the remarks previously done, the number of different possible must be decreased. This can be done decreasing the number of pixels involved in the mask or decreasing the number of values associated with a pixel.

3.2 The General Ranks

Here our motivation is to respect the vision of a scene that relies more on differences of grey levels than on the absolute values. So, within the mask, the grey level values are replaced by their rank when they are sorted according to the grey level values. The method affects the same rank when the grey levels have the same value.

Then the maximum number of values involved in the mask is 9 and this leads to a very large decrease in the number of different possible patterns.

The number of symbols used to represent the grey levels is limited to 9. It can be noticed that the patterns in figure 1 (a) and (b) are different, yet, the coded pattern (c) that is generated is the same in both cases. It is one of the limitations of this coding. Of course information is lost. Further the method relying on this coding process will be called general rank method.

Image Pattern 1:

2	8	6
21	31	31
32	32	32

Image Pattern 2:

130	136	134
149	159	159
160	160	160

Coded pattern:

0	2	1
3	4	4
5	5	5

Fig. 1. patterns coded using general rank method

3.3 Grey Level Quantization

The previous method has lead to 9 possible values associated with a pixel in the pattern considered. A simpler way would be to consider only k grey levels to characterise the intensity level of the pixels. Most often it is sufficient to observe an image. More over the images we are dealing with in the graphical drawings are essentialy black and white. A quantisation in k equal classes would lead to unstable results, so we have chosen to use a classsification method of the grey levels in k classes by way of a k-mean algorithm. Further the method relying on this coding process will be called **k-mean** [8]. We have experimented different values of k. In figure 2 we present an example with 9 clusters.

Grey level	**0-20**	**21-76**	**77-96**	**97-120**	**121-146**	**147-174**	**175-203**	**204-229**	**230-255**
cen-ter	15	56	86	107	133	159	190	217	242
N° Class	0	1	2	3	4	5	6	7	8

Fig. 2. K-mean example

The example depends on the drawing and considers 9 classes. We can see that some classes may be closer than others.

3.4 Cross Patterns

An other way to decrease the number of possible patterns is to limit the number of pixels in the pattern. To remain coherent with the 2D topology we have chosen to consider the smallest neighborhood of the pixel defining 4-connectivity. It is precised in figure 3.

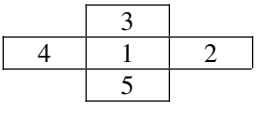


Fig. 3. Mask shape indicated as cross mask

In this case we have also achieved a drastic decrease in the number of grey levels considered as we have considered only 3 of them. This number is in fact issued from the nature of the images we are working on. They are rather black and white images. A k-mean with k equal to 3 has been processed on the set of pixels of each image. The number of possible patterns is therefore equal to 3^5=243, that is about the same as the initial number of grey levels but the information contained in the value is not the same. The k-mean classification makes the method independent on the illumination of the scanned image. Further the method relying on this coding process will be called crossmean method.

3.5 Zipf Curve Construction

Whatever the encoding process used, Zipf graphs can be built. Now in order to study a family of images, these plots have to be compared. A close look at the curves shows they are not always globaly linear, that is to say Zipf law does not hold for the whole patterns. It depends on the coding process. Nevertheless some straight line segments can be observed. According to the coding process used these zones can be interpreted. With the second process for example, we observed the left part of the graph is concerned with the regions in the image whereas the right part gives information on the contours present in the image. Then, we can extract on the one hand some structure indication of the regions and on the other hand the structure of the contours within the images.

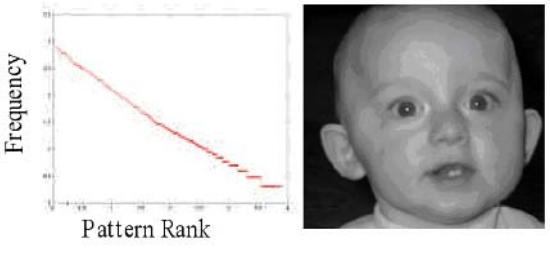


Fig. 4. Zipf curve associated to an image with respect to general rank method

Then we have chosen to consider in each curve up to three different linear zones. They are automatically extracted into three zones as shown in figure 5 using a recursive process.

The splitting point in a curve segment is defined as the furthest point from the straight line linking the two extreme points of the curve. The fact is that the image carries a mixt of several phenomena that are highlighted in the process and we can model them. Several power laws are involved and then several exponent values can be computed.

The Zipf curve has been drawn with respect to a logarithmic scale, therefore, it is necessary to begin with a resampling of the curve.

Then the output of the process is made of 3 meaningful values associated with the picture (the ornamental initial letter figure 5). They correspond to the 3 slopes or leading coefficients.

From this material we are going to learn the characteristics of the different styles.

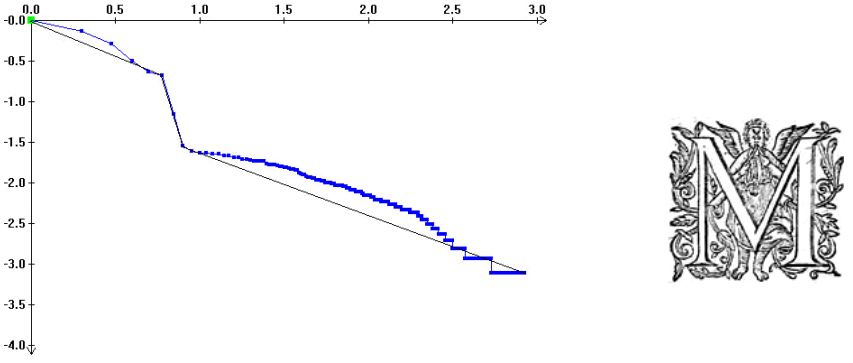


Fig. 5. Example of an initial letter and its Zipf plot where are indicated the different straight zones extracted

4 Training

We use a data base made of more than 300 images of initial letters. They have been extracted from different books. They can be considered as belonging to one of three styles, each represented by a number of the ornamental initial letters {S1, S2, S3}. In order to be able to classify the ornamental initial letters we have separated this set in a learning set and a test set according to the ratios indicated in table 1.

Table 1. composition of the learning and test sets

Style	Training set	Test set
S1	28	163
S2	10	22
S3	24	68
other	0	43

In order to verify the possible efficiency of every exponent associated with one of the models based on Zipf law we have presented, we are going to calculate all the ornamental initial letters Zipf curves of every style and deduce the average and the standard deviation associated with every style. They are indicated in table 2. To have more efficient results we decide to process a normalisation step before the parameter computations. We apply an histogram normalization filter in order to take better advantage of the image spectrum.

We can notice that the general rank method is not satisfactory at all. The K-Means method remains a good method for all slopes, but execution time is relatively long. Finally the crossmean method is the most discriminative as far as the first two slopes are concerned. That is why in the next tests we will use in priority the parameters extracted in the crossmean and k-mean Zipf curves.

Table 2. Analysis of the learning test according to extracted features

Methods	style	Average & standard deviation of slope 1		Average & standard deviation of slope 2		Average & standard deviation of slope 3	
General Ranks	S1	-0.15	0.2	-1.19	0.8	-0.66	0.4
	S2	-0.22	0.1	-0.79	0.2	-0.46	0.3
	S3	-0.22	0.1	-1.29	0.2	-0.83	0.1
k-mean k=3	S1	-0.31	0.3	-1.37	0.9	-0.77	0.5
	S2	-1.7	0.7	-0.43	0.2	-0.97	0.3
	S3	-1.38	0.8	-0.65	0.7	-0.82	0.1
Cross mean	S1	-0.99	0.6	-1.06	0.5	-4.49	1.9
	S2	-2.67	0.4	-0.75	0.1	-4.53	0.9
	S3	-1.71	0.2	-0.51	0.1	-4.27	0.3

5 Results

The first approach we have implemented relies on the modeling of a style by the average values computed in the learning phase. Then distance is calculated using the usual euclidian distance. $\{\mu_1, \mu_2, \mu_3 \ldots\}$ are the averages learned for every style and P_j are the different parameters considered in the method.

$$\text{distance}(X, S_i) = \sqrt{\sum_j (\mu_i^j - Pj)^2} \tag{2}$$

The results we obtained in such a way are indicated in table 3.

Table 3. Results using Euclidean distance and single prototype style

Recognition	S1	S2	S3	other
Style1	**25.15%**	1.23%	26.99%	46.63%
Style2	0.00%	**36.36%**	22.73%	40.91%
Style3	0.00%	0.00%	**94.52%**	5.48%

We can notice that the results are not satisfactory. Indeed, the method does not take into account the classes geometry in the different styles. In regard to these results we have decided to use the Mahanalobis distance. Besides, some experiments have been done, in order to choose the best value of the grey level number in the k-mean method. They are illustrated in figure 6.

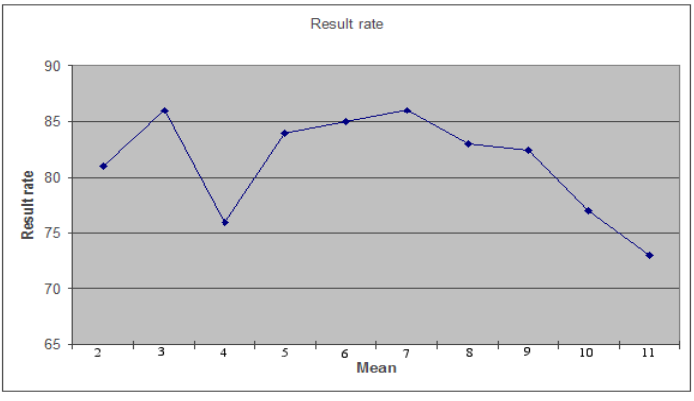


Fig. 6. Evolution of the recognition rate according to the number of the grey level used in the final encoding process

From the previous study we have chosen to use three grey levels, this enables to speed up the process. We notice that the most efficient results have been reached with the k-mean method. Results are indicated in table 4.

Table 4. Results using only k-mean parameters and Mahanalobis distance

Style	Images in the test base	Images detected in test base	Recognition rate	False accepted rate
Other	14.77%	39.38%	**64.58%**	75.78%
Style 1	50.46%	41.85%	**75.00%**	9.56%
Style 2	11.69%	0.92%	**7.89%**	0.00%
Style 3	23.08%	17.85%	**69.33%**	10.34%

Table 5. Results using knn method

Style	Images in the test base	Images detected in test base	Recognition rate	False accepted rate
Other	14.77%	20.00%	**52.08%**	61.54%
Style 1	50.46%	51.08%	**86.59%**	14.46%
Style 2	11.69%	12.00%	**97.37%**	5.13%
Style 3	23.08%	16.92%	**66.67%**	9.09%

In table 4 we have indicated the recognition rate as well as the false recognition rate for each class. But the Mahanalobis distance presumes that the slopes of the images to be classified in the class, are distributed according to a normal distribution

but we noted that it was not always the case. So we have decided to implement the k nearest neighbours (knn) method where no hypothesis has to be done. Results are indicated in table 5.

The results seem to be more satisfactory. We have experienced the combine use of several encodings but the results have not been significantly better than those presented here.

6 Indexation

Besides our algorithm can be used in an indexation way. Choosing an initial letter in our database represented with parameters extracted from Zipf law formula and an Euclidean distance comparison we print the n nearest initial letters. Our results are good as we can see in Fig 7, and encouraged us to follow this way for our future developments.

Fig. 7. Example of indexation results

We took in account the slopes but also the break points between each slope and it gave better results as we can see in table 6. A drawing is featuring by six numbers, the three slopes studied before and the three break points of the curve.

Table 6. Results for indexation

K	Style 1	Style 2	Style 3
1	91%	100%	92%
3	100%	100%	100%
5	100%	100%	100%
10	100%	100%	100%

7 Conclusion

Here we show the use of a model developed in the field of 1D phenomena can give good results in case of drawings. This law allows to define global parameters based on details. According to the type of encoding used, the nature of information differs. Other encoding processes can be experimented and the method can be applied to other problems involving graphical drawings as ornate headings and manuscripts indexation or recognition [9-10]. An other pattern size and an other number of classes could be experiment according to the problem. The method is invariant under any rotation and change of scale and enable us to apply it on drawings scaned in different resolutions and conditions.

Acknowledgements

Here we want to thank all the persons in the CESR (Centre d'Etudes Supérieures de la Renaissance) located in Tours University, France. They have made this study possible. Indeed, the initial letters scanned images have been provided as well as the expertise associated with them have been provided by members of the CESR.

References

1. Journet N., Eglin N., Ramel J.Y., Mullot R.: Text/Graphic labelling of Ancient Printed Documents. Proc. Eighth International Conference on Document Analysis and Recognition (2005) : 1010-1014.
2. Eakins J.P.: Content base image retrieval – can we make it deliver? Proc. 2nd UK Conference on image retrieval, Newcastle upon tyne: (1999)
3. Melessanaki K., Papadakis V., Balas C., Anglos D.: Laser induced breakdown spectroscopy and hyper-spectral imaging analysis of pigments on an illuminated manuscript. Spectrochimica Acta Part B 56 (2001) : 2337-2346
4. Zipf G.K.: Human Behavior and the Principle of Least Effort. Addison-Wesley : (1949)
5. Cameron A., Cubelli R., Della Sala S.: Letter assembling and handwriting share a common allographic code. Journal of Neurolinguistics 15(2): 91-97, 2002.
6. Dellandrea E., Makris P., Vincent N.: Zipf analysis of audio signals. Fractals 12(1) : 73-85, 2004.
7. Caron Y., Charpentier H., Makris P., Vincent N.: Power Law Dependencies to Detect Regions Of Interest. Proc. 11th International Conference DGCI 2003, Naples, Italy, (2003).
8. Hartigan J. A., Wang M. A.: K-mean clustering Algo. JSTOR revue : 100-108.
9. Avilés-Cruz C., Rangel-Kuoppa R., Reyes-Ayala M., Andrade-Gonzalez A., Escarela-Perez R.: High-order statistical texture analysis font recognition applied. Pattern Recognition Letters 26(2) : 135-145, 2005.
10. Moalla I., Lebourgeois F., Emptoz H., Alimi A.M.: Contribution to the Discrimination of the Medieval Manuscript Texts: Application in the Palaeography. Document Analysis Systems (2006) : 25-37

Recognition for Ocular Fundus Based on Shape of Blood Vessel

Zhiwen Xu[1,2], Xiaoxin Guo[1,2], Xiaoying Hu[3], Xu Chen[1,2], and Zhengxuan Wang[1,2]

[1] Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, Changchun, 130012, China
[2] College of Computer Science and Technology, Jilin University, Changchun, 130012, China
[3] The First Clinical Hospital, Jilin University, Changchun, 130012, China
xuzw@email.jlu.edu.cn

Abstract. A new biometric technology-recognition ocular fundus based on shape of blood vessel skeleton-is addressed in this paper. The gray scale image of ocular fundus is utilized to extract the skeletons of its blood vessels. The cross points on the skeletons are used to match two fundus images. Experiments show high recognition rate, low recognition rejection rate as well as good universality, exclusiveness and stability of this method.

1 Introduction

Biometrics is a new research area about automatic recognition of a person based on his/her physiological or behavioral characteristics. Because biometric features are neither like codes which are easy to be forgotten or breached nor like holdings which are likely to be stolen or removed, biometric technology is a more reliable, convenient, effective and popular ID authentication solution. In principle, biometric features can be applied to ID authentication just due to their inherent qualities, that is, universality, which means everyone possesses his biometric features; exclusiveness, which means each individual has his unique features; stability, the feature to be utilized in authentication is to remain steady at least for a period of time; and extractability, which means biometric features can be measured quantitatively. However, a biometric feature, though with the four qualities above, may be infeasible in a practical system, because a practical system has to take the following three aspects into consideration: capability, which refers to veracity, speed and robustness of authentication and required resources; acceptability referring to the degree to which a given biometric technology would be accepted; and deceivability referring to how hard a transaction fraud could cheat on the system. In this paper, we present a new biometric technology, which is recognition ocular fundus based on shape of blood vessel skeleton. The cross points on the skeletons are used to match two fundus images. Experiments show high recognition rate, low recognition rejection rate as well as good universality, exclusiveness and stability of this method.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 131–139, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Related Work

The shape of blood vessel for ocular fundus is an important indicator to diagnose such diseases as hypertension, vascular sclerosis, coronary artery sclerosis and diabetes. Many general image processing methods can be applied to images of blood vessels of ocular fundus. Intuitively, the simplest method is to binarize an image using a global threshold determined by the histogram. However, the image segmentation results are usually not so good due to influence of uneven illumination, diverged focus and noise. The part adaptive binary method also turns out bad, because of the physiological complexity of ocular fundus. The gradient edge detection (Sobel algorithm) is advisable to extract image with ideal edge, whereas at the same time it amplifies noise and discontinuity of digitalized image. Besides, elliptical vein section together with such external factors as photo, digitalization etc attributes to blurred edge. Thus gradient edge detection isn't practical. In a bid to reduce noise and increase continuity of edge extraction, Marr et al propose zero-crossing edge detection method (Log operator). Though it is quite precise in extracting edge, it detects overmany tiny variations and many unexpected closed circles that are hard to be separated from circular bleeding spot and tiny angioma. Considering that zero-crossing can neither always correspond to real edge nor always signal edge position precisely, Ulupinar [1] makes relevant revision. Canny [2] raises optimal operator suitable for any randomly-shaped edge extraction. However, due to utilization of Gaussian filter, it still remains imprecise occasionally. Though morphological gradient [3] operator is easy and quick, it's confined to image with pepper and salt noise. Relax method is used to extract linear blood vessels. Since arteries and veins cross over each other, blood vessels have to be segmented into several pieces. Hueckel Edge operator applies fitting method of blood vessel to extract edge [4]. It is insensitive to noise and effective in intense texture region, but it demands tremendous calculation. Taking into consideration the inherent features of blood vessel [5], being that blood vessel is of linear shape and gradient direction on the left threshold is just opposite to that on the right threshold, Tascini puts forward the method of search edge direction [5]. The edge of blood vessel in the second period of hypertension is rather blur and of very low contrast so that it can't be traced. In order to improve that, Rangayyan R. M. et al suggests reinforcing linear feature of blood vessel. But the result isn't good due to exudation for blood vessel of ocular fundus. Chauduri et al. employs Gaussian models of twelve different directions to filter fundus blood vessel image [6]. Then, those methods aren't good at enhancing image for blood vessel of hypertension patients. In addition, utilization of Gaussian models of fixed size makes it fail in dealing with blood vessels with salient variation in diameter and highly curved shape. Substitution of Gaussian models of different sizes is sure to cause excessive computation. The literatures [8, 9] make a study on matching of blood vessel image. The literatures [10, 11] make a study of point pattern algorithm of fingerprint. Based on the researches mentioned above, together with biometric features for blood vessel of ocular fundus, this paper addressed the recognition for blood vessel of ocular fundus, by employing gradient vector curve to analyze

the features for blood vessel of ocular fundus. The shape of skeleton feature for blood vessel of ocular fundus using contrast-limited adaptive histogram equalization is extracted at first in this paper. After filtering treatment and extracting the shape of blood vessel.

3 Algorithm to Extract Blood Vessel of Ocular Fundus

The diameter of blood vessels of ocular fundus may vary due to ageing and diseases. However, their distribution and direction usually remain unchanged. This encourages the use of shape for vector curve that analyzes distribution of blood vessel orientation to be utilized as biometric features for blood vessel of ocular fundus. The concrete steps concerned with extracting algorithm as the following:

(1) Enhance the contrast of the gray scale image and utilize contrast limited adaptive histogram equalization. At first, deal with subregions in the image and

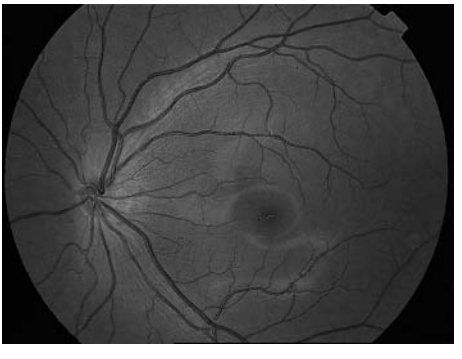


Fig. 1. Fundus image of gray scale

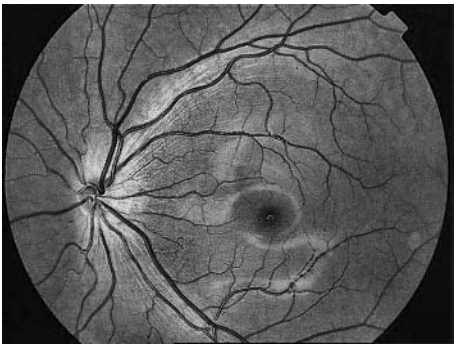


Fig. 2. Gray scale enhanced image

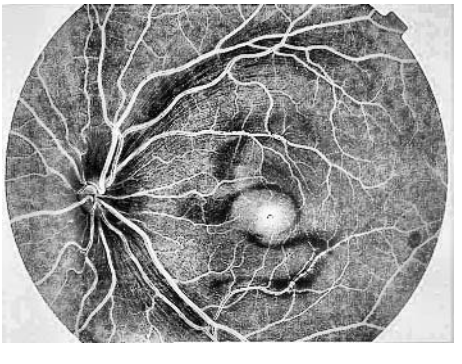


Fig. 3. Inversion to enhance gray scale

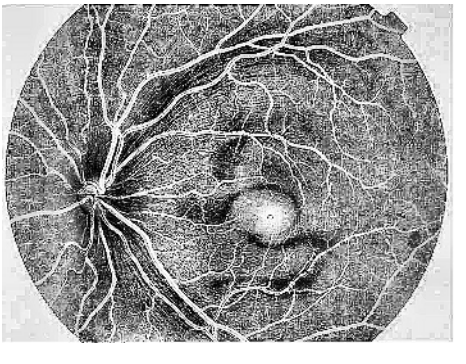


Fig. 4. Filtering using of un-sharp algorithm

then converge adjacent small regions with method of bilinear interpolation in order to get rid of artificial edges. Fig.1 is a gray scale ocular fundus image; Fig.2 is scale enhanced image; Fig.3 is inversion of Fig.2.

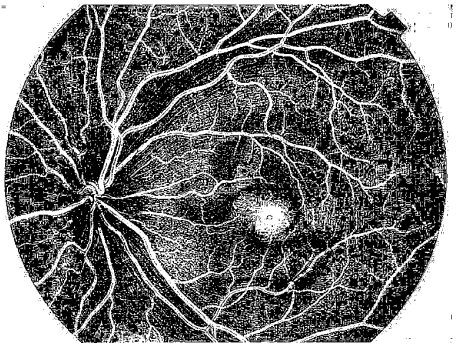


Fig. 5. Threshold transform of gray scale

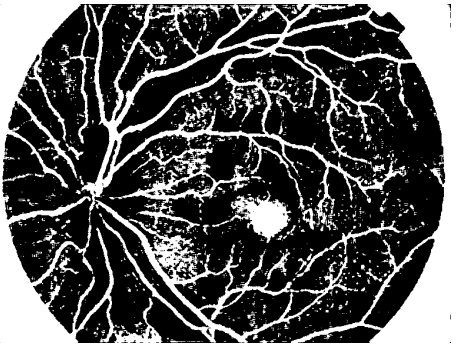


Fig. 6. Median filter

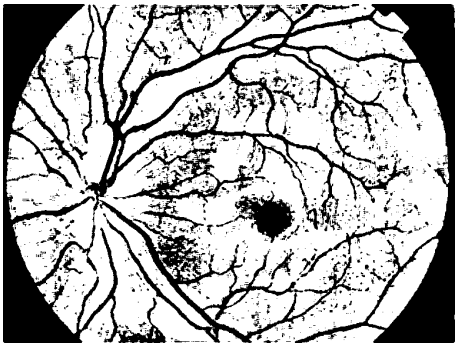


Fig. 7. Exchange of 0 and 1

Fig. 8. Filling holes

(2) Filter the image by means of un-sharp algorithm for software of Matlib 7.0 (Fig.4).

(3) Binarize. Determine all the local maximums over a supposed threshold and input gray scale image as a parameter. In a dimorph image, local maximum of output dimorph image is assumed to be 1, and the remainder 0, which are used to search out region whose brightness changes most. (Fig.5). At last, median filter is utilized (Fig.6).

(4) Fill holes. Holes are those dark regions surrounded by whiter setting. Exchange 0 with 1 in the binary image (Fig.7) and erase holes (Fig.8).

(5) Extract shape of skeleton on the basis of erosion(Fig.9). Cut edge disturbance of image (Fig.10).

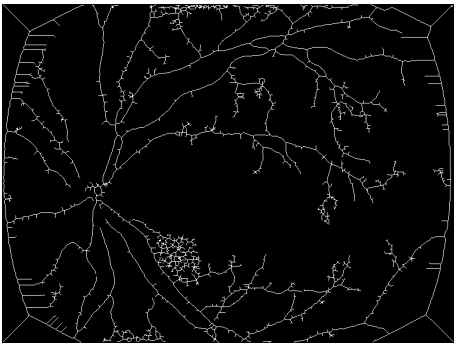


Fig. 9. Skeleton extracting

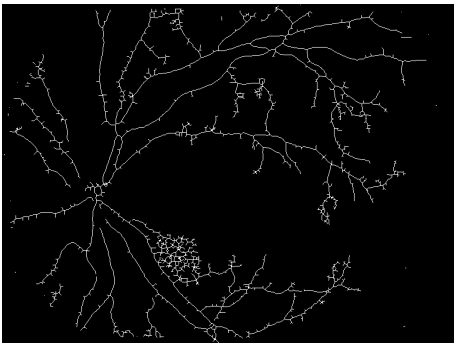


Fig. 10. vector curve of blood vessel skeleton

4 Matching Algorithm for Blood Vessel Images of Ocular Fundus

A image is represented by a set of feature points extracted from its skeleton obtained in Section 3 and matching two images is done by matching their feature points. We denote P as the set of feature points extracted from a test blood vessel image, and Q the set from a standard blood vessel image stored in the template database. These two sets can be formulated as follows.

$$P = \{p_1, p_2, \cdots, p_m\}$$

$$P = \{(x_{p1}, y_{p1}, \theta_{p1}, z_{p1}), (x_{p2}, y_{p2}, \theta_{p2}, z_{p2}), \cdots, (x_{pm}, y_{pm}, \theta_{pm}, z_{pm})\} \quad (1)$$

$$Q = \{q_1, q_2, \cdots, q_m\}$$

$$Q = \{(x_{q1}, y_{q1}, \theta_{q1}, z_{q1}), (x_{q2}, y_{q2}, \theta_{q2}, z_{q2}), \cdots, (x_{qm}, y_{qm}, \theta_{qm}, z_{qm})\} \quad (2)$$

where $x_{pi}, y_{pi}, \theta_{pi}, z_{pi}$ records four pieces of information of the i^{th} feature point in P: x direction, y direction,rotated factor and zoom factor; $(x_{qj}, y_{qj}, \theta_{qj}, z_{qj})$ records the same four pieces of information of the j^{th} feature point in Q. Matching two images of blood vessels should also consider situations such as rotation, translation, scaling etc. Suppose the two images of blood vessel could match completely. The input image of blood vessel can be transformed in a certain way (rotation, translation, zoom and rotation) to get template image. So set P can be transformed roughly into set Q by means of rotation, translation or zoom. Try to search out the transforming method that is able to match as many points between the two sets as possible. If two points stay close to each other and points to similar direction after being transformed, they are considered matching. However, with certain transformation, some points of one set could not search their correspondence points in another set.

In order to transform a given feature point of input image of blood vessel to a corresponding position in the template image of blood vessel, the corresponding

transformation factor should be known. The algorithm in this paper deals with image of blood vessel for the same differentiation. In ideal condition, the zoom factor is Z. Assume a given point in the input point set P is $p_i(x_{pi}, y_{pi}, \theta_{pi}, 1)$, which is transformed by the following formula into $p_i(x_{pi}^T,\ y_{pi}^T,\ \theta_{pi}^T,\ Z_{pi}^T)$; and assume a given point in the template point set is $q_j(x_{qj}, y_{qj}, \theta_{qj}, z_{qj})$. If $(x_{pi}^T, y_{pi}^T, \theta_{pi}^T = (x_{qj}, y_{qj}, \theta_{qj}, z_{qj})$, the transforming factor is that p_i is similar to q_j on the condition of ($\triangle$x, $\triangle$y, $\triangle\theta$, $\triangle$ z). The images of blood vessel are captured using the same facilities and the same distance. So p_i is simplified as $(x_{pi}, y_{pi}, \theta_{pi})$ and q_j as $(x_{qj}, y_{qj}, \theta_{qj})$. The zoom factor in the course of transformation is ruled out.

$$\begin{vmatrix} x_{p_i}^T \\ Y_{p_i}^T \\ \theta_{p_i}^T \\ 1 \end{vmatrix} = \begin{vmatrix} \cos\triangle\theta & -\sin\triangle\theta & 0 & \triangle x \\ \sin\triangle\theta & cos\triangle\theta & 0 & \triangle y \\ 0 & 0 & 1 & \triangle\theta \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x_{p_i} \\ Y_{p_i} \\ \theta_{p_i} \\ 1 \end{vmatrix} \tag{3}$$

Where $\triangle$x and $\triangle$y are the translation factors in x direction and y direction respectively, $\triangle\theta$ is rotation factor, and $\triangle$Z=1 is zoom factor. a blood vessel skeleton can be judged as whether the same or not with a template skeleton.

4.1 Determination of Reference Point and Calculation of Transformation Factor

In the process of blood vessel matching, the calculation of matching reference point is a great importance. Clustering method is utilized to get a precise matching reference point and a group of transformation parameters. Though this method can lead to a quite precise matching reference point, it involves excessive computation.

Any single feature of point p_i in the input point set and any single feature point q_j in the template point set form a point pair. Another two feature points p_1, p_2 chosen from the input point set and q_1, q_2 from the template set form a feature point subset. Thus two triangles $p_i p_1 p_2$ and $q_j q_1 q_2$ come into being in the input image and template image respectively. p_i, p_1, p_2 and q_j, q_1, q_2 are three vertices for the two triangles respectively. In order to determine whether the point pair (p_i, q_j) is a pair of possibly matching reference points, the similarity of two triangles is to be clarified in this paper. If two triangles are of a high similarity, the feature subsets constituted by vertices of this two triangles are thought to form a type of matching and they are regarded as possible matching point pairs. Consequently, the transforming parameter according to two feature subsets becomes transforming parameter for two images. Till now two images of blood vessel can be transformed in accordance with the obtained transforming parameter, which leads to a further examination of similarity between two triangles.

According to triangle stability principle, three sides bound a triangle. So the distance between the three feature points and their mutual spatial relation can be used to examine the similarity of triangles constituted by the two sets of feature points. Calculating triangle similarity and transforming parameter includes.

(1) Calculate respectively corresponding side-length to vertex p_i and q_j, $|p_1p_2|$ and $|q_1q_2|$.

(2) If $||p_1p_2|\text{-}|q_1q_2|| > D_1$, one corresponding side of the two triangles is not of the same length, then the two triangles are not congruent. The examination ends. Re-choose vertices p_i and q_j and two nearest feature points (p_1, p_2) and (q_1, q_2). Return to the first step.

(3) Otherwise, calculate respectively the distance form p_i to p_1, q_j to p_1 and from p_2 to q_1, p_2 to q_2:$|p_ip_1|, |p_ip_2|$ and $|q_jq_1|, |q_jq_2|$. If $||p_ip_1| - |q_jq_1|| \leq D_2$ and $||p_ip_2| - |q_jq_2|| \leq D_2$, or$||p_ip_2| - |q_jq_1|| \leq D_2$ and $||p_ip_1| - |q_jq_2|| \leq D_2$, it's proved the three sides of the two are of similar length and the two triangles are almost congruent. Otherwise Re-choose vertices p_i and q_j and two nearest feature points (p_1, p_2) and (q_1, q_2). Return to the first step.

(4) According to corresponding vertex of the two triangles, calculate orientation disparity between possibly matching feature points, $\triangle\theta_{p_1q_j}$, $\triangle\theta_{p_1q_1}$, $\triangle\theta_{p_2q_2}$. The formula is.

$$\triangle\theta_{pq} = \begin{cases} \theta_p - \theta_q, if(\theta_p - \theta_q \geq 0) \\ theta_p - \theta_q + 180, if(\theta_p - \theta_q < 0) \end{cases} \tag{4}$$

If angle disparity between corresponding vertices is similar i.e. $\triangle\theta_{p_1q_j} \approx \triangle\theta_{p_1q_1} \approx \triangle\theta_{p_2q_2}$, the angle between the two feature points (p_i, p_1, p_2) and (q_j, q_1, q_2) is supposed to satisfy a rotation relationship. The formula of this rotation is.

$$\triangle\theta = \frac{1}{3}(\theta_{p_iq_j} + \theta_{p_1q_1} + \theta_{p_2q_2}) \tag{5}$$

Otherwise no matching is formed between the sets. Re-choose vertices p_i and q_j and two nearest feature points (p_1, p_2) and (q_1, q_2). Return to the first step.

(5) Choose (p_i, q_j) as a transforming circle for the rotation and then rotate (q_j, q_1, q_2). The consequent point is (q_j, q_1', q_2') . Calculate spatial disparity in x direction and y direction respectively which are $(\triangle x_{p_iq_j}, \triangle y_{p_iq_j})$, $(\triangle x_{p_1q_1}, \triangle y_{p_1q_1})$, $(\triangle x_{p_2q_2}, \triangle y_{p_2q_2})$. The formula is.

$$\Delta x_{pq} = x_p - x_q \tag{6}$$

$$\Delta y_{pq} = y_p - y_q \tag{7}$$

Now if $\triangle x_{p_iq_j}) \approx \triangle x_{p_1q_1'}) \approx \triangle x_{p_2q_2'})$ and $\triangle y_{p_iq_j}) \approx \triangle y_{p_1q_1'}) \approx \triangle y_{p_2q_2'})$, the two feature point subsets (p_i, p_1, p_2) and (q_j, q_1, q_2) meet a kind of transformation relationship in x, y direction. The two subset match whose rotation, translation and transformation factor are $(\triangle\theta, \triangle x, \triangle y)$ respectively, and

$$\triangle x = \frac{1}{3}(x_{p_iq_j} + x_{p_1q_1} + x_{p_2q_2}) \tag{8}$$

$$\triangle y = \frac{1}{3}(y_{p_iq_j} + y_{p_1q_1} + y_{p_2q_2}) \tag{9}$$

According to the obtained reference point $(p_i, q_j)$and transformation factor $(\triangle\theta, \triangle x, \triangle y)$, the shape of blood vessel skeleton is judged whether the same or not.

4.2 Matching Shape of Blood Vessel Skeleton

After obtaining the matching reference point and transformation factor between the two images of blood vessel, rotate and translate the input image of blood vessel in order to determine whether the two images of blood vessel skeleton have the same geometry. Then according to formula (3), calculate feature point coordinates of the transformed input skeleton of blood vessel and vector of curve orientation of the corresponding region. And then place the set of feature point for the transformed input skeleton of blood vessel on the feature point set of the template image, and examine the number of overlapping feature points between the two feature point sets. Since matching isn't very precise, a pair of matching points can't be completely superposed, and certain disparity exists as to position or orientation. This demands certain disparity tolerance. In response, a called threshold box method is adopted in this paper. At first define a rectangular region around every feature point in the feature point set of template skeleton of blood vessel as its corresponding threshold box. As long as a feature point of the transformed input skeleton of blood vessel, after being superposed, falls into the very region and points to similar direction, the two points are regarded as a matching pair. Calculate the total number of matching points and present the matching result.

5 Experimentation

In order to validate that the skeleton shape of blood vessel can serve as a biometric feature for blood vessel of ocular fundus, a TRC-50/50VT fundus camera produced by Japanese Topcon company is used to collect the fundus images of 1000 people as the test data. Each person has ten images taken at different time. To obtain False Non Match Race, or False Rejection Rate, a matching algorithm is made between every image of blood vessel of ocular fundus T_{ij} and its other sample images for blood vessel of ocular fundus F_{ik} $(0 \leq j \leq k \leq 9)$ and then the total matching times should be ((10x9)/2)x 1000=180000. To obtain False Match Rate, or False Acceptance Rate, a matching algorithm is made between the first sample template T_{i0} of every image for blood vessel of ocular fudus in the database and the first image F_{i0} of other image for blood vessels of ocular fundus in the same database. Calculate the ultimate matching result between images for blood vessel of ocular fudus. The total matching times should be (1000x199)/2=99500.

Extracting features from two images for blood vessel of ocular fudus follows several steps. At first, according to gray scale image, fix the brightest region of the window, The region's center of the optical disk acts as the origin. Then starting with the horizontal direction; search out the first point of branch of the three vector curves of blood vessel skeleton. This point of branch is regarded as feature point. And then calculate the matching reference point of the two images for blood vessel of ocular fudus. At last, display practical matching result. The results of the cross-comparison experiment carried out on the 1000 images of blood vessel of ocular fudus are: zero false recognition, 13 false rejection and

0.013 recognition rejection. In this paper, similarity between different images for blood vessel of ocular fundus is measured, where four units are adopted as threshold box: 0.643 of shape skeletons of blood vessel of ocular fudus for different people overlap less than 0.28; 0.896 overlap less than 0.51; 0.973 overlap less than 0.73 and 0.9995 overlap less than 90.

6 Conclusion

Image processing for blood vessel of ocular fundus has been applied to medical research for more than ten years. Everyone has unique and steady features for blood vessel of ocular fundus. The diameter of blood vessel of ocular fundus may change and capillary vessels may increase. However, these changes have no effect to the skeleton shapes of the blood vessels. Anyway, due to certain difficulty, such as, feature extraction, it has not been well utilized. At present, with continuous progress in extracting technology, the biometrics for blood vessel of ocular fundus tends to become an effective biometric technology, as shown in this paper.

References

1. Ulupinar F, Medioni G: Refining edges detected by LoG operator. Computer Vis Graph and Image Process, 51:275 298, 1990.
2. Canny J: Acomputational approach to edge detection. IEEE Trans, PAMI,1986, 8:679 698.
3. Peng J, Rusch Ph: Morphological filters and edge detection application to medical imaging. Annual International Conference of the IEEE Engineering In Medicine and Biology Society, 1991, 13 (1): 0251 0252.
4. Huang C C, Li C C, Fan N, et al: A fast morphological filter for enhancement of angiographic images. Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1991, 13(1): 0229 0230.
5. Tascini G, Passerini G, Puliti P, et al: Retina vascular network recognition. Proc SPIE, 1993, 1898: 322 329.
6. Chauduri S, Chatterjee S, Katz N, et al: Detection of blood vessels in retinal images using two-dimensional matched filters. IEEE Trans Med Imaging,8: 263 269, 1989.
7. Ji T-L, Sundareshan M K, Roehrig H: Adaptive image contrast enhancement based on human visual properties. IEEE Trans Med Imaging,13:573 586, 1994.
8. Matsopoulos G K, Mouravliansky N A, Delibasis K K, et al: Automatic retinal image registration scheme using global optimization techniques. IEEE Trans Information Technology in Biomedicine, 3(1): 47 60, 1999.
9. Maes F, Collignon A, Vandermeulen D, et al: Multi-modality image registration by maximization of mutual information. IEEE Trans Med Img, 16 (2): 187 198,1997.
10. Zhan X-S, Ning X-B, Yin Y-L, Chen Y: An improved point pattern algorithm for fingerprint matching. Journal of Nanjing University, 2003, 39(4): 491 498.
11. Qi Y, Tian J, Deng X: Genetic algorithm based fingerprint matching algorithm and its application on automated fingerprint identification system. Journal of Software,11(4),pp488 493, 2000.

Adaptive Noise Reduction for Engineering Drawings Based on Primitives and Noise Assessment

Jing Zhang, Wan Zhang, and Liu Wenyin

Dept of Computer Science, City University of Hong Kong, Hong Kong SAR, PR China
{jzhang, wanzhang, csliuwy}@cityu.edu.hk

Abstract. In this paper, a novel adaptive noise reduction method for engineering drawings is proposed based on the assessment of both primitives and noise. Unlike the current approaches, our method takes into account the special features of engineering drawings and assesses the characteristics of primitives and noise such that adaptive procedures and parameters are applied for noise reduction. For this purpose, we first analyze and categorize various types of noise in engineering drawings. The algorithms for average linewidth assessment, noise distribution assessment and noise level assessment are then proposed. These three assessments are combined to describe the features of the noise of each individual engineering drawing. Finally, median filters and morphological filters, which can adjust their template size and structural element adaptively according to different noise level and type, are used for adaptive noise reduction. Preliminary experimental results show that our approach is effective for noise reduction of engineering drawings.

Keywords: Adaptive Noise Reduction, Engineering Drawings, Linewidth Assessment, Noise Assessment.

1 Introduction

Noise reduction is a fundamental problem ([1], [2], and [3]) of image processing and pattern recognition, which attempts to recover an underlying perfect image from a degraded copy. It plays an important role in automatic engineering drawings analysis since engineering drawings are usually scanned from paper drawings or blueprints, in which many factors may generate noisy document images. The noises in engineering drawings can be in different types and levels, which greatly affect the results of vectorization, recognition, and other processing, and hence, dramatically reduce the overall performance of engineering drawings analysis.

Current approaches to noise reduction can be broadly classified into order statistical methods, transform domain methods, and fuzzy methods. In order statistical methods, median filter [4] and rank order filter [5] are representatives, which use statistical theory to detect and reduce noise in images. Transform domain methods apply signal processing methods for noise reduction by using transformation methods, such as Fourier Transform and Wavelet Transform [6]. Fuzzy methods seek to use nonlinear filters and learning theories, such as fuzzy filters [7] and neural networks [8], to reduce noise.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 140 – 150, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Although many approaches have been proposed to various noise reduction problems, engineering drawings were not paid much attention to. Current approaches ignore the special features of engineering drawings and different types and levels of noise. They employ general image processing methods to reduce noise in engineering drawings. Although they do achieve some promising results, noise reduction for engineering drawings is still not always satisfactory.

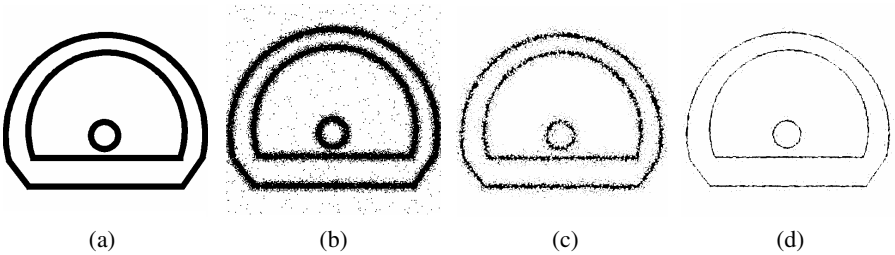
In this paper, we mainly assess the noise of engineering drawings with similar linewidths of primitives from two aspects: 1) the average linewidth of primitives, and 2) distribution and level of noise, based on which we can apply adaptive noise reduction. The arrangement of the paper is as follows: In Section 2, we analyze the special features of engineering drawings and categorize the noise into different types and levels. In Section 3, we present our linewidth assessment algorithm based on Medial Axis Transform. In Section 4, we discuss our methods used to assess noise distribution and noise level. The Adaptive Noise Reduction (ANR) method is proposed in Section 5. Some experimental results are shown in Section 6 and conclusions are shown in Section 7.

2 Features and Noise in Engineering Drawings

Engineering drawings have certain special features: 1) the possible linewidths are limited to several discrete values; 2) the edge of primitives (e.g., lines and arcs) is smooth; 3) the background and the primitives are monochrome. **Fig. 1** shows four engineering drawings with different qualities. From **Fig. 1**(a) we see that the linewidth of primitives is nearly equal and the edge of primitives is smooth. There is no noisy point on either background or primitives. However, the qualities of the other three are not so good due to different types and levels of noise.

There are various types and levels of noise in engineering drawings, as classified and modelled by existing researchers. Pavlidis [9] enumerated three types of distortion noise generated by scanners. Kannugo et al. [10] explored a nonlinear global and local document degradation model. Zhai et al. [11] summarized four types of common noise in engineering drawings (i.e., Gaussian noise, high frequency noise, hard pencil noise, and motion blur noise) and validated their models.

For binary engineering drawings, we categorize the noise into three basic types: 1) Gaussian noise, 2) high frequency noise, 3) hard pencil noise. In addition to types, the noise in engineering drawings can be at different levels, which indicate how noisy the images are. Next, we will discuss the assessment of image quality in terms of both primitives and noise.



(a) (b) (c) (d)

Fig. 1. Examples of engineering drawings

3 Linewidth Assessment of Primitives

In this section, we discuss the detail of our proposed method for linewidth assessment. As we mentioned previously, the linewidths of primitives, such as lines and arcs, in engineering drawings are limited to several values. We design our method based on the following assumptions: some engineering drawings have similar linewidths of primitives; they have moderate noise levels; the distance between primitives is usually much greater than their linewidths; the size of a noisy region is usually smaller than the average linewidth. Otherwise, it is difficult for our noise reduction methods (even human beings) to distinguish the gap between primitives and useful data from noisy data. Hence, linewidth is very important information to be used for both preserving useful features and removing noise.

We use a thinning algorithm based on Medial Axis Transform (MAT) [12] to calculate the average linewidth. MAT uses a recursive method to extract the skeletons of primitives from a binary image. In each iteration, the points satisfying certain conditions are removed from the primitives. The skeleton obtained by MAT consists of the set of points that are equally distant from two closest points of the boundary of primitives. Assume that the total number of iteration required is *I*, the linewidth after the i^{th} iteration is d_i, and the number of points that have just been removed from the primitives during the i^{th} iteration is N_i. The lines are thinned at both sides when $d_i > 2$, that is, $d_{i+1} = d_i - 2$. When $d_i = 2$, the lines are only thinned by one pixel, that is, $d_{i+1} = d_i - 1$, $i \in [1, I-1]$.

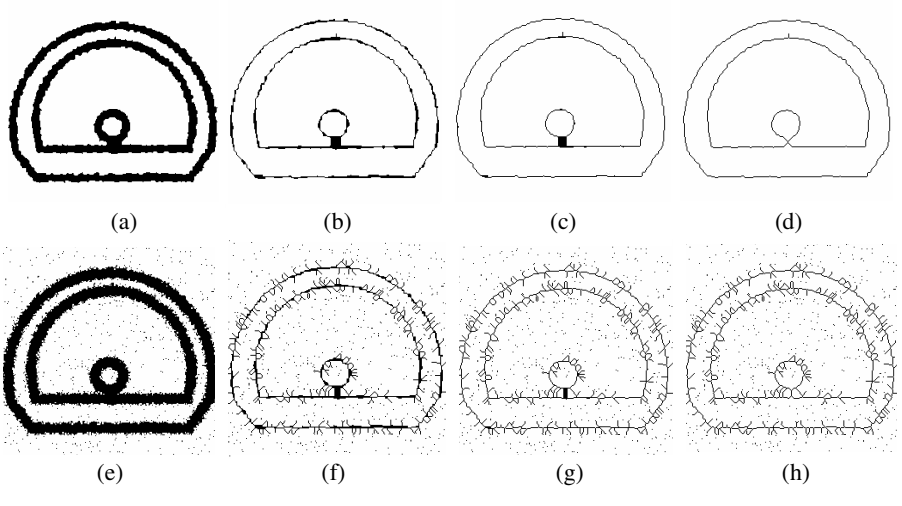


Fig. 2. Examples of thinning procedure. (a) is the original image; (b) and (c) are the thinned images of (a) after the 4th and 5th iteration, respectively; (d) is the skeleton image of (a). (e) is (a) with some noise added; (f) and (g) are the thinned image of (e) after the 4th and 5th iteration, respectively; (h) is the skeleton image of (e).

Obviously, N_i becomes smaller when i increases. Finally, when $N_i = 0$, the skeleton is extracted from the primitive successfully. As mentioned in Section 2, a characteristic of engineering drawings is that the linewidths are almost equal. It means that linewidths of most primitives become one pixel at the same time during the thinning procedure. Hence, in the first several iterations, the change of N_i / N_1 is small but at some iterations it dramatically drops. In **Fig. 2**, (a) is an original image, (b), (c) and (d) illustrate the thinned images of (a) at different iteration. In **Fig. 3**, (a) and (b) illustrate the curves of N_i / N_1 and $(N_i - N_{i+1}) / N_1$ during the thinning process of **Fig. 2**(a). We can see that N_i / N_1 has sharp drops at the 4th and 5th iterations. Correspondingly, nearly all lines become one pixel wide after the 5th iteration except the part where the circle and line touch together, as shown in **Fig. 2**(b) and (c). All the 6th to 11th iterations are used to thin this conjoint part only, whose width cannot reflect the real linewidths of primitives. Hence, the changes of N_i / N_1 between the 6th to 11th iterations become small and these iterations should not be taken into account when we assess the average linewidth of primitives.

According to the analysis above, we know that the bigger the change of N_i / N_1 at one iteration, the more lines reach one pixel wide at that iteration. When the change of N_i / N_1 is bigger than a threshold T_N, that is $(N_i - N_{i+1}) / N_1 >= T_N$, $i \in [1, I-1]$, we use $(N_i - N_{i+1}) / N_1$ and i to calculate the average linewidth of primitives. Let S ={ i | $(N_i - N_{i+1}) / N_1 >= T_N$ at ith iteration, $i \in [1, I-1]$ }. Assume || S ||=L. $S(l)$, $l = [1, L]$, is the lth element in S. Take **Fig. 2**(a) for example, if we let $T_N = 0.25$, $(N_i - N_{i+1}) / N_1 >= T_N$ when iteration times i=4 and i=5, as shown in **Fig. 3**(b), hence ||S||=2, S(1)=4 and S(2)=5. When $I = 1$, it means that the linewidth of primitives is already one pixel wide. When I=2, it means that the lines are only thinned once by either 1 or 2 pixels before they become one pixel wide, we use average value 1.5 pixel to indicate it. Of course, the finally obtained skeleton is one pixel wide. Hence, the linewidth of primitives is 1.5+1=2.5 pixels and the possible error is less than only 0.5 pixel. When I>2, we can use following equations to calculate the average linewidth W_{line}:

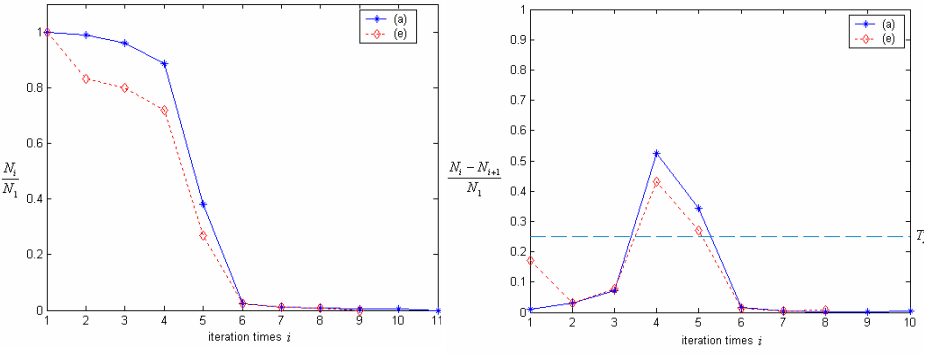
$$N_{sum} = \sum_{l=1}^{L} (N_{S(l)} - N_{S(l)+1}),$$

$$I_{avg} = \sum_{l=1}^{L} \frac{(N_{S(l)} - N_{S(l)+1})}{N_{sum}} \times S(l),$$

$$W_{line} = 2 \times I_{avg} + 1,$$

where, we first calculate I_{avg}, which is the average number of iterations the primitives have undergone. It is calculated as the weighted sum of all iterations which

result significant change of $N_i - N_{i+1}$, with an iteration's weight being the percentage of the removed noisy points at such iteration. The linewidth is just twice the average iteration number plus 1.



(a) (b)

Fig. 3. Curves of N_i / N_1 and $(N_i - N_{i+1}) / N_1$ of **Fig. 2**(a) and (e)

Meanwhile, the proposed linewidth assessment method is robust to noise, as we shown in **Fig. 2**(e)-(h). In **Fig. 2**, (e) is a noisy version of (a). We can see that most lines of (a) and (e) become one pixel wide at the same iteration and the curves of N_i / N_1 and $(N_i - N_{i+1}) / N_1$ of **Fig. 2**(a) and (e) are much similar, as shown in **Fig. 3**. The largest difference is caused by the noise which is thinned in the first several iterations. The average linewidths of primitives of **Fig. 2**(a) and (e) computed by the proposed method are 9.79 and 9.76, respectively.

Using the proposed method, when $T_N = 0.25$, the average linewidths of the four images in **Fig. 1** are 6.30, 5.78, 3.00, and 2.50, respectively. Experiments show that we can obtain more precise linewidths using this method.

4 Noise Distribution and Level Assessment

After we obtain the average linewidth, we need to assess the detail of the noise. Images (b), (c) and (d) in **Fig. 1** show some typical forms of noisy images. For this purpose, we describe the noise from two aspects: 1) noise distribution which is assessed by block method and 2) noise level which is assessed by signal to noise ratio.

4.1 Noise Distribution Assessment

In engineering drawings, there are mainly two kinds of distribution of noise: 1) the noise distributes evenly in the whole drawings, as shown in **Fig. 1**(b); 2) the noise mainly distributes at surrounding of the primitives, as shown in **Fig. 1**(c). We call them as TYPE I and TYPE II respectively. In this paper, we use block median filter to

distinguish these two types of noise. We divide the document image into local blocks by the size about 10×10 pixels, as illustrated in **Fig. 4**. Because we only need to detect noise rather than to remove noise at this stage, we use a 3×3 median filter to detect noise in all blocks one by one. When a noisy point is removed by the median filter in a block, this block is a noisy block. Assume there are $M\times N$ blocks in one image, among which Z blocks are noisy. We can calculate the distribution of the noise D_{noise} as follows:

$$D_{noise}=\frac{Z}{M\times N}.$$

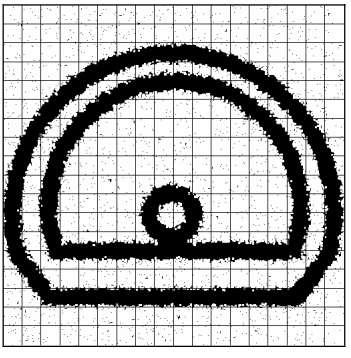


Fig. 4. The block method for analysis of noise distribution

Given a pre-set threshold $T_{distribution}$, the noise type is TYPE I if $D_{noise}>=T_{distribution}$, and TYPE II otherwise. For **Fig. 1**(b) and (c), if let $T_{distribution}=0.5$, the obtained values of D_{noise} are 0.7769 and 0.4250 respectively. This means that the noise distribution of **Fig. 1**(c) (TYPE II) is more concentrative than that of **Fig. 1**(b) (TYPE I).

4.2 Noise Level Assessment

Next, we assess the noise level. For different noise level, we should use different de-noise method to obtain the best quality, because improper use of noise filter can reduce both noise and useful information of primitives greatly.

We use the signal to noise ratio (*SNR*) to describe the noise level of an image. We employ a median filter whose template size is $1.5\cdot W_{line}\times1.5\cdot W_{line}$ to compute *SNR*. Such filter can reduce noise while preserving the primitives. Assume the primitives to be black and the background white. First, we count the number of all black pixels in the image and denote it as Q. Then the median filter is used once to wipe off noise and we count the number of the remaining black pixels again. We denote this number as P. P is the number of primitive points and reflects the signal

level. $Q-P$ is the number of noisy points that have been removed by the filter and reflects the noise level. If $Q-P=0$, it means that there is no noise in the image. When $Q-P\neq 0$, we define the SNR of an image as:

$$SNR=\frac{P}{Q-P};$$

Usually, lower *SNR* means higher noise level. For instance, the *SNR* of **Fig. 1**(b) and (c) are 2.399 and 1.443, respectively. It means that the noise level of (c) is higher than that of (b). However, there is another form of degradation of engineering drawings, as shown in **Fig. 1**(d), where the primitives are too thin and discontinuous. When the median filter is applied, the primitives are also regarded as the noise and therefore wiped off from the image. As a result, its *SNR* is very small (only 0.285). For these different cases, different methods should be employed for noise reduction, as we will explain in the next section.

5 Adaptive Noise Reduction

Many techniques for noise reduction replace each pixel with certain function of the pixel's neighborhood. Because useful features and many noises usually have common frequency components, they are not separable in the frequency domain. Hence, linear filters tend to either amplify the noise along with useful features, or smooth out the noise and reduce useful features simultaneously.

To minimize the conflict between useful features and noise, researchers have introduced a number of adaptive noise reduction algorithms, which essentially attempt to preserve or amplify useful features while reducing noises. Median filter and morphological filters are, perhaps, the most well-known and popular filters for adaptive noise reduction. The median filter is very good at reducing some types of noise (e.g., Gaussian noise and "salt and pepper" noise), while preserving some useful features (e.g., edges). It is not so good, however, at removing dense noise, and it degrades thin lines and those features smaller than half the size of its template. The morphological filters include erosions, dilations, openings, closings, and their combinations. The action of a morphological filter depends on its structural element, which is a small pattern that defines the operational neighborhood of a pixel. The effectiveness of the median filters and morphological filters greatly relies on the size of the template and the structural element. Hence, it is very important to choose them carefully.

Based on the assessment results of primitives and noise we obtained in Section 3 and 4, we develop an adaptive noise reduction (ANR) method. We choose the median and morphological filters to reduce noise and also adjust the size of the template and the structural element adaptively according to the assessed linewidth and noise information. Let W_{line}, D_{noise} and SNR denote the linewidth, noise distribution and noise level of one image, W_{ideal} is a given linewidth, $T_{distribution}$ and T_{level} are pre-set

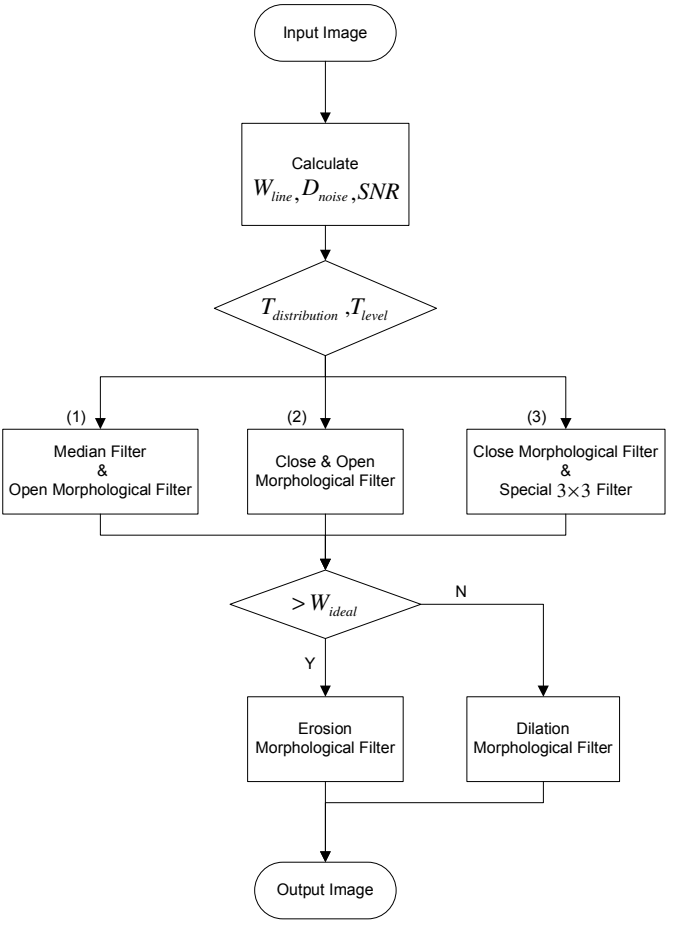


Fig. 5. The flowchart of ANR

thresholds for D_{noise} and SNR, d_{SE} is diameter of the circle structural element. (1) If $D_{noise} >= T_{distribution}$ and $SNR >= T_{level}$, the main noise is Gaussian noise combined with some high frequency noise, we first use a median filter with a $1.5 \cdot W_{line} \times 1.5 \cdot W_{line}$ template to remove Gaussian noise. Then an open morphological filter with a circle structural element, $d_{SE} = 0.8 \cdot W_{line}$, to reduce high frequency noise and smooth primitives. (2) If $D_{noise} < T_{distribution}$ and $SNR >= T_{level}$, the noise distributes surrounding the primitives concentratively and the main noises are hard pencil noise and high frequency noise combined with some Gaussian noise. Hence, we use a close morphological filter with a circle structural element, $d_{SE} = 0.5 \cdot W_{line}$, to remove gaps caused by hard pencil noise in primitives

and an open morphological filter with a circle structural element, $d_{SE} = 0.8 \cdot W_{line}$, to reduce high frequency noise and dense Gaussian noise and smooth primitives. (3) If $SNR < T_{level}$, it means the primitives are too thin and maybe discontinuous. In this condition, we first use a close morphological filter with a circle structural element, $d_{SE} = W_{line}$, to connect primitives, then in order to avoid losing useful information, we apply a special 3×3 filter to remove noise, which, for a binary image, can change the value of the centre element only when the values of all other 8 neighbour elements are different from it. In this way, all single noisy points can be removed while the primitives can be preserved, even they are one pixel wide.

After removing the noise from the image, according to W_{line}, we use an erosion or dilation morphological filter to adjust the linewidth to the given width W_{ideal}, so that all de-noised images may have similar linewidth to the original noiseless images with W_{ideal} being their ideal linewidth. **Fig. 5** shows the flowchart of our ANR method.

6 Experimental Results

We have implemented a prototype system based on our proposed method. We use some noisy images of engineering drawings chosen from the Symbol Recognition Contest of GREC'03 [13] for testing. **Fig. 6** and **Fig. 7** show the experimental results of four images. In **Fig. 6**, the top row contains the images with different types and levels of noise and the bottom row are the results of our adaptive noise reduction approach with $T_N = 0.25$, $T_{distribution} = 0.5$, $T_{level} = 1.0$ and $W_{ideal} = 5$.

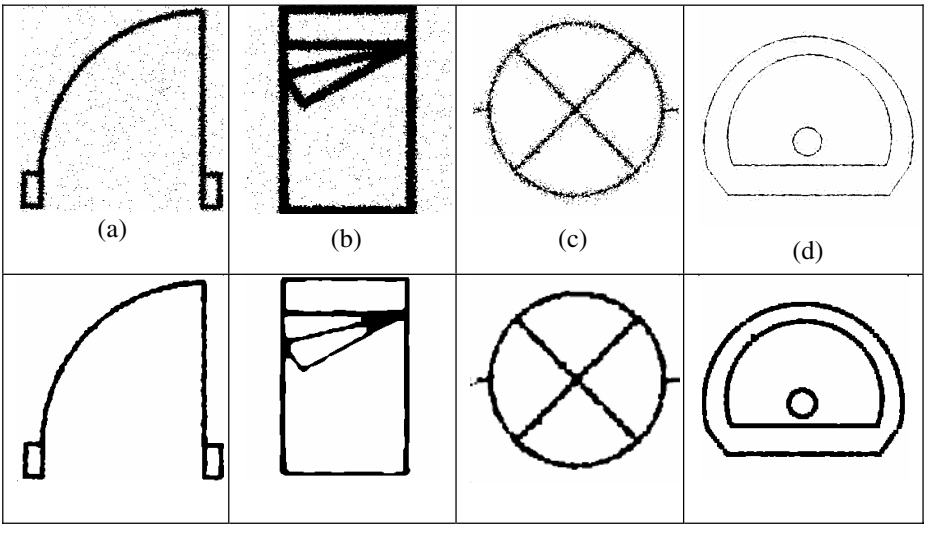
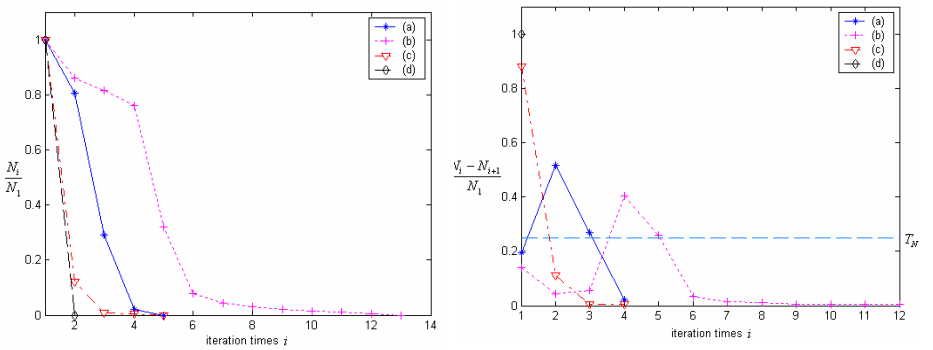


Fig. 6. Comparison between original images and de-noised images



(a) (b)

Fig. 7. Linewidth assessment of the top four images of **Fig. 6**. (a) shows the curves of N_i/N_1 and (b) shows the curves of $(N_i - N_{i+1})/N_1$. Note that there are only two iterations for **Fig. 6**(d), hence there is only one point for it in **Fig. 7**(b).

Fig. 7 includes curves of N_i/N_1 and $(N_{i-1} - N_i)/N_1$ of the four images. We can see that there is a sharp drop on each curve, where the ordinal number of the iteration reflects the linewidth. **Table 1** shows the results of W_{line}, D_{noise} and *SNR* calculated by the proposed method. From the experimental results, we can see that our proposed methods can effectively reduce most noise in engineering drawings while preserving the useful features (e.g., smoothing edges of primitives and adjusting average linewidth). These noise reduction results provide us a good basis for vectorization and recognition of the contest symbols.

Table 1. The results of noise assessment of four images in **Fig. 6**

	W_{line}	D_{noise}	*SNR*
Fig. 6(a)	5.70	0.646	3.678
Fig. 6(b)	9.77	0.648	7.937
Fig. 6(c)	3.00	0.386	1.414
Fig. 6(d)	2.50	0.229	0.285

7 Conclusion and Future Works

In this paper, we analyzed the special features and various types and levels of noise in engineering drawings and proposed an Adaptive Noise Reduction (ANR) method based on linewidth assessment, noise distribution assessment and noise level assessment. Compared with other noise reduction method, the proposed method can adjust the template size of median filter and structural element of morphological filter adaptively according to different types and levels of noise. The method can remove the noise while keeping the useful information of primitives. Experimental results proved effectiveness of our proposed methods.

However, some problems still need to be solved. One problem is how to deal with primitives with various linewidths in a single engineering drawing. It is possible to use the curve in **Fig. 3**(b) to detect the dominant linewidths by detecting the peaks. Once we know the candidate linewidths, we can focus our work on adaptive adjustment of parameters of the proposed method for different primitives according to their linewidths in one engineering drawing. Another problem is how to smooth or sharp edges further while keeping much smaller features of primitives. We will continue our research on these problems and enhance the performance of our proposed adaptive noise reduction method for engineering drawings.

Acknowledgement

The work described in this paper is fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 1147/04E].

References

[1]. H. C. Andrews, "Monochrome Digital Image Enhancement", Applied Optics, Vol. 15, No. 2 (1976) pp. 495-503.
[2]. A. Lev and S. W. Zucker, and A. Rosenfeld, "Interactive Enhancement of Noisy Images", IEEE Trans. on Systems, Man and Cybernetics, Vol. 7, No. 6 (1977) pp. 435-422.
[3]. G. A. Mastin, "Adaptive Filters for Digital Image Noise Smoothing: An Evaluation", Computer Vision, Graphics and Image Processing, Vol. 31, No. 1 (1985) pp. 103-121.
[4]. J. Ishihara, M. Meguro, and N. Hamada, "Adaptive Weighted Median Filter Utilizing Impulsive Noise Detection", Application of Digital Image Processing, Proc. SPIE 3808 (1999) pp. 406-414.
[5]. M. Miloslavski and T. S. Choi, "Application of LUM Filter with Automatic Parameter Selection to Edge Detection", Applications of Digital Image Processing, Proc. SPIE 3460 (1998) pp. 865-871.
[6]. H. Oktem, K. Egizarian, and V. Katkvnik, "Adaptive De-noising of Images by Locally Switching Wavelet Transforms", ICIP (1999).
[7]. F. Russo and G. Ramponi, "A Fuzzy Filter for Images Corrupted by Impulse Noise", IEEE Signal Processing Letters, Vol. 3, No. 6 (1996) pp. 168-170.
[8]. H. Kong and L. Guan, "A Neural Network Adaptive Filter for the Removal of Impulse Noise in Digital Images", Neural Network, Vol. 9, No. 3 (1996) pp. 373-378.
[9]. T. Pavlidis, "Recognition of Printed Text Under Realistic Conditions", Pattern Recognition Letters, Vol. 14, No. 4 (1993) pp. 317-226.
[10]. T. Kanungo, R. M. Haralick, and I. Phillips, "Global and Local Document Degradation Models", Proc. ICDAR (1993) pp. 730-734.
[11]. Z. Jian, L. Wenyin, D. Dori, and L. Qing, "A Line Drawings Degradation Model for Performance Characterization", Proc. ICDA (2003) pp. 1020-1024.
[12]. J. C. Mott-Smith, "Medial Axis Transforms", Picture Processing and Psychopictoris, Academic Press, New York (1970).
[13]. http://www.cvc.uab.es/grec2003/SymRecContest/images.htm, GREC2003 Symbol Recognition Contest.

Extraction of Index Components Based on Contents Analysis of Journal's Scanned Cover Page

Young-Bin Kwon

Department of Computer Engineering, Chung-Ang University
Seoul, 156 – 756, Korea
ybkwon@cau.ac.kr

Abstract. In this paper, a method for automatically indexing the contents to reduce the effort that used to be required for input paper information and constructing index is sought. Various contents formats for journals, which have different features from those for general documents, are described. The principal elements that we want to represent are titles, authors, and pages for each paper. Thus, the three principal elements are modeled according to the order of their arrangement, and then their features are generalized. The content analysis system is then implemented based on the suggested modeling method. The content analysis system, implemented for verifying the suggested method, gets its input in the form containing more than 300 dpi gray scale image and analyze structural features of the contents. It classifies titles, authors and pages using efficient projection method. The definition of each item is classified according to regions, and then is extracted automatically as index information. It also helps to recognize characters region by region. The experimental result is obtained by applying to some of the suggested 6 models, and the system shows 97.3% success rate for various journals.

1 Introduction

Various studies on off-line character recognition have been done for the last 40 years. However, utilization of such technology is at a low level due to their difficulty in application. Analysis of the structure of a document performs the role of processing the characters through the recognizer after separating the character, graphic and picture. The document structure analysis method recognizes the characters from a document and stores it as a coded data and the others, picture and graphic part, in a compressed format [1].

Such analysis method is required nowadays to digitize the mass amount of data. However, analysing of a random type of document is a hard task with the current technology and therefore is limited to certain types of documents. The analysis of the two major types of document, journal and newspaper, is still a hard task and researches are being done in this area [2].

In this paper, we will conduct a study on the recognition module of the index, which is a part of a web-based paper database system used for the storage, management and analysis of research papers. The database of academic paper

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 151 – 161, 2006.
© Springer-Verlag Berlin Heidelberg 2006

generally consists of an input, storage, search and output part, of which the input part is the most complex and time-consuming part. The input of paper information can be processed manually, scanning of already published papers, and automatically recognizing published papers. The manual method shows the least error rate since it is typed in manually. However, this method is too time and effort consuming. The method using a scanner is the least effort-consuming method, but cannot be used for the searching and taking a part of the paper. Finally, the method through automatically recognizing the scanned image can utilize the advantages of the above two methods and therefore is considered to be the most effective method. However, due to the various types of journals and documents, it requires a high-level technology and also is time and effort consuming when it comes to the part for constructing the database.

We have proposed a method for automatically creating the index to reduce the effort. The various formats are sorted for generalization and then analyzed for each format to be used as the basis for the recognition process. From this 8-bit gray image, the connected components are extracted and projection is processed for each area in our method. The outline information extracted from the input image is analyzed to classify the title and the graphic images and form the text line from the characters. Projection and analysis of the extracted text line to form a block of character with a meaning for the analysis of the index. The image and graphics are separated without any information and therefore are processed as itself. The title, author and page information are separately stored. In the final stage of analysis, verification by the form of the journal is processed to allocate the blocks of characters according to their meaning.

2 Analysis of Previous Researches

Fletcher and Kasturi utilized the Hough Transformation method for grouping the connected components of the input image into a logical text line, and suggested a method for distinguishing graphics and characters based on this method [3]. It was not affected by the skew by using the Hough Transformation method and was able to group the slanted text lines within the images. However, this method does not deal with the touching character problem, has many limitations on the document being used and can only separate the characters and graphical images. Wahl suggested a method of separating character, line graphic and half tone images using the Run-Length Smoothing algorithm [4]. This method distinguishes the graphic and text line from the separated block using a few characteristics. This method has a fast document processing speed by using the run-length smoothing method and adopts a rather simple method for distinguishing the character blocks and the graphic blocks. This method is difficult to obtain a good result if the space between the connected components is small. Hirayama suggested a method that mixes the run-length smoothing method and the connected component analysis method [5]. The text lines are first checked using the run-length smoothing method and the characteristics obtained through this process are used to form the group of characters. Tsujimoto also adopted the run-length smoothing method to analyze the document before separating the connected characters in his method suggested [1]. This method adopts the run-

length smoothing method to form the connected components before separating them into candidates according to the conditions. This method shows a rather short run-length which makes it applicable on complex documents.

Studies on document recognition techniques are limited to certain areas due to the special characteristics of each document. The studies we have looked at are limited to classifying the charts, images and character group of a document. However, even with these methods, it was impossible to recognize the meaning of each area and was inefficient when applied to documents with irregular shapes. Therefore, we have limited the object of classification to the table of contents of a journal of cover page to analyze the meaning of each area element, and used the information as a basis for the character recognition in order to extract the text line candidate along with their meaning.

Many researches on table of contents (TOC) are done [6-9]. Part of speech tagging, a labeling approach, for automatic recognition of TOCs is done by Belaid et al[6] utilizing a priori model of the regularities present in the document structure and contents. Mandal extract the structural information using a priori knowledge from the TOC develop digital library in order to identify the 137 different TOCs[7]. Tsuroka et al. proposes a method of image based structure analysis and conversion of TOC of books into XML document [8]. Lin introduces a method to detect and analyze TOCs based on content association. The associations of general text and page numbers are combined to make the TOC analysis move accurate. The researches mentioned above are all OCR based approach in order to obtain more accurate results. The approach of Belaid and Lin show an idea to develop the analysis of content information from the journal tables. To simplify their approach, they used the OCR result of pages numbers to identify the fields. In this paper, we propose a method of no OCR based content analysis system i.e., graphic mode, and compare with the OCR based system.

3 Modeling of the Table of Contents of a Journal

The forms of the journal differ by the organization publishing it. The form of the table of contents is as various as the journal and therefore almost impossible to generalize it. In analyzing the table of contents, it is essential to analyze the objective of it. There are many obstacles such as images, logos or advertisement of the publishing organization on the table of contents. However, they all contain information of the journal such as the title, author name and the page number the article is listed. Since the main objective of this study is to effectively extract this information, we have to analyze the format based on these facts. We will first analyze the table of contents of the journals which specializes in the area of computer science. The contents of the text line are listed in the order of a dotted line, author name and page number. This type of structure is not easily analyzed as it seems using the existing analysis tools, which is why structural analysis cannot be generally applied in all areas. The computer will recognize the different columns such as the title and author name, visible to the human eye, as simply pixels with different values. The ordinary structural analysis systems would simply classify the text line, images and graphic area and chart area and then apply the off-line character recognition on the text line area. However, we would get a totally different result of analysis with the ordinary

method. The title of the article or the name of the organization are not information that is essential and take on the form of a banner, and therefore would not have a big effect on the result of our analysis even if recognized as graphical areas. However, this area actually contains characters, dotted lines, another character and numbers all mixed together. This type of format would be almost impossible to analyze for the recognition system, although visible to the human eye. One of the reasons is the existence of the dotted line. A series of dots are not generally ground on normal documents. Indicators such as ',' and '.' may appear on text lines, but they only appear independently as an indicator or division and do not appear consecutively. Therefore, the dotted lines will be recognized as a graphical area. In our system, such dotted lines will be recognized as a special character for dividing the document in the structural analysis. The title part of the document is an easy part for the analysis. It can be recognized as an image located between paragraphs. The right hand part of the dotted lines can also be distinguished as a candidate for a text line. However, there still lies a problem in the characteristic of the text line including the author information.

In the contents part, the sub topic area can be simplified. The starting point is the same as the other text lines as well as the font. However, the indicator has a strong characteristic which makes its length relatively short compared to other text lines. The text lines containing the paper information have a fixed location for the starting point as well as the end point. Therefore, the length of the text line is about 2/3 of the others, and the width of the end point is also about 2/3 of the whole for the sub topic area. The text lines for the actual table of contents show a fixed pattern. If all the information can be fit into a single text line, the starting point of each text line is the same as is the end point. Therefore, we would acquire a regular value for the length between the starting point and the end point. The text line containing the information about each paper can be divided into specific terms. The title, author name, and the number of page are some of them. These terms are the objects to be used for the indexing through our analysis process. Even the result we would acquire is far from the term extraction we would like to achieve and would require a thorough analysis of its structure. Irregular shaped identifiers exist between terms such as the title, author and page number which makes it difficult to recognize using the computer. The order of each term is also unfixed and therefore we are going to limit the objects to be analyzed to the title, author and page number terms.

4 Implementation of the Table of Contents Recognition System

The whole structure of the suggested system is shown in Fig. 1. The input image is divided into the cover and graphic area for separate recognition and the analysis of the table of contents is processed thoroughly to extract all the terms from the text line.

The noise is removed in the first stage and the horizontal projection is applied on the binary file as the second stage. We can acquire the line distribution through this process. Then vertical projection is applied to the line extracted to acquire the jaso and syllable distribution. The document is segmented to leave only the table of document as the object to be recognized and detailed analysis is processed to acquire the connected components. Area extension and merging is processed to extend the area. The characteristics will be applied to these groups to acquire the blocks of

candidates. This enables the classification of the candidate blocks through applying the models. A list for the Title, Author, and Page is made for the recognition and its result is used for the indexing of paper information. Many scanners support the auto-feed function and skew angle becomes very small. Thus, skew correction is not considered in this paper.

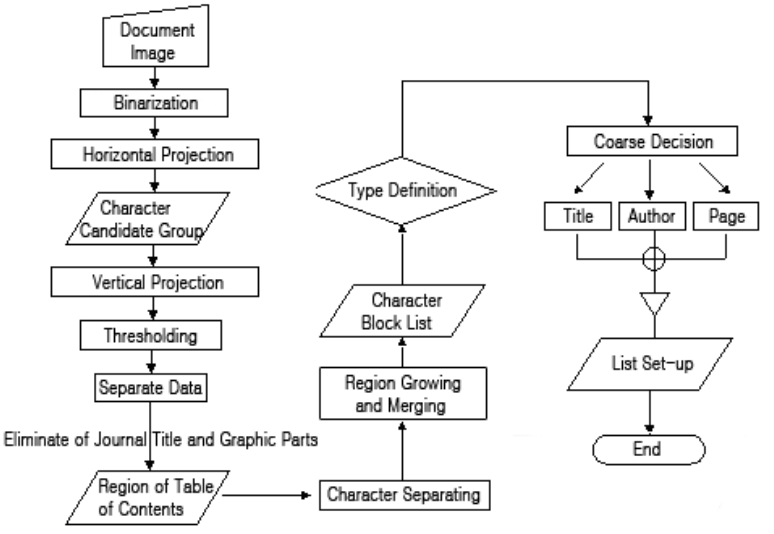


Fig. 1. Overall structure of Recognition System

4.1 Line Candidate Extraction

In the second stage, the candidates for the structural analysis are extracted from the group of candidates. Horizontal Projection is applied to the pre-processed image to acquire the accumulated distribution. Horizontal projection is processed to acquire the location of the candidates. The line extraction is even possible in the case of table of contents. The outline information required for the structural analysis is acquired through this process. The part where the resulting value exceeds the threshold is analyzed as text lines and the rest is treated as just blank space. The skew should be minimized because the result is sensitive to it. The threshold value is applied based on horizontal projection to store the values from the candidate groups. The reason for applying the threshold value is to remove the unnecessary noise. Projection is applied vertically this time on parts of the image. We will perform vertical projection on the area acquired by the horizontal projection. In other words, vertical projection is applied for a secondary analysis on the line areas. By applying a calculation on the values acquired by the vertical projection, we can classify the jaso and syllable for the pixel distribution.

Horizontal projection is applied to each line to locate the characters within the text line. The classification is processed by dividing the area into having a value 0 or others. All the values with even a single pixel must be found in order to extract each syllable area. Little special characters such as a colon are important factors in

analyzing the shape of the table of contents. Even after the second stage of analysis, we cannot verify whether the special characters extracted are colons or comma. By indexing the division line, the terms 0-1, 2-3, 4-5, … form the outline on each side. Since the division line becomes the top and bottom outline, connected component or run-length smoothing method is not required to acquire the rectangle for the area.

4.2 Area Expanding and Element Analysis

It is difficult to acquire the necessary information from the syllable, and therefore we have to merge or extend them in order to acquire the text line candidates. The distance between the areas should be used as the basis for the extension. The average distance between each line is acquired and if the distance is smaller than it, the area is merged based on this distance. The areas that are a distance away from the front or rear side are discarded to remove the noise. Through this process, alphabet based classification is processed. This information can be used as word information during the character recognition. Instead of applying the connected component extraction method or run-length smoothing for acquiring the outer rectangle, we applied the projection twice in order to maintain the accuracy and save time. The time consumed by this process turned out to be about 1/20 of that using the connected component extraction method and 1/6 of that using the run-length smoothing method.

The horizontal direction rules are applied to the result for re-extension and merging to apply the connected of elements to the group of blocks and save them on a list. The meaning of each candidate area is acquired by applying a structural analysis. The pattern must be analyzed in order to classify the blocks containing a meaning. The area formed by the extended outer rectangles must be connected as elements and stored in a list. The first major characteristic found in a table of contents is the starting location of the text line. The starting location is always the same when the contents are assumed to be expressed on a single line as is the end point. We can judge that it is about a single paper based on this, and if it is written inside, we can assume that it is related to a block above or below.

The location and the font size is analyzed first in order to find out the sub topic area. Most sub topic areas use the italic or gothic font and use a font larger than the other text to distinguish it from the title. It is also located in the front-hand side. The sub topic area is first extracted based on these facts. The type of font being used cannot be distinguished using the projection method adopted in our research. Therefore, we have to use the pre-acquired font size and location information. Each line has a certain height in the table of contents. The area containing a height which exceeds this limit is distinguished as a candidate and the location is acquired through horizontal projection. It is classified as a sub topic area when the size is larger than this value and the location is relatively in the front of the area.

The exceptional features such as the dotted lines or blank spaces become an effective identifier in our system. The block is divided into two sides based on the line or blank before dividing it into the front and back side. The front side can be classified as the title, back side as the author and page or vice versa. This of course, is assuming that the terms are divided into three. Therefore, if only two terms are printed or if there are more than four, the recognition turns out a different result. In case of T-A-P, the front side would be T-A and the backside P or just the T in the

front and A-P in the back. Based on this assumption, it is decided whether the area exists on the block that can be classified. Generally, a blank space exists between words and the block size of the author area is larger than the others on the same page which can be used for the analysis for the classification. If the adjacent areas that satisfy the following conditions, area R[i] becomes the page number and R[i-1] becomes the author. If the area is located on both pages, the location based on the starting point is used to find out the division between the areas. The location of the text line is located near the block that it belongs to even if there is a line change in between. Based on this fact, the location information is used for merging with either the line above or below. If the rule applied for merging is irregular or outside the scope assumed in this paper, it would be difficult to classify the area.

4.3 Form Verification

A pre-layout is extracted after the area analysis process. The characteristics of the analyzed candidates of blocks are used to decide the model for the table of contents. After the model has been decided, the characteristics are applied for the specifics and during this process, the order of the terms are used for further correction of the models. The final-layout is extracted after the decision on these terms. The meaning has been given for each block based on the contents of the text line, which means that they have to be listed as an element for each term for application in the process of character recognition and in indexing of the journals.

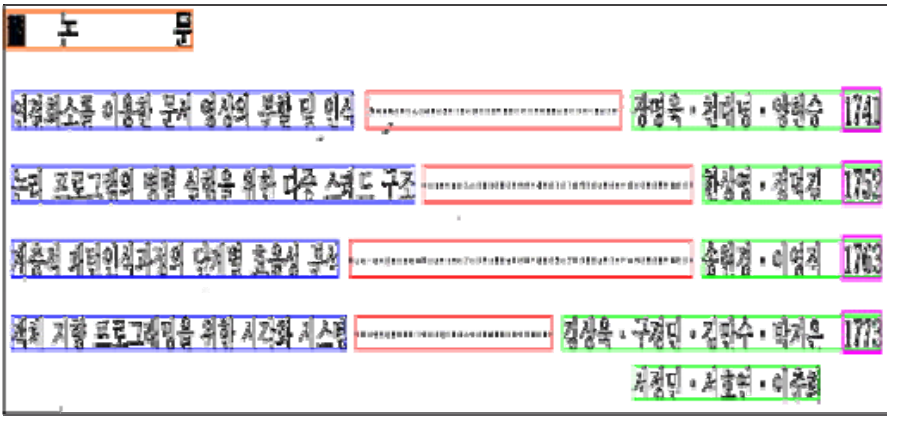


Fig. 2. Structure Analysis of Table of Contents

4.4 Coarse Detection for TAP Type

Despite having divided the work following the table type, there are still many different ways to place entry components on the page. In Fig. 3 are displayed some examples of layouts we might encounter for the TAP type.

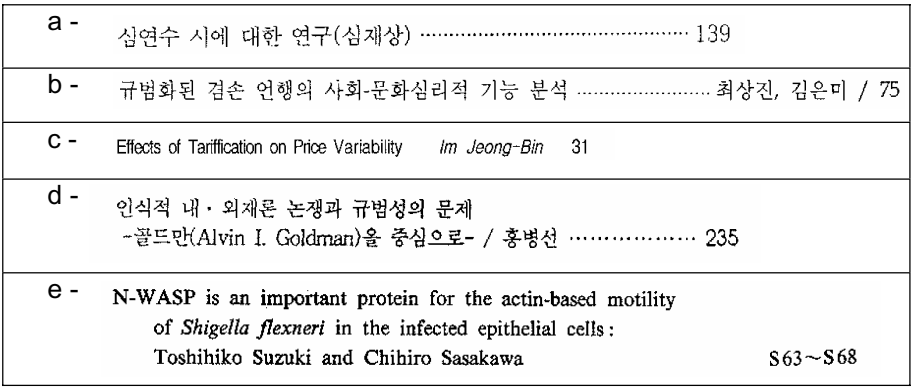


Fig. 3. Examples of different TAP types

As one may have noticed, titles, authors or pages are sometimes only separated by spaces, some other employ special characters, such as slashes (Fig. 3-b and 3-d), two points (Fig. 3-e), etc. We can group them in two classes: single or double symbol separators. In fact, we may classify the tables following the subtypes:

T1 T2 T3 : subtype1
T1 - T2 T3 : subtype2
T1 T2 - T3 : subtype3
T1 - T2 - T3 : subtype4

The routines detecting separators in the page will find their use in this part. We need to find which separators are effectively used in the table. We first need to know the number of entries in the table. There is most of the time less entry than regions of interests. Indeed, in the case of TAP, some long titles can take a whole line, pushing the authors and page numbers to the next ROI (see Fig. 3-d). One way to find the number of entries is to count the lines having a page number. And that's the reason why we separated the layout types in two groups: we needed the page number to be on the left or on the right side of the table to spot them efficiently. In order to find whether a line has a page number, a module is implemented. You may also look at Fig. 4 to ease the comprehension. Let's take the TAP or ATP layout type: the program basically compares the last character's position and the table's right margin. If they are close enough, then the ROI contains a page number, and we increment the entry counter by one. Knowing which are the ROIs containing a page number will be decisive in the next step, and the index of these precise ROIs is thus temporarily stored.

A loop is performed over the blocks to count the number of separators of each type. We then call the dominant separator the one having the most occurrences. If a separator is effectively used in the table, then there is at least as much separators as entries. But to compensate separators detection errors, the program uses a threshold value based on the entry number. If this condition is fulfilled, the dominant separator is set as the effective table separator, meaning that it is the one used throughout this table. Of course, if this condition is not satisfied, we fall in the subtype1, and the

Fig. 4. Position of the threshold value for the function counting entries

classification will use another method. There is a condition on the separators' position in the page, to make the difference between subtype2 and subtype3. Indeed, in the case of the TAP type for instance, we have two options: T-A P or T A-P.

Finally, the last tool we need to start the classification is based on the spaces. Entry components are often separated by space. In the case of *subtype1*, where there are no separators in the table, we may notice larger spaces between them. For a given ROI, the two biggest spaces have a very strong probability to be the ones separating the components. A procedure which saves the two biggest spaces of each ROI is subsequently added, based on the computation of the distance between consecutive vertical gray-lines.

5 Results and Analysis

For the test, the 12 different kind of journals including Korean and English journals, which can be classified as T-A-P have been used for the structural analysis. The extraction of the title, author and page has been tested on these journals to test the performance of our system. Based on the analysis method suggested in this research, the result of the test is shown in Table 1. Comparing with the results based on OCR and association proposed by Lin [9], he accomplishes 94% from 10 different journals. Our method shows more accurate analysis results.

Table 1. Experimental results

Number of entries	Analysis results	Success rate (%)
263	256	97.3

The following problems were found in the table of contents where the analysis has been unsuccessful.

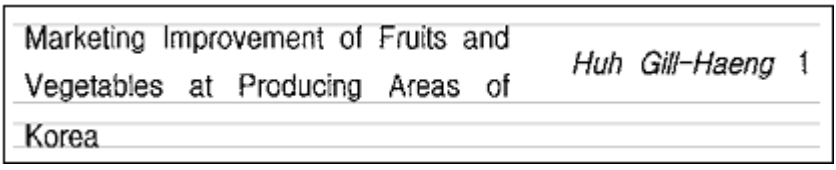


Fig. 5. Example of ROI extraction problem with non-vertically aligned text

- Most problems are caused by the page layout: Images, lines, and text enhancement (highlighted page numbers for instance) prevent the ROI extraction from functioning well. Therefore the whole analysis is jammed. The process may still be conducted well using the frame analysis feature, which gives good results in most cases.
- Non-vertically aligned text also cause troubles, for the first step in the algorithm is to separate white and non-white zones by drawing horizontal lines (Fig. 5 gives a good idea of the issue). Correcting this problem would lead to think of another way of extracting ROIs, and so to reengineer the entire algorithm. Fortunately, this case is very rare.
- A major issue concerns the analysis of dense text lines. Our algorithm uses spaces to determine where the limit of a zone is situated, but if text is too dense, the division might not operate correctly. There is no immediate solution to the problem, but the interface lets the user correct any wrong detection by selecting manually titles, author and page numbers.
- A part of the algorithm is dedicated to the suppression of any non-relevant ROIs (titles, header and footer text…). It is mostly based on the ROI average size. Sometimes, text that does not belong to the table may be kept for further analysis, because its size is the same as the table of content's.
- Some tables present really intricate layouts: some have their background colored with stripes; others have a colored background, which cannot be suppressed by a threshold. A treatment should be applied for each different case, based on the table specific properties. The major issue is that there are as many treatments to program as there are different layouts.

6 Conclusion and Future Works

Existing document analysis systems have not been able to effectively analyze the table of contents since they contained many exceptional features not found on any other documents. Many researches show the OCR based approach in order to simplify the problem. We have analyzed and listed the various characteristics of the journal's table of contents. The main element of the table of contents is the title, author and page. Therefore, we have made 6 different models with the combination of the three elements through generalization. Finally, we have applied projection to analyze the structural characteristics and classified each term based on these characteristics for extracting the meaning of each area. The meaning information can be automatically extracted from the graphical elements as the journal's index information to help the recognition as characters without OCR help.

The analysis system implemented in our research has an advantage over the other existing systems in that it can extract the abstract information by each term and can improve the success rate without any a priori knowledge by using the meaning information of each term before the actual analysis.

Since our system is only applicable on a few generalized models, future works should concentrate on extending the scope of the recognition to all models. It is difficult to perfectly analyze the various types of table of contents and is limited to some specific models. Also, the character recognition should be performed simultaneously and some methods to apply the statistical data on the result to improve the result can also be adopted in the process of analysis.

References

1. S. Tsujimoto and H. Asada, "Major Components of A Complete Text Reading System", Proceedings of IEEE. Vol.80, No.7, pp.1133-1149, 1992.
2. D. Wang and S.N. Srihari, "Classification of Newspaper Image Block Using Texture Analysis", Computer Vision, Graphics and Image Processing, Vol.47, pp327-352, 1989.
3. L.A. Fletcher and R. Kasturi, "A Robust Algorithm for Text line Separation from Mixed Text/Graphics Images", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.10, No.6, pp.910-918, 1988.
4. F. M. Wahl, K. Y. Wong, and R. G. Casey, “Block Segmentation and Text Extraction in Mixed Text/Image Document”, Computer Vision Graphics and Image Processing, Vol. 22, pp. 375-390, 1982.
5. Y. Hirayama, "A Block Segmentation Method for Document Image with Complicated Column Structures", Proceedings of the 2nd International Conference on Document Analysis and Recognition, pp.91-94, 1993
6. A. Belaid, L. Pierron, and N. Valverde, “Part-of-Speech Tagging for Table of Contents Recognition”, Proceedings of the International Conference on Pattern Recognition, pp451-454, 2000
7. S. Mandal, S.P. Chowdhury, A.K. Das, and B. Chanda, “Automated Detection and Segmentation of Table of Contents Page from Document Images”, Proceedings of the 7th International Conference on Document Analysis and Recognition, pp.398-402, 2003
8. S. Tsuruoka and C. Hirano, “Image-based Structure Analysis for a Table of Contents and Conversion to XML Documents”, Proc. DLIA Workshop, 2001
9. X. Lin and Y. Xiong, Detection and Analysis of Table of Contents Based on Content Association, Hewlett-Packard Technical Report, HPL-2005-105, May 31, 2005

Crosscheck of Passport Information for Personal Identification

Tae Jong Kim[1] and Young Bin Kwon[2]

[1] Department of Computer Engineering, Chung-Ang University
Seoul, 156 – 756, Korea
tjkim@cvlab.cau.ac.kr
[2] Department of Computer Engineering, Chung-Ang University
Seoul, 156 – 756, Korea
ybkwon@visionnet.cse.cau.ac.kr

Abstract. This paper proposes a character region extraction method and picture separation used for passports by adopting a preprocessing phase for passport recognition system. Character regions required in recognition make black pixel and remainder of the passport regions make white pixel in the detected character spaces. This method uses MRZ sub-region in order to automatically decide the threshold value of the binary image and this value is applied to the other character regions. Tthis method also executes horizontal and vertical histogram projection in order to remove picture region of the binary image. After the region detection of the picture area, the image part of the passport is stored in the database for face images. The remainder of the passport is composed of characters. The extraction of the picture area shows 100% of extraction ratio and the extraction of the characters for the recognition shows 96% of extraction ratio on ten different passports. From the obtained information, crosscheck process of MRZ information and field information of passport is implemented.

1 Introduction

A person's identification number, used at airports is on an increase through globalization and development of transportation. With the increase in the number of user, the time consumed for the immigration control judgment has also increased. Immigration control judgment is used to manage immigration and immigration person searching forgery passport possessor, emigration and immigration forbidder, wanted criminal, and emigration and immigration ineligibility persons such as alien. A passport recognition system is required to make these immigration control judgment more efficient and precise. Most existing passport cognition system extracts and recognizes the MRZ (Machine Readable Zone) code and the picture of the passport [1, 2]. MRZ code refers to the recognition code that expresses the substance of the passport substance by 44 characters per each line for passport recognition. However,

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 162 – 172, 2006.
© Springer-Verlag Berlin Heidelberg 2006

it is limited to distinguishing forged passport that recognize MRZ code. To recognize the MRZ code as well as the data, this paper proposes a method which extracts all characters to make binary image and correct picture region in passport image. If we can compare the data in passport with the MRZ code, we can improve the effectiveness of distinguishing forged passports.

Extent recognition system does not require special binarization method because it only recognizes MRZ code. However, binarization method that separates background and character in passport of color image is needed to recognize all data in passport. This paper proposes a binarization method which uses character RGB property and histogram of MRZ region. Extraction of picture region is proposed by a method which executes horizontal and vertical histogram projection and analyzes the result value to decide the top and left boundary as well as the height and width. Scanned image uses resolution of 200 dpi. After extraction of each character, a crosscheck of the traced characters is performed in order to compare corresponding information.

2 Information Extraction Process in Passport

The first process of extracting the information in passport makes the binary image of the interest region and must remove picture image among portion that is extracted in passport. After removing the picture image, we can extract the characters and the necessary information. The whole system for passport recognition is shown in Fig. 1. This paper deals with the character extraction and crosscheck routine without recognition process.

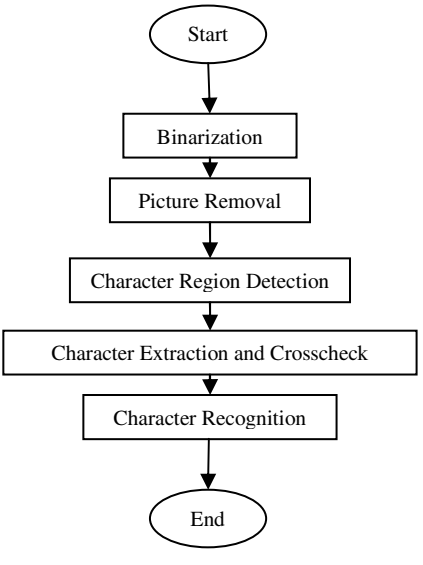


Fig. 1. Overall Process of Passport Recognition

3 Binarization of Passport Image

3.1 Binarization Phase

First of all, the RGB value of the character is analyzed for the whole image in order to make binary image. Making binary image is performed through two phases as shown in Figure 2 to establish a threshold value automatically.

Fig. 2. Two Phase of Binarization

3.2 Background Elimination Using Special Specificity of RGB

The R, G, and B value of character region has fewer than 200 in passport image and three values are considered of similar value similarity with black and white image. Also, the background of MRZ that is applied at the next phase has a value that is close to white. Each pixel is checked for value R, G and B and every pixel with a single value of more than 200 is judged as the background. The background of MRZ code region and extracted pixel is revised with white color of RGB (255, 255, 255) value.

3.3 Threshold Value Decision Using MRZ Region

Threshold value used for judging the character region must be decided to extract the character region in passing above the first phase. Threshold value decision that can judge character region used MRZ that exist in all passport of world. This paper decides threshold value searching for lower part line among MRZ code for two lines without using whole MRZ region and using that region. A method to set region is as following. The original image is shown in Fig. 3 and the result image in Fig. 4.

1. Run horizontal histogram projection according to Equation 1 from lower part of image
2. Decide i value that V[i] is greater than zero for the first time by start line and search again form i and decide j value that V[j] is equal to zero by end line.

$$V[i] = \sum_{j=0}^{width} f(j) \qquad \begin{cases} f(x) = 0 & if \quad color(x) = RGB(255,255,255) \\ f(x) = 1 & Otherwise \end{cases} \quad \text{(Eq.1)}$$

As ditto result, threshold value is established by R, G, B histogram calculation of each pixel except for the background region. If the threshold value is established to the maximum value, color pixel may still remain. Therefore, dwindle when accumulated value includes about 80% of whole pixel and is decided by threshold value. A threshold value, Rt, Gt and Bt, can be settle as 173. By last step, pixel that is RGB(Rt, Gt, Bt) in whole image is zero that express black pixel and remainder pixel is 1 that express white pixel. Fig. 5 shows the result of binary image of passport.

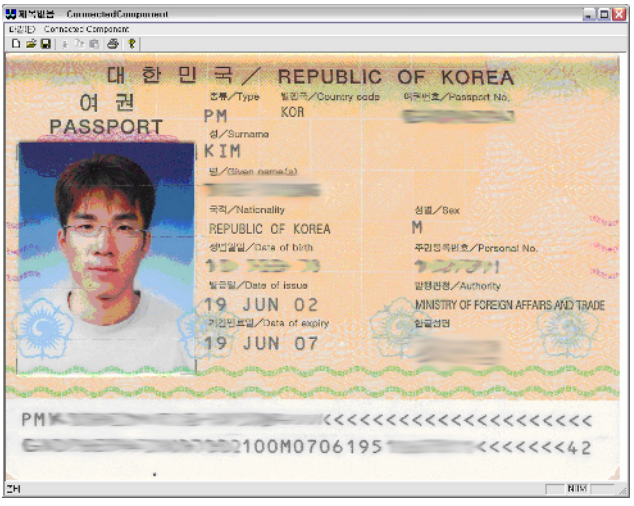


Fig. 3. Example of Passport Image

Fig. 4. Detected MRZ for Threshold Decision

4 Picture Removal of Binary Image

In Fig. 5, the result of binary image contains picture region and characters. Non-characters, such as picture region must be removed to extract the character region correctly. Several research results that separate character and non-character region are published [3-7]. In this paper, we use a method that uses the height and width of line after making word region using extension and reduction process of each connected components. The removal of the picture region is accomplished using the character

and non-character extraction algorithm [4]. A process that removes picture region is as follows:

1. Run horizontal histogram projection using Equation 1.
2. Decide upper boundary and height of picture using result value.
3. Run vertical histogram projection using Equation 1.
4. Decide left boundary and width of picture using result value.

4.1 Upper Boundary Detection of Picture Region

We trace a line that does not include black pixel using horizontal histogram projection result. Although fixed word space does not exist in passport, space exists in upper of

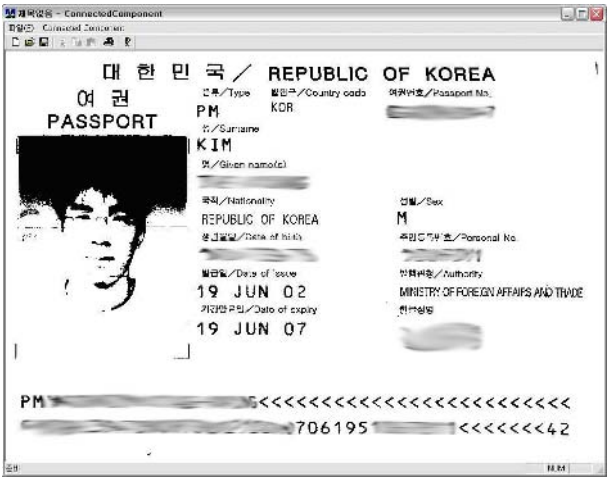


Fig. 5. Binary Image of Passport

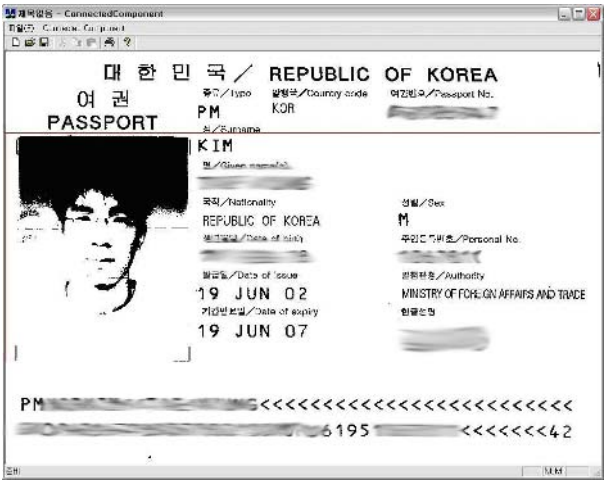


Fig. 6. Boundary of Picture Region

picture. Therefore, we have searched result V[i] value of projection result and find i that V[i] has zero. As can see in Fig. 5, part that V[i] has zero is three parts in upper direction of picture region. Part that can be decided by the upper boundary of picture region among these parts uses property that picture region is fixed in passport. It can be in position change according to scanned state, but picture region is keeping fixed height. Therefore, we can search for i that V[i] is zero in more than fixed height value h that is established by threshold value and decide that the i value is upper boundary. Fig. 6 shows the decided upper boundary.

4.2 Height Detection of Picture Region

If the upper boundary of picture region is decided, height of picture region can be decided from the upper boundary. First, we can deicide MRZ's upper boundary by the projection method. A picture region is located between two upper boundaries. Thus, picture region can be decided searching V[i] value for upside direction from MRZ and finding that V[i] value is non zero for the first time. Value that subtract i value from the upper boundary gets into height of picture region. Fig. 6 shows the decided height of picture region.

4.3 Left Boundary Detection of Picture Region

We can decide the left boundary using the result value of a vertical histogram projection. The left boundary can be decided searching V[i] value from left of passport image and finding that V[i] value is nonzero for the first time.

4.4 Width Detection of Picture Region

Decision of a width of picture region uses method to decide non-character region [4]. Since the upper/left boundary and height of picture region were decided before, we run vertical histogram projection within the region. Then, we examine result value

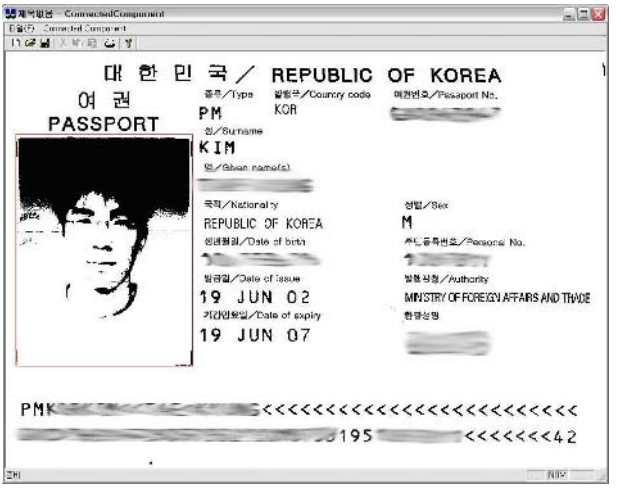


Fig. 7. Picture Region Detection

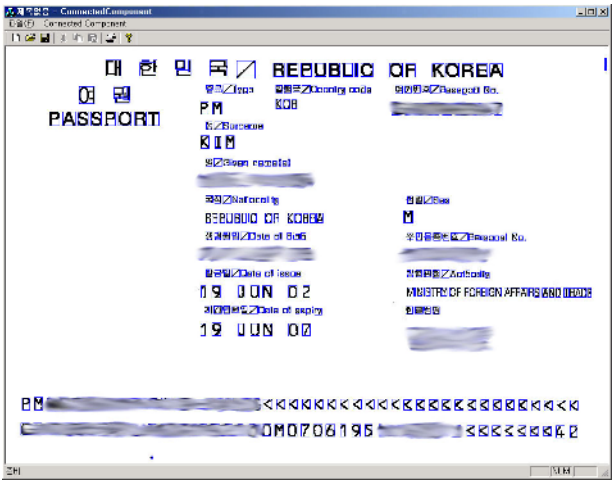


Fig. 8. Character Extraction of Passport Image

V[i] and calculate number that V[i] is non zero consecutively. And if V[i] is zero, we is fixed as in the case decide the upper boundary of picture. If i cost is smaller than w, we repeat same work from portion that next time V[i] is nonzero. The w is established 2/3 of general picture image width.

Fig. 7 shows the detected picture region and Fig. 8 shows picture removal of passport image and the connected component search result for character part extraction.

5 Passport Data Extraction

5.1 MRZ Data Extraction

The MRZ data can be obtained by the lower 2 lines of the passport images as shown in Fig. 8. The ICAO provides the rules of the generation of MRZ data [1]. If we generate the automata for MRZ analysis, we can obtain the MRZ data without any recognition because the field separator is defined by the alphabet <. The process of automata analysis is as follows:

1. Starting character of fist line is P (fixed size).
2. Second character is passport type (fixed size).
3. Next three characters define the issuing country (fixed size).
4. Variable name fields composed of first name and name divided by the separator. If the name field can not fulfill the whole field of first line, the other field is filled by separators.
5. The second line starts with the passport number field with variable size. This field 6. is finished by a check digit field.
6. Next is a nationality field with three characters (fixed size).
7. Six digit of date of birth field (YYMMDD). This field is finished by a check digit field.

8. One character for gender (M or F, fixed size).
9. Six characters of date of expiry (YYMMDD). This field is finished by a check digit field.
10. Variable personal information fields. If this field can not fulfill the whole field of second line, the other field is filled by separators except last two fields of check digits.

Passport automata for MRZ are shown in Fig. 9 and Fig. 10 illustrates the screen of extracted MRZ data.

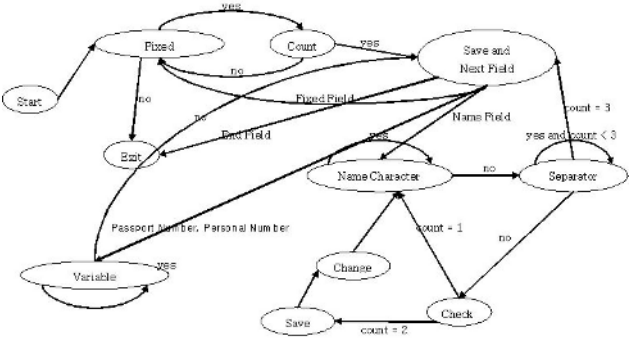


Fig. 9. Passport Automata

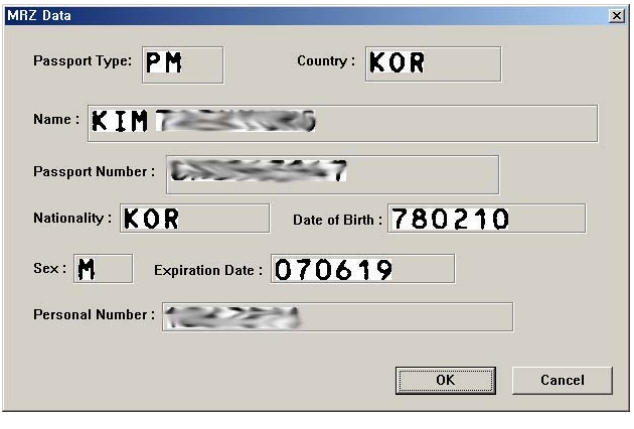


Fig. 10. Extracted MRZ Data

5.2 Information Extraction of Upper Part of Passport

The connected components of Fig. 8 contain the character information. If the character part is successfully detected, we can extract the information from the MRZ and detected character part. This character part has essentially same information of MRZ. Most of information is identical and some part is different form but contains same information such as month information. In reality, upper character part contains more information than MRZ. For example, month information in MRZ is two digit

numbers. The upper character part is represented by three characters. The table of comparison of each field is presented in Table 1.

Table 1. Comparison of Passport Fields

	Identical Field	Non-Identical Field
Passport Type	√	
Country	√	
Name	√	
Passport Number	√	
Nationality		√
Date of Birth		√
Sex	√	
Expiration Date		√
Personal Number	√	

The field of nationality has different number of characters. Thus, these fields are treated carefully. Fig. 11 shows the extraction of candidates characters for crosscheck.

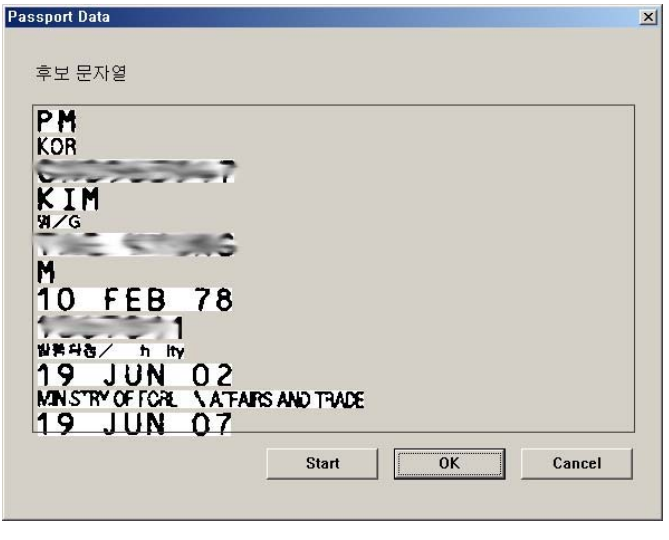


Fig. 11. Extraction of Character Part of Passport

6 Data Extraction Results

Character extraction experiment is performed with ten different passport images. Table 2 shows the extraction results of 10 different passport images of different countries. The database which we used seems small but the gathering of real passport

image is not easy way. Because of privacy problem, we must obtain the permission of use. The picture area detection and character part extraction results are shown in Table 2. The MRZ data extraction shows good results. But the extraction of character part shows moderate result. As shown in Fig. 11, the extracted characters are not either same size or same font. Thus, the character recognition for the extracted data is required for the precise comparison.

Table 2. Extraction Results of Passport Information

Picture Area Detection	Extraction of the characters	MRZ Data Extraction	Crosscheck/Field
10/10 (100%)	1370/1420 (96%)	852/880 (96%)	68/90 (75%)

7 Conclusions

This paper proposes a method to extract characters and picture in passport to implement passport recognition system. Most pre-process for passport recognition used way to extract MRZ code and picture region. However, this paper is using double extraction method for all character of passport as well as MRZ code. If passport recognition system is implemented by applying this method, precise recognition can be improved comparing MRZ code and data in passport. And extraction of picture region can be used to data for face recognition in hereafter. The research that extracts character in foreign passport as well as domestic passport is needed to implement automatic passport recognition system to after this subject. The extraction of the picture area shows 100% of correct separation ratio and the extraction of the characters for the recognition shows 96% of extraction ratio away ten different passports. Also, the method that can automatically compare passport data with MRZ code is also implemented for the crosscheck purpose. The crosscheck method shows moderate rate of comparison with MRZ and extracted character part. Thus, character recognition must be applied. After recognition, we may re-apply the comparison of crosscheck concept in order to verify the fraud of passport. For further study, a priori knowledge of the passport such as location information and the colour may use to improve the recognition.

References

1. ICAO, Document 9303
2. Jae-Uk Ryu, Tae-Kyoung Kim and Kwang-Baek Kim, “The Passport Recognition by Using Enhanced RBF Neural Network”, Proceedings of the Korea Intelligent Information System Society Conference 2002, Vol. 11, pp.529-534, 2002
3. Jae Chan Namkung, Howang Bin Ryou and Yun Namkung, “A Study on the Korean Character Segmentation and Picture Extraction from a Document”, Journal of the Korean Institute of Telematics and Electronics, Vol. 25, No. 9, pp.1091-1101, 1988
4. In Dong Lee, Oh Seok Kwon, Tae Kyun Kim : “A Method to Extract Characters and Non-characters Separate from Document Image, Journal of the Korea Information Science Society, Vol.17, No.3, May, pp.247-258, 1990.

5. F. M. Wahl, K. Y. Wong, and R. G. Casey, “Block Segmentation and Text Extraction in Mixed Text/Image Document”, Computer Vision Graphics and Image Processing, Vol. 22, pp. 375-390, 1982.
6. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Addison-Wesley, 1992.
7. Dacheng Wang and Sargur N. Srihari, “Classification of Newspaper Image Blocks Using Texture Analysis”, Computer Vision Graphics and Image Processing, Vol . 47, pp.327-352, 1989.

String Extraction Based on Statistical Analysis Method in Color Space

Yan Heping, Zhiyan Wang, and Sen Guo

School of Computer Science & Engineering, South China University of Technology, Guangzhou 510640, China
gzyhp@126.com, wzhyan@ieee.org

Abstract. A method based on statistical characteristics and color space consistent with human visual perception for pixels classification is brought forward in this paper. In the airline coupon color design, we use colors to distinguish different object, the idea is embodied in this method. The marked characteristics suitable for object pixels classification have been found by analysis the statistic characteristics of all sorts of pixels. The experiments have proved that this method is simpler, more efficacious and can support data analysis for the whole coupon project.

1 Introduction

Our airline coupon project group [1] developed a recognition and management system. It has processed millions of coupons for years. Its mean recognition rate is about 95%, and we wish to improve it. So we have studied on how to convert image character into graphic one to recognize it. In this research we found it is very important to extract pixels of character perfectly.

Researches [2] regard pixels extraction as classification of pixels in the coupon image; pixels on (belong to) characters which are to be recognized are classified to the foreground pixels (character objects, see Fig. 1(a)) and the others are classified to the background. In this research, the HSV space is used to extract pixel features and a neural network (NN) method which is based on the Principal Components Analysis (PCA) is used as a pixel classifier to classify all pixels into some foreordain sorts. Experiment result shows a good effect is reached.

1.1 Problems

But, there are still some problems in the segmentation result images, which include (1) some background pixels which have distinct visual perception with object pixels, see Fig.1(a)., are classified into object pixels set by mistake. (2) in some cases, some unknown color sorts are appeared in the coupon image, such as manual characters, see Fig.1(b).; in this instance, the classifier will not work normally; it will classify these color pixels into an adjacent sort; obvious, it is not reasonable.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 173 – 181, 2006.
© Springer-Verlag Berlin Heidelberg 2006

1.2 Reasons

For pixel classification task, color features are used to simulate the classification process of human visual perception. So, if the color space is more uniform and more consistent with human visual perception, the classification effect is better.

(a) (b)

Fig. 1. (a) The obvious error in pixels classification; (b) Incorrect pixels classification for unknown color class. In the upper: object pixels; In the lower: the original image; Within the ellipse: incorrect object pixels.

By data analysis in the HSV color space, one reason for the problem (1) is about uniformity and consistency of HSV space. There are many existing system for arranging and describing color, such as RGB, YUV, HSV, LUV, CIEXYZ, CIELAB, Munsell system, etc [3-4]. But, most of them are usually different from human perception. Among all the existing color systems, the Munsell color system is the best in simulation human color vision [4]. So, we select the Munsell color system.

The data analysis results show another reason for problem (1): there are over-learning or lack-learning in the NN training process. The color number in a color space is tremendous, and just some sorts of color pixels are selected to train the NN; so it is inevitable to classify a color into a color sort, which maybe farther from another sort in human visual perception.

To problem (2), it is a certain problem of this NN method. In NN design, the number of color sorts is foreordained. When an unknown color sort appeared, the classifier works abnormally is reasonable.

Additional problems are still existed. For example, the selection of suitable samples is difficult, and the classifier is complex in practical work.

1.3 Ideas for Solving Problems

Firstly, we should solve the color space problem. As said above, the Munsell color system is a better selection. Analysis from the design of coupon, we can see different objects are in different colors. The unchangeable colors are including 5 sorts as analyzed in [2]. And the casual unknown color sorts, most in the handwriting are also in a visual different color. So we can say the design of coupon is using the striking contrast colors to distinguish the different sorts of objects in coupon.

In classifier design, because of the limitation of NN and the demand of data analysis, we consider to reduce complexity and improve generalization ability. By analysis the background of this problem, it is a statistical problem in essential. Considering an

ideal instance, the unchangeable background only includes some single colors and the objects are in the colors. So it is a simple task to classify pixels according to the color features in any color space. As a matter of fact, in the process of coupon going to be pressed and printed, the disturbing such as the ink infiltration, outspread, especially color superimposition, these single colors will be changed; they will form a complex color distribution in any color space, see Fig.2.

But, the changing of these single colors should obey some statistical rules. So to grasp the color distribution of all kinds of pixels is the essential to design an effective classifier.

In this research, we analyzed the statistical characteristics of all sorts of pixels in the HVC color space. Then, based on these analysis results, we designed a simple classifier to extract the object strings. It is effectively proved by practical work.

This paper is organized into 5 sections. In section 2, we introduce the color space and color distance used in our research. The 3rd section analysis the statistical characteristics of the fixed color sorts, mainly in the object sort, sort 1; then based on these analysis results, we propose a method to separate the object pixels from others. In section 4, we give an experimental results compared to method in [2]. At the last section, a conclusion is given.

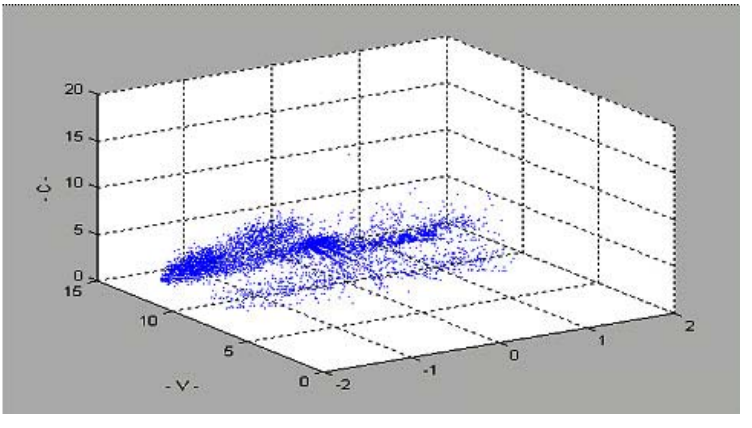


Fig. 2. Color distribution in the HVC color space

2 Color Space Selection

Although there are so many color-order systems, most of them are inconvenient to apply in segmentation because the color expressed is usually different from human perception. Among all the existing color systems, the Munsell color system is the closest to the human color vision [4].

The Munsell System describes all possible colors in terms of its three coordinates, Munsell Hue, Munsell Value, and Munsell Chroma [4]. A color in the Munsell color system can be written as HV/C. But, it is impossible to calculate color difference of two colors by such representation. We have to convert such representation into real

numbers. For this reason, we use the hue, value, chroma space(HVC color space) instead of the Munsell colors in terms of tri-attributes of human color perception[5]. In essential, the HVC color space is same as the Munsell color system, so the HVC color space also gives the best performance for experiments with variety of color spaces [6].

Before using the HVC color system to deal with images, we have to transform images from the RGB space to the HVC space. There are many ways to transform the images between the RGB space and HVC space. Here we use the improved mathematical transform of RGB coordinates to the HVC color system described in Gen et al[7]. Suppose *r*, *g*, *b* represent the three components red, green and blue in RGB color space. See [7] for the color transformation of RGB to HVC.

To calculate the color difference in the HVC color space, we make use of National Bureau of Standards (NBS) color distance [8] instead of the Euclidean distance measure.

It is found that in the HVC color space, the human color perception has close relation to the NBS color distance. The relation of human color perception and NBS distance is shown in Tab.1. From the table, we know that if the NBS color distance of two colors below 3.0, human beings will regard the colors almost the same.

Table 1. The correspondence between the human color perception and the NBS distance

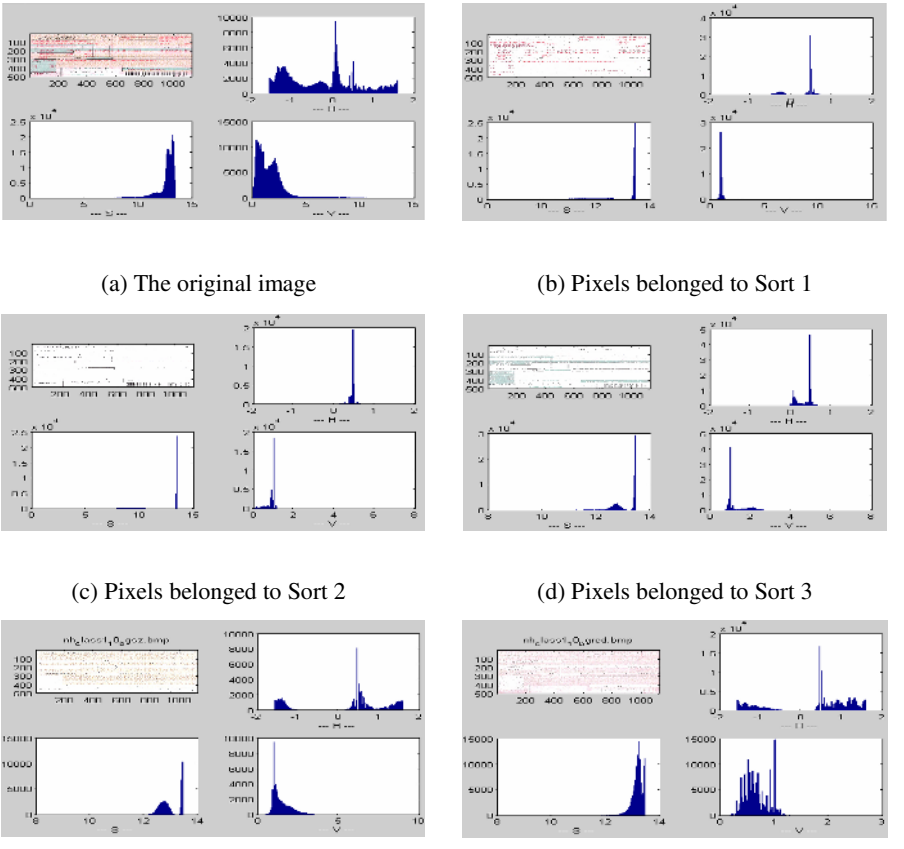
NBS Value			Human Perception
0	-	1.5	Almost the same
1.5	-	3.0	Slightly different
3.0	-	6.0	Remarkably different
6.0	-	12.0	Very different
12.0	-		Different color

3 Statistical Analysis for Pixels Color features

The original image to segment in our application is shown in Fig.3(a).

By analysis the colors appeared in the image, we classify most of the pixels into 5 fixed classes: red(object characters), black (background characters and form lines), green (background), yellow (background characters), and low red (background). There may be some uncertain color pixels in it, such as other smear spot or manual handwriting in it. Our object is to extract pixels in sort 1 and eliminate all other pixels.

In order to analysis the statistical characteristics of all kinds of pixels, we classified all these pixels in the image to the five sorts manually, as shown in Fig.3(b). to Fig.3(f).. After the transformation of the pixel values from RGB to HVC, the histograms of three coordinates of all sorts are showed in the corresponding parts in Fig.3. In the H parts, as shown in Fig.3, different sorts are located in different sections obviously. But in the V and C parts, this characteristic is not so obvious. This result is consistency with the design idea of airline coupon as we analyzed before, which use



(a) The original image

(b) Pixels belonged to Sort 1

(c) Pixels belonged to Sort 2

(d) Pixels belonged to Sort 3

(e) Pixels belonged to Sort 4

(f) Pixels belonged to Sort 5

Fig. 3. All sorts of pixels and histograms of their tri-attributes in a typical image

striking contrast colors to distinguish the different objects. So, the colors are mainly represented by hues.

Observing the H value distribution, we found it include two parts separated by 0. We classify the pixels into two parts by this threshold. By zooming in the image which just include two kinds of pixels, and observing carefully we can find that pixels in the second part whose H value is large than 0 is not the red object pixels indeed. The two kinds of pixels are in a much color difference. It means that the other sort pixels are classified into the object sort by mistake. By calculating means of the two groups of pixels, and calculating their NBS distance, the result shows that they are in a biggish difference, in slightly different level. So we are sure that the H value of sort 1 is in the section less then 0. We can also use this method to analysis data; it shows the advantage of our method. Further analysis in all coordinates reveals that the histogram of H value has obvious characteristics: the cohesion within sort 1 is strong and the distances between sorts are far. For example, we can separate sort 1 from sort 2 and 3 directly. But this characteristic is not so obvious for the V value and C value.

3.1 The Statistical Characteristic Analysis

Based on the analysis above, if we can find the probability distribution of the H-attribute of all sorts of pixels, then we can separate them by using the Bayesian Theorem.

Our research is based on the same sorts of airline coupons, which are chosen by visual quality. In this statistical analysis, our samples are came from the 5 sorts of color pixels, a sort of color pixels in an image form a sample, and there are 5 sorts of samples. We set the confidence in 0.05.

By hypothesis testing, we get results as follows:

1. Samples in sort *i(i=1,...,5)* are in the same distribution; this is the foundation to separate them with others.

2. The distribution samples in sort 1 do not obey the normal distribution.

According to [9], the color distribution in printing paper maybe obeys the normal distribution or Passion distribution. But in our research we find it is not suitable to our case. Seeing from the fitted curve, we can see that the curve is too steep and the distribution is too concentrated in the mean point. The deflection and steepness is not conformed to the characteristics with normal distribution. See Fig.4.

3. The distribution of samples in sort 1 is similar with the log normal distribution.

By the analysis and observation above, we think that the distribution of samples in sort 1 should obey the log normal distribution. But the K-S testing and the rank testing denied this hypothesis. So, we think it is not a standard distribution, just similar with the log normal distribution.

4. The samples in sort 1 can be separated from the other sorts by a threshold.

3.2 The Extraction of Pixels in Sort 1

The distributions of samples in sort 1 and sort 2, sort 3 is not overlapped. We can separate them directly. But the distribution of samples in sort 1 and sort 4, sort 5 have a little overlap, see Fig.5.

Seeing from the overlapped curves, we can separate samples in sort 1 from sort 4 and sort 5 by select a threshold. In practical, we select the intersection of the red curve

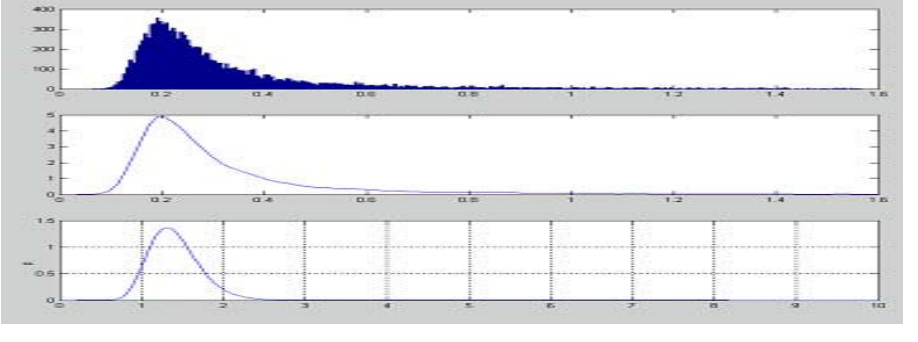


Fig. 4. Comparison of the estimative density distribution curve. Note: curves are symmetric flipped; Upper: the histogram of sort 1 pixels; Middle: the estimative density distribution curve of sort 1; Lower: the log normal distribution curve.

and the green curve as the theshold (T) to extract pixels in sort 1. The experiential value is -0.63 in our experiment, and the misclassification rate is under 0.05.

So, this method includes two parts, the statistical analysis part and classification part, which are describe as following:

Part 1: statistical analysis

1. Select a sort of airline coupon images, in which the pixels in sort 1 are in the same distribution; these images are come from a batch of coupon tickets.
2. Collect pixels in sort 1, 4 and 5 by manual visual observation.
3. Use the three sorts of samples to estimate their probability distribution curves, then determinate the threshold (T) by their intersection.

Part 2: Classification

1. Input a coupon image.
2. Calculate the H value (H_{ij}) in HVC space for each pixel P_{ij}.
3. Classify P_{ij} into foreground if $P_{ij}>T$; otherwise, classify P_{ij} into background.
4. Output the foreground image.

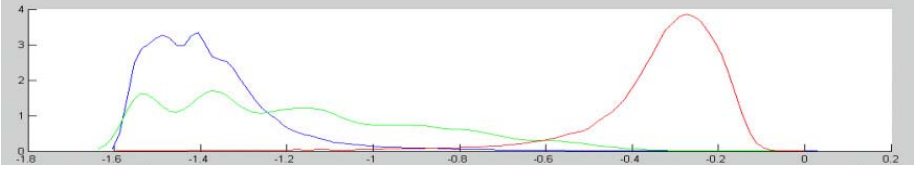


Fig. 5. The estimative density distribution curve of samples in sort 1(red), sort 4(blue) and sort 5(green)

3.3 Performance

In practical operation, different from the method in [2], which needs a complex training process, we just need to classify all pixels into two sorts: sort 1 pixels and the other. So it has the time complexity of $O(n)$, n is all the pixels in the original image, and need no extra space. It is a simple and effective method to extract strings.

4 Experiment and Results

We use our method to extract the object strings in the airline coupon project, which separate an image into two images of the same size, one is the object level and the other is the background level. We only concern on the object image.

By the subjective evaluation, we think this classification method is effective; the string objects image is cleaner compared to the result image of method in [2], see Fig.6.To prove its performance quantitatively, we use our evaluation system[10] to evaluate it., see formula 1.

$$u_f = C_f / |\mathrm{Fs}|;\ u_b = C_b / |\mathrm{Bs}|;\ u_t = \left[C_f + C_b\right] / |\mathrm{T}| \tag{1}$$

Where, c_b represent the counts of pixels which should be classified to background pixel set but have been classified to object pixel set, c_f represent the counts of pixels which should be classified to object pixels set but have been classified to background pixels set, F_s and B_s represent the object pixel set and the background pixel set of the standard segmentation image and $T = F_s \cup B_s$; u_f is the object pixel misclassified rate, u_b is the background pixel misclassified rate, and u_t is the total misclassified rate. The two indicators u_f and u_b are mainly used to analysis the algorithm and data in detail. The indicator u_t is mainly used to evaluate the integrated performance of the algorithm. The results are shown in Tab.2.

Table 2. Results comparing to method in [1]

	μ_b	μ_f	μ_t
Our method	0.0441	0.0526	0.0456
Method in [1]	0.0931	0.0486	0.0908

Seeing from the results, we can find that the method in [2] is a little better in the indicator of u_f. This is because of that its training samples are retained as more sort 1 pixel as possible. But, our method is much better in u_b and u_t. By all counts, it is a simple and effective method to extract strings from the airline coupon images.

Fig. 6. Subject effect comparing to method in [2]; Upper: the result images of method in [2]; Middle: the original images; Lower: the result images of our method

5 Conclusion

In this paper, we propose a simple method to extract strings from the airline coupon. It is based on the Munsell color system and on the statistical analysis. It is effective proved by practical work in the airline coupon project.

In the further research, we should study the method to determinate the threshold automatically and dynamically, for the H-attribute of the object pixels maybe drifted when the sorts of coupons increased.

References

1. S. Zhao, et al, "A High Accuracy Rate Commercial Flight Coupon Recognition System", Proc. of 7th International Conf. on Document Analysis and Recognition, 2003, Edinburgh, pp. 82-86.
2. Y. Li, et al, "String Extraction in Complex Coupon Environment Using Statistical Approach", Proc of 7th International Conf. on Document Analysis and Recognition, 2003, Edinburgh, pp. 289-294.
3. X. Wand and C.C. Kuo, "A new approach to image retrieval with hierarchical color clustering", IEEE Trans. on CSVT, vol.8, no.5, Sep., 1998.
4. F.W. Billmeyer and M. Saltzman, "Principles of Color Technology", 2nd Ed., New York, Wiley, 1981.
5. S.C. Pei and C.M. Cheng, "Extracting color features and dynamic matching for image database retrieval", IEEE Trans. on CSVT, vol.9, no.3, pp.501-512, Apr., 1999.
6. J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions", IEEE Trans. on PAMI, vol.17, no.7, pp. 729-736, July 1995.
7. M. Bartkowialk and M. Domanski, "Vector median filters for processing of color images in various color spaces", Proc. IEE Conference on Image Processing and Its Applications, 1995, pp. 4-6.
8. Y. Gong, G. Proietti, and C. Faloutsos, "Image indexing and retrieval based on human perception color clustering", Proc. IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998, pp. 578-583.
9. X. Lu, Color Science in Encapsulation, ZhenZhou University Press, 2002.

Interactive System for Origami Creation

Takashi Terashima, Hiroshi Shimanuki, Jien Kato, and Toyohide Watanabe

Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
{takashi, simanuki}@watanabe.ss.is.nagoya-u.ac.jp,
{jien, watanabe}@is.nagoya-u.ac.jp

Abstract. This paper proposes a new system which supports origami creators who have no special knowledge about origami creation to create their unique works easily in 3-D virtual space. Moreover, 2-D diagrams or 3-D animation are automatically made for describing the folding processes so that people can re-build these works. Users can decide folding operations and create works by an interactive interface. For easy creation, two methods are proposed. One is a method for representing overlapping-faces of 3-D virtual origami in order to support users' recognition of origami's conformation. As a result, users can input information about folding operations easily and correctly. The other one is a method for deriving halfway folding processes according to users' intents. Even if users have rough images about shapes of origami works, they may not be able to start creating an origami model as their imagination. Namely, the system shows folding processes from square to basic forms until they can start do it by themselves. We expect that the common people will create and publish their unique works and more people will enjoy origami.

Keywords: Origami, Interactive Interface, Computer Graphics, 3-D Virtual Model, Origami Base.

1 Introduction

Origami, one of the Japanese traditional cultures, is perceived worldwide as the art of paper folding which has abundant potential. Making origami assists not only the enhancement of concentration and creativity but also rehabilitation exercise, antiaging effects, and so on. Traditionally, people play origami based on drill books (text books) or materials on web pages [1] in which the folding processes consist of simple folding operations are illustrated by diagrams. Recently, a system which recognizes folding operations from origami drill books and displays 3-D animation of folding processes were proposed [2] [3]. On the other hand, these drill books or materials are made and exhibited by limited persons who have special knowledge about origami creation. It is difficult for the people who have no special knowledge about origami creation to create their unique works and to describe the folding processes by diagrams so that people can re-build them (i.e. to publish works). The main reason of this is botheration of using tangible papers thorough trial and error processes. Another reason is trouble of making drill books or other instructional materials. For these reasons, few people create new

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 182–194, 2006.
© Springer-Verlag Berlin Heidelberg 2006

origami works and it is not often that innovative works are made in public. Therefore, an environment that facilitates creative activities is required.

This paper proposes an Interactive System for Origami Creation. This system supports origami creators including the people who have no special knowledge about origami creation. Users can transform virtual origami by operating this system interactively. Using the system, they are able to create their unique works easily and comfortably. Moreover, 2-D diagrams or 3-D animation which describe the folding processes can be automatically made for publishing. We expect that the common people will create and publish their unique works and more people will enjoy origami. As related work, a system that represents dialogical operations of origami in 3-D space has been introduced [4]. However, origami creation is not considered by using the system.

In order to let users input their intended operations without any mistakes or difficulty, we consider a user interface and some functions which ease users' operations and help their recognition about shape of 3-D virtual origami. Hereafter, we first show the framework and user interface of this system in section 2. Then, two methods for improving the usability of our system are proposed in section 3 and section 4. One of the methods is for representing virtual origami. The other is for deriving halfway folding processes. Finally, we show the conclusions and future prospects in section 5.

2 Framework and Interface

In this section, we show the framework of our proposed system and outline the system. Then, we show user interface and consider how to improve the usability of the system.

2.1 Framework

Figure 1 and Figure 2 show the basic framework of the system and interaction between the system and a user, respectively. The system first receives positional information of a fold line determined by user's input. Then, the feasible folding operations are constructed based on the crease information, which is superficial and incomplete. They are obtained by maintaining consistency of crease patterns under some geometrical constrains [5]. All the feasible candidates are simulated against an internal model of origami. As a consequence, several different origami states are obtained from each candidate. Subsequently, the system presents resultant models corresponding to those candidate operations. Finally, the user selects his/hers desired operation. In this way, this interaction enables users to input folding operations easily. Namely, users can transform virtual origami variously by the basic mouse action. By the repetition of the interaction, the system can understand a sequence of folding operations required to create an origami work, and represent them in the form of 3-D animation or a sequence of 2-D diagrams.

2.2 User interface

Figure 3 shows user interface of proposed system. A state of origami at some step is displayed on the left of the window, while the states which are simulated according to candidate operations (see the previous section) are displayed on the right of the window.

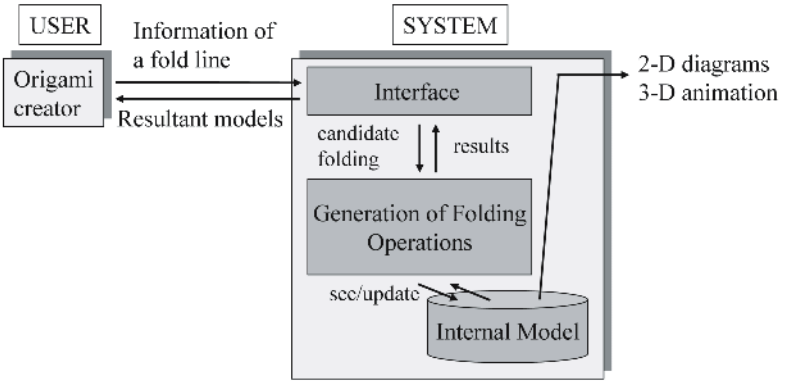
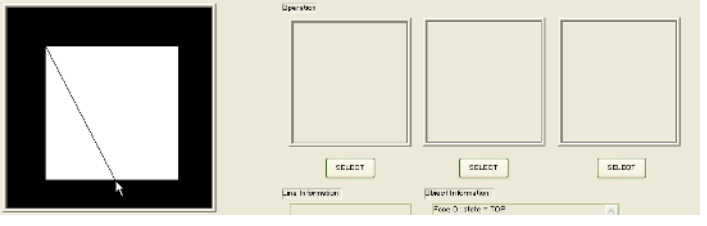
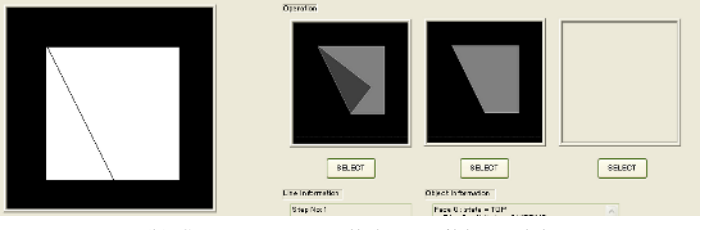


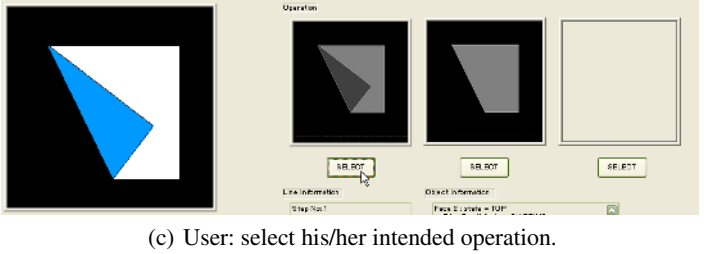
Fig. 1. Framework of proposed system



(a) User: input a fold line.



(b) System: present all the possible models.



(c) User: select his/her intended operation.

Fig. 2. Interaction for folding operation decision

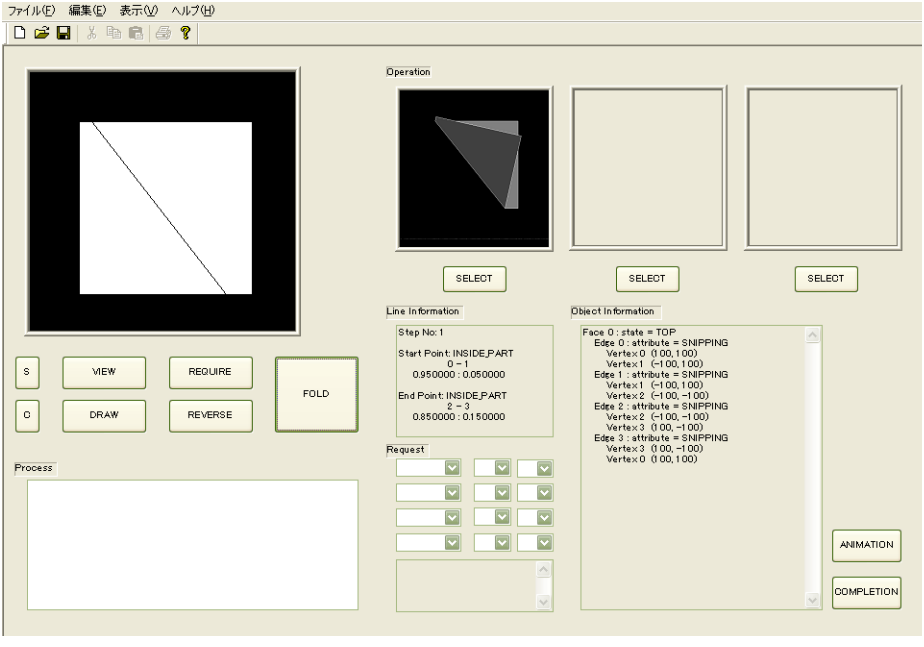


Fig. 3. User interface

The left graphic has two modes, view mode and draw mode. In view mode, users can see the origami model from all viewpoints and can not input anything. On the other hand, in draw mode, users can draw a fold line from fixed viewpoint. By the idea having two modes, users can understand the shapes of origami model and can draw a fold line correctly.

Generally, there are several considerations to improve the usability of the system. In order to design an ideal user interface for easy-to-use system, we discuss three elements: intended users, cognitive load, and operational error.

Intended Users. People often feel that to create origami works with real paper is too much trouble, for example, paper crumples up through a trial and error process. Moreover, it is difficult to remember the folding processes for the created works, and also difficult to describe the folding processes in a sequence of diagrams for publication. From these backgrounds, the aim of intended users of the system is to create and to publish their unique origami works comfortably and easily by using the system. Additionally, we assume that intended users do not have special knowledge about origami creation (such as design knowledge based on a crease pattern) and they have rough images about shapes of origami works (such as a four-legged mammal).

Cognitive Load. There are various kinds of folding operations. Since users have to give the desired folding operations correctly, an environment which enables users to understand the configuration of origami intuitively and to input folding operations by simple actions is required.

As mentioned previously, our proposed system enables users to input various kinds of folding operations through the interaction between the system and users. At this time, users' required action is only to input folding operations through the basic mouse action. Furthermore, users can see an origami model from all viewpoints in view mode.

Operational Error. There is a possibility that users draw a fold line at the wrong (undesired) position. We should consider preventive measures and countermeasures against this operational error.

As a preventive measure, an environment which enables users to understand the configuration of origami intuitively (mentioned in section 2.2) is required. Furthermore, as a countermeasure, the system has an undo/redo function which allows users to undo their inputs from any step in case of operational error.

2.3 Required Methods

From these discussions, we must propose following two methods. One is a method for representing 3-D virtual origami, discussed in section 2.2 and 2.2. Generally, an origami model is constructed by planar polygons corresponding to faces of origami. Therefore, when an origami model is displayed, multiple faces on the same plane (called overlapping-faces) probably seem to be one face. This incorrect perception occasionally obstructs users' inputs. Consequently, we must propose a method for representing overlapping-faces of 3-D virtual origami in order to support users' recognition of origami's conformation in both view mode and draw mode. As a result, users can input information about folding operations easily and correctly.

The other one is a method for deriving halfway folding processes. Even if users have rough images about shapes of origami works as mentioned in section 2.2, they may not be able to start creating an origami model as their imagination or may not be able to continue at a step, especially when they do not have special knowledge about origami creation. In order to deal with such case, we must propose a method for deriving halfway folding processes according to users' intents. Namely, the system shows folding processes to users until they can start do it by themselves.

In this paper, we describe these methods in detail. The former method is proposed in section 3, while the latter method is proposed in section 4.

3 Method for Representing Origami

This section specifically describes our method for representing overlapping-faces of 3-D virtual origami for the user interface.

3.1 Our Approach

In order to represent virtual origami 3-dimensionally, we consider the extended (i.e. ideal) representation as the reconfiguration of a 3-D origami model. Specifically, overlapping-faces are moved apart slightly by rotating polygons along a rotation axis determined from figurations and relationships of faces. Because of the reconfiguration in 3-D space before 3-D rendering, this method has the advantage that an origami model

can be seen from all viewpoints without any renewed reconfigurations if once it is reconfigured. Namely, the reconfiguration depends not on users' viewpoints, but on the origami model.

The elementary transformation is a movement (i.e. rotation) targeted at two faces which are adjoining each other. Order and portions of movement are based on figurations and relationships of overlapping-faces. We discuss which faces should be moved, which portions of the faces should be rotated, and what order they should be rotated in.

3.2 Free-Portion

We define a "free-portion" (part of a face) as the portion that is not restrained by the adjoining face and can move freely. Such free-portions should be moved (i.e. rotated). In order to find out a free-portion, firstly, we define a "free-edge" as follows.

Free-Edge. Given two faces (F_1 and F_2) that are on the same plane and are adjoining each other, an edge E of F_2 is a "free-edge" to F_1 if following conditions are all satisfied:

- E is not an edge of F_1, but an edge of F_2.
- E and F_2 are not covered by other faces.

In order to determine whether the latter condition is satisfied, cross-sections of origami are generated by cutting origami perpendicular to E. Figure 4 gives examples of free-edge and not-free-edge. At State C, both sides of $F2$ are covered by other faces, and these faces are joined on the same side of E. Namely, E and F_2 are not covered by other faces, and E is a not-free-edge. This definition is used to determine free-portion as follows.

Free-Portion. A free-portion of a face F_2 to the adjoining face $F1$ is determined by following steps.

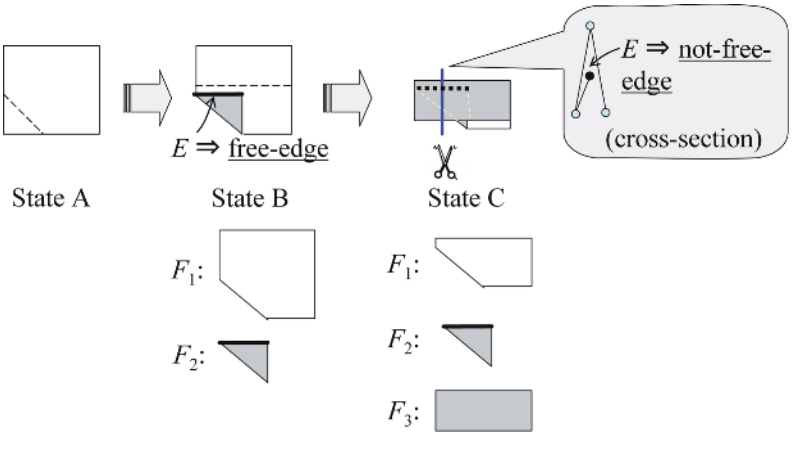


Fig. 4. Examples of free-edge and not-free-edge

1. Determine whether each edge of F_2 is a free-edge to F_1.
2. Draw a line L that connects two points between free-edge and not-free-edge.
3. Define the polygonal area enclosed by the free-edges and L as a free-portion.

This line L becomes the rotation axis when the free-portion is rotated. Figure 5 show examples of determining free-portion. In the case of F_1, the free-portion is the triangular shape (i.e. half of F_1). On the other hand, in the case of F_2, the free-portion is the whole face F_2. More specifically, F_2 is unrestrained in moving by F_3. In addition to these examples, there are cases where no free-portions of some faces exist since polygonal area can not be formed in step 3.

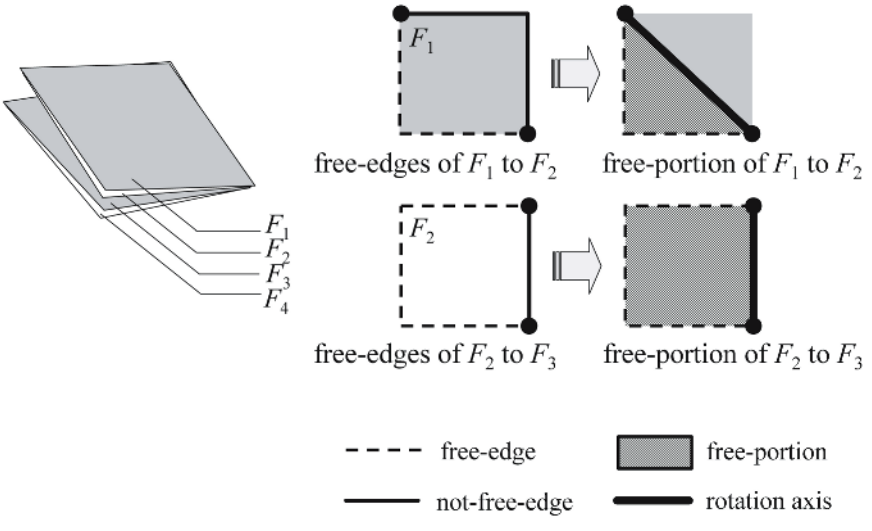


Fig. 5. Examples of determining free-portion

3.3 Grouping of Faces

In Figure 5, the free-portion of F_3 to F_4 is the triangular shape like that of F_1 to F_2. If the free-portion of F_3 is rotated before the rotation of F_2 (whole area is the free-portion), the free-portion of F_3 collides against F_2 and the reference plane of the rotation of F_2 gets fuzzy.

To solve this problem, we propose a method that groups overlapping-faces based on dependency relation about their movements. Namely, if a face can move independently of another face, the two faces are classified into different groups. Otherwise, they are classified into the same group. This grouping of faces determines the order of face's movements. The procedure for grouping overlapping-faces is described as follows.

Procedure for Grouping.

1. Make the order list of overlapping-faces.
2. Determine free-portion of each face to the adjoining face behind it (beginning at the bottom).

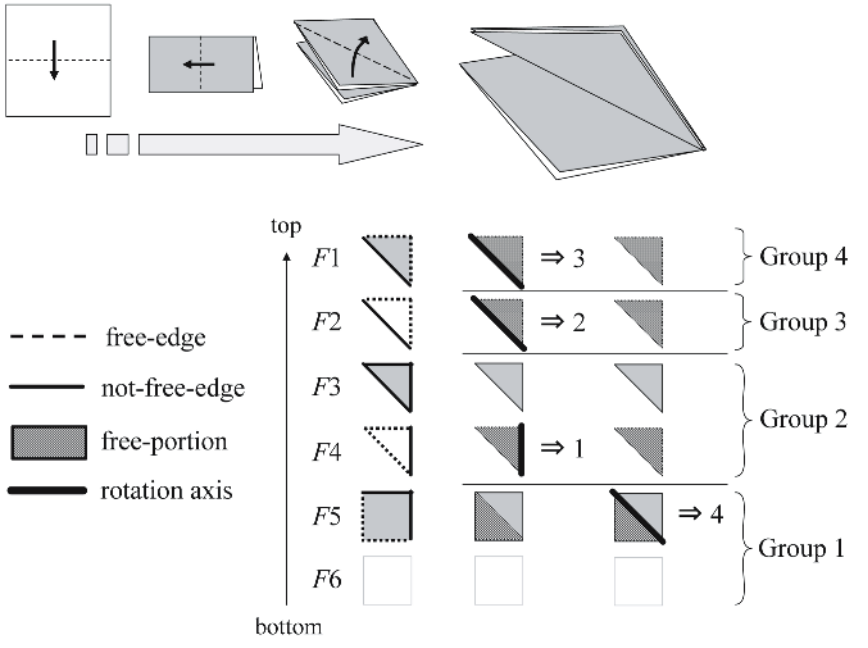


Fig. 6. Examples of grouping faces

3. Let the faces that whole area is the free-portion be chief faces of their groups. Let the undermost face also be chief face.
4. Crassify each not-chief face into the group the nearest behind chief face belongs to.

This grouping solves the problem described above. More specifically, no faces collide against other faces by moving all faces which belong to the same group before the rotation of each free-portion. Each rotation angle can be decided in consideration of angular difference between anteroposterior groups.

3.4 Representation Algorithm

Our proposed method for representing 3-D virtual origami is summarized as follows.

Representation Algorithm.

1. Make the order list of overlapping-faces.
2. Determine free-portion of each face to the adjoining face behind it (beginning at the bottom).
3. Determine chief faces and classify other faces with appropriate groups.
4. Rotate set of faces in each group collectively along the chief's axis (i.e. chief face of the group and faces which belong to the group). Rotation angle is constant.
5. Rotate free-portions of overlapping-faces in sequence along respective axes.

Figure 6 shows example of representing 3-D origami based on this algorithm. In this case, four chief faces and four groups are formed. Subsequently, sets of faces in

group 2, group 3, and group 4 are rotated along their chiefs' axes. Finally, the free-portion of F_5, only not-chief face which can move, is rotated along own axis.

4 Method for Deriving Halfway Folding Processes

If users can not start or continue folding virtual origami, the system should show folding processes to users until they can do it. In this section, we propose a method for deriving halfway folding processes according to users' intents.

4.1 Our Approach

It is sure that users who have rough images about shapes of origami works have the most difficulty in folding virtual origami from square to some step. For example, when a user wants to create a four-legged mammal (such as a dog), can he/she specify the first operation of the folding process? Moreover, can he/she know how to fold to make six corners which will become four legs, a head, and a tail eventually? The answers to these questions are probably "No" if the user does not have special knowledge about origami creation.

Noting this, we propose a method for deriving folding processes from square to some step so that users can start creating origami. In the case of above example, the system should derive and show the folding process until six corners are composed. After that, in order to create a dog, the user will be able to fold origami to determine corners' positioning, balance, and so on.

4.2 Origami Base

We use the idea of an origami base [6, 7] in our deriving method. An origami base is a specific form at the intermediate stage of folding origami from square (initial state) to the specific work. The base has about the same number of corners as the corresponding work. Figure 7 shows an example of an origami base. Crane base has five corners corresponding to five parts of the work: crane's head, tail, body, and two wings. Furthermore, various works can be created from one common origami base. In Figure 7, works which have about five parts can be derived from crane base. There are about twenty origami bases, and most origami works are derived from one of them.

Each origami base has several long and short corners. Moreover, corners can be grouped based on their constructional symmetry. For example, in the case of crane base (see Figure 7), there are four long corners and one short corner. These four long corners are divided into two groups: the group of two upward corners (called group A) and that of two downward corners (called group B). As above, corners of an origami base have two attributes, length and symmetry.

4.3 Supporting Origami Creation Based on Origami Base

As mentioned previously, parts of origami works are closely associated with corners of origami base. Therefore, when users have intents about parts of origami works, the

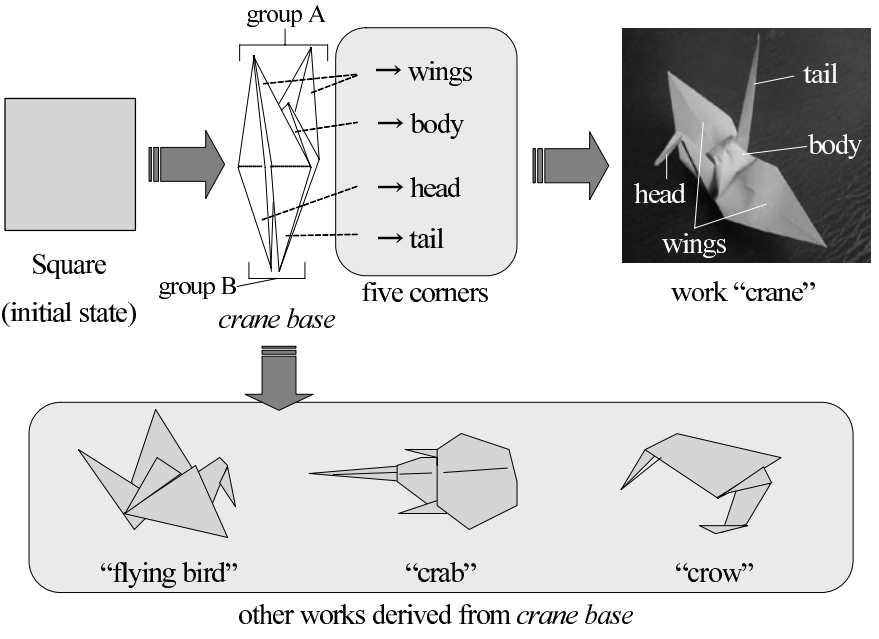


Fig. 7. Example of origami base

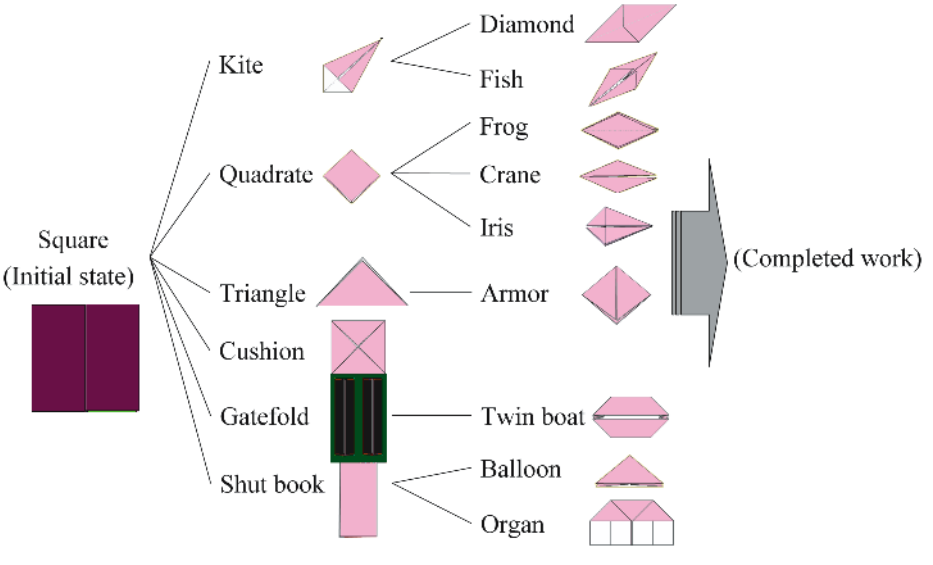


Fig. 8. Derivation graph of origami bases

system should select the origami base corresponding to works of users' intents. Our system teaches the folding process transforming an origami model from square to the origami base. We show how to select the origami base according to users' intents.

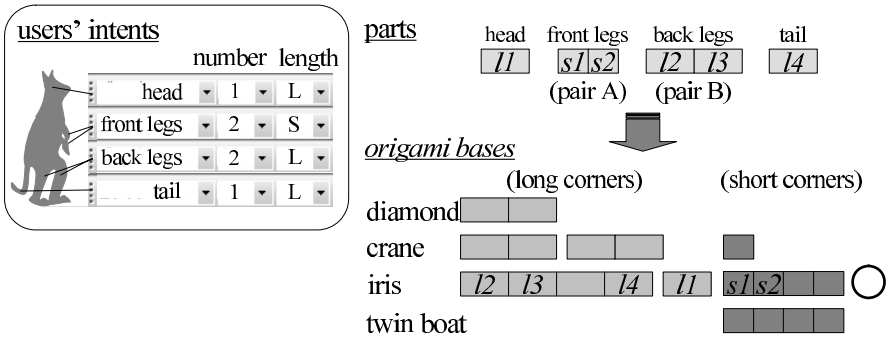


Fig. 9. Example of selecting the origami base

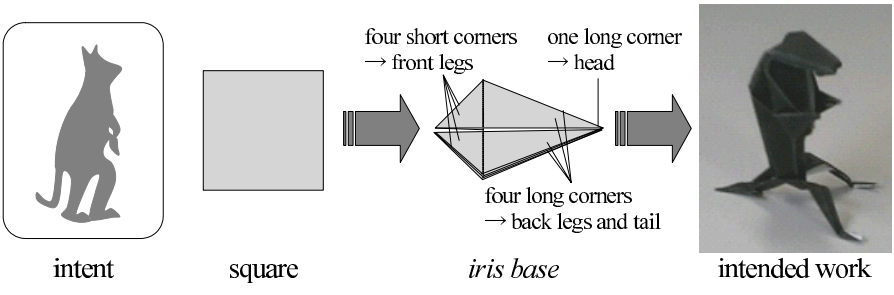


Fig. 10. Example of creating intended work from the base

In users' intended work, there may be a pair of the same kind of parts. The same kind of parts should be derived from the corners of the same group. For example, in the case of Figure 7, wings of the work are a pair. Therefore, it is undesirable that one wing is derived from the corner of group A and the opposite wing is derived from the corner of group B.

From this discuss, the rules for selecting the origami base according to users' intents are as follows.

Rules for Selecting the Origami Base. If a user wants to create a work which has m long parts and n short parts, the system selects the origami bases which satisfy the following conditions.

- Have more than m long corners and more than n short corners.
- Have enough groups which can correspond to each pair of the same kind of parts.

You can consider parts of works as objects and can consider corners of origami bases as containers. In this method, the containers which can accommodate all objects are selected.

Figure 9 shows an example of selecting the origami base according to users' intents. Now, the user wants to create a work of an animal which has two short legs (pair A),

two long legs (pair B), a head, and a short tail. For the sake of simplicity, we assume that there are four origami bases: diamond, crane, iris, and twin boat base. In this case, only iris base is selected, because it has more than four long corners and more than two short corners, and has the group of four long corners corresponding to pair B and the group of four short corners corresponding to pair A. Not having enough corners or groups that can correspond to parts or pairs of the work, other three bases are not selected.

Users can start creating origami works from the selecting origami base which has similar shape to their intended works. Namely, by deriving halfway folding processes, difficulties of users' creation from a square can be overcome. Figure 10 shows an example of creating intended work described above from iris base. The work similar to rough image can be actually created from iris base selected by the system. In this way, it is sure that intended works are easily created from origami base.

5 Conclusions

In this paper, we proposed the system which supports origami creators who have no special knowledge to create their unique works easily in 3-D virtual space. Moreover, the system automatically makes 2-D diagrams or 3-D animation for describing the folding processes so that people can re-build works. Users can decide folding operations and create works by an interactive interface. We discussed three elements about user interface: intended users, cognitive load, and operational error. Consequently, we proposed two methods: a method for representing virtual origami 3-dimensionally, and a method for deriving halfway folding processes by using origami base. By the former method, users can input information about folding operations easily and correctly. By the latter method, users can start creating origami works by themselves. These two methods overcome the difficulties of users' creation of origami works.

As our future work, we must consider advanced methods for deriving halfway folding processes.

Firstly, we should deal with users' complicated intents. For example, when users' intended works have many (more than ten) parts, all existing origami bases can not correspond to them. We consider that this problem is possible to be solved by combination of several origami bases. Actually, there is a basic form called dinosaur base which can be transformed into dinosaurs with lots of parts. Half of this form comes from crane base, and the other half comes from frog base. Namely, a basic form which has more corners may be produced by combining several origami bases. Therefore, we have to consider combination of origami bases.

Secondly, deriving halfway folding process after starting to create must be considered. This paper proposed a method for deriving folding processes from square to some step. However, users may want to vary or add their intents along the way. For this purpose, we consider that the system should recognize where present state are in the derivation graph. Moreover, the learning in the derivation graph will be required.

Finally, we should take into account the characteristics of origami base other than the number, the length, and symmetry of corners. For example, considering alignment of corners according to users' intents, the system will be able to provide more appropriate origami base for users. We must consider what characteristics are useful and how they are input by users.

References

1. Alex Barber. *"Origami"*. http://www.origami.com/index.html.
2. J. Kato, T. Watanabe, H. Hase, and T. Nakayama. "Understanding Illustrations of Origami Drill Books". *J. IPS Japan*, 41(6):1857–1873, 2000.
3. H. Shimanuki, J. Kato, and T. Watanabe. "Recognition of Folding Process from Origami Drill Books". In *Proc. of 7th International Conference on Document Analysis and Recognition*, pages 550–554, 2003.
4. S. Miyazaki, T. Yasuda, S. Yokoi, and J. Toriwaki. "An Origami Playing Simulator in the Virtual Space". *J. Visualization and Computer Animation*, 7(6):25–42, 1996.
5. H. Shimanuki, J. Kato, and T. Watanabe. "Constituting Feasible Folding Operations Using Incomplete Crease Information". In *Proc. of IAPR Workshop on Machine Vision Applications*, pages 68–71, 2002.
6. Patricia Gallo. *"ORIGAMI"*. http://www.netverk.com.ar/~halgall/origami1.htm.
7. Tomohiro Tachi. *"TT's Origami Page"*. http://www.tsg.ne.jp/TT/origami/.

Using Bags of Symbols for Automatic Indexing of Graphical Document Image Databases

Eugen Barbu, Pierre Héroux, Sébastien Adam, and Éric Trupin

LITIS, Université de Rouen,
F-76800 Saint-Etienne du Rouvray, France
Eugen.Barbu@univ-rouen.fr

Abstract. A database is only usefull if it is associated a set of procedures allowing to retrieve relevant elements for the users' needs. A lot of IR techniques have been developed for automatic indexing and retrieval in document databases. Most of these use indexes depending on the textual content of documents, and very few are able to handle graphical or image content without human annotation.

This paper describes an approach similar to the bag of words technique for automatic indexing of graphical document image databases and different ways to consequently query these databases. In an unsupervised manner, this approach proposes a set of automatically discovered symbols that can be combined with logical operators to build queries.

1 Introduction

A document image analysis (DIA) system transforms a document image into a description of the set of objects that constitutes the information on the document in a way that can be processed and interpreted by a computer [1]. Documents can be classified in mostly graphical or mostly textual documents [2]. The mostly textual documents also known as structured documents respect a certain layout and powerful relations exist between components. Examples of such documents are technical papers, simple text, newspapers, program, listing, forms... Mostly graphical documents do not have strong layout restrictions but usually relations exist between different document parts. Examples of this type of documents are maps, electronic schemas, architectural plans...

For both categories of documents, graph based representations can be used to describe the image content (e.g. region adjacency graph [3] for graphical and Voronoi-based neighbourhood graph [4] for textual document images).

This paper presents an approach similar with the "bag of words" method from Information Retrieval (IR) field applied to graphical document images. A ocument representation is built based on a bag of symbols found automatically using graph mining [5] techniques. In other words, we consider as "symbols", the frequent subgraphs of a graph-based document representation and we investigate if the description of a document as a bag of "symbols" can be profitably used in an indexing and retrieval task.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 195–205, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The approach has the ability to process document images without knowledge of models for document content. Frequent items are used in clustering of textual documents [6], or in describing XML documents [7], but we do not know any similar approach in the DIA field.

In the area of research for document image indexing, approaches based on partial document interpretation exist [8]. The images are automatically indexed using textual and graphical cues. The textual cues are obtained from the results proposed by an OCR system. The graphical indices are obtained by user annotation, or by an automatic procedure. In [9], Lorenz and Monagan present an automatic procedure. Junctions of adjacent lines, parallel lines, collinear lines and closed polygons are used as image features for indexing. Then, a weighting schema is used to reflect the descriptive power of a feature. In our paper, we also use term weighting but on a representation from a higher semantic level than the simple features used in [9].

The outline of this paper is as follows. Section 2 presents the graph representation used and shows how we create this representation from a document image. Section 3 presents the graph-mining method used. In Sect. 4, we describe how we search documents based on dissimilarities between bags of objects. Section 5 shows experimental results. We conclude the paper and outline perspectives in Sect. 6.

2 Graph Representation

Eight levels of representation for document images are proposed in [10]. These levels are ordered according to their aggregation relations. Data array, primitive, lexical, primitive region, functional region, page, document, and corpus level are the representation levels proposed.

Without loosing generality, in the following paragraphs we focus on a graph-based representation build from the primitive level. The primitive level contains objects such as connected components (set of adjacent pixels with the same color) and relations between them. From a binary (black and white) document image we extract connected components. The connected components are represented by graph nodes. On each connected component we extract features. In the current implementation, the extracted characteristics are rotation and translation invariant features based on Zernike moments [11]. These invariants represent the magnitudes of a set of orthogonal complex moments of a normalized image.

Let I be an image and $C(I)$ the connected components from I, if $c \in C(I)$, c is described as $c = (id, P)$, where id is a unique identifier and P the set of pixels the component contains. Based on this set P, we can compute the center for the connected component bounding box and we can also associate a feature vector to it. Based on that, $c = (id, x, y, v), v \in R^n$. Subsequently, using a clustering procedure on the feature vectors, we can label the connected component and reach the description $C = (id, x, y, l)$ where l is a nominal label. The graph $G(I)$ representing the image is $G = G(V(I), E(I))$. Vertices $V(I)$ correspond to connected components and are labelled with component labels. An edge between vertex u and vertex w exists if and only if $\sqrt{(u.x - w.x)^2 + (u.y - w.y)^2} < t$,

where t is a threshold that depends on the global characteristics of image I (size, number of connected components,...).

The following paragraph presents the clustering procedure used to associate a label to each connected component.

Clustering methods can by categorized into partitional and hierarchical techniques. Partitional methods can deal with large sets of objects ("small" in this context means less than 300) but needs the expected number of clusters in input. Hierarchical methods can overcome the problem of number of clusters by using a stopping criterion [12] but are not applicable on large sets due to their time and memory consumption.

In our case the number of connected components that are to be labelled can be larger than the limit of applicability for hierarchical clustering methods. On the other hand, we cannot use a partitional method because we do not know the expected number of clusters. Based on the hypothesis that a "small" sample can be informative for the geometry of data, we obtain in a first step an estimation for the number of clusters in data. This estimation is obtained using an ascendant clustering algorithm with a stopping criterion. The number of clusters found in the sample is used as input for a partitional clustering algorithm applied on all data.

We tested this "number of cluster estimation" approach using a hierarchical ascendant clustering algorithm [13] that employs Euclidean distance to compute the dissimilarity matrix, complete-linkage to compute between-clusters distances, and Calinsky-Harabasz index [12] as a stopping criterion. The datasets (T_1, T_2, T_3) (see Table 1) are synthetically generated and contain well separated (not necessary convex) clusters.

Table 1. Data sets description

T	$\lvert T\rvert$	number of clusters
T_1	24830	5
T_2	32882	15
T_3	37346	24

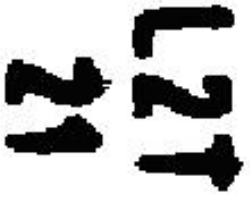
Considering S the sample extracted at random from a test set, in Table 2, we present predicted cluster numbers obtained for different sample sizes. After repeating the sampling procedure several times, we obtain a set of estimations for the number of clusters. We can see that by using a majority voting decision rule we can find the good number of clusters in most of the cases and even when the sample size is very small (50 or 100) compared to the data set size.

We used our sampling approach combined with the k-medoids clustering algorithm [14] on the connected components data set from images in our corpus (see Sect. 5). The k-medoids clustering algorithm is a more robust version of the well known k-means algorithm. The images from our corpus contain 6730 connected components. The proposed number of clusters using ten samples of size 600 is [16,14,17,16,16,19,7,17,15,16] and by considering the majority voting, we use 16 clusters as input to the partitional clustering algorithm.

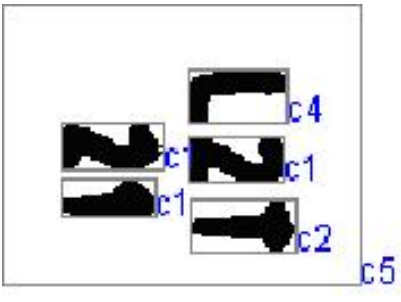
Table 2. Proposed cluster numbers

$\|S\|$	50	100	300	500	600	700
T_1	[6, 8, 7, 6, 5, 6, 6, 6, 5, 5] **6**	[5, 7, 9, 7, 5, 5, 7, 5, 5, 7] **5**	[7, 5, 7, 8, 7, 5, 5, 5, 7, 7] **7**	[8, 7, 5, 5, 5, 5, 5, 5, 5, 5] **5**	[5, 5, 5, 5, 5, 7, 7, 7, 7, 5] **5**	[5, 5, 7, 5, 7, 5, 5, 7, 5, 5] **5**
T_2	[9, 15, 15, 14, 13, 15, 13, 13, 14, 15] **15**	[15, 15, 13, 15, 15, 15, 15, 15, 15, 15] **15**	[15, 15, 15, 15, 15, 15, 15, 15, 15, 14] **15**	[15, 15, 15, 15, 15, 15, 15, 15, 15, 15] **15**	[15, 15, 15, 15, 15, 15, 15, 15, 15, 15] **15**	[15, 15, 15, 15, 15, 15, 15, 15, 14, 15] **15**
T_3	[11, 7, 9, 18, 7, 7, 6, 4, 14, 8] **7**	[6, 14, 23, 21, 7, 17, 23, 16, 12, 11] **23**	[22, 24, 23, 19, 23, 24, 21, 21, 24, 24] **24**	[21, 25, 25, 24, 22, 24, 23, 24, 24, 24] **24**	[20, 25, 21, 24, 19, 23, 24, 25, 24, 22] **24**	[23, 20, 21, 20, 25, 24, 24, 21, 25, 24] **24**

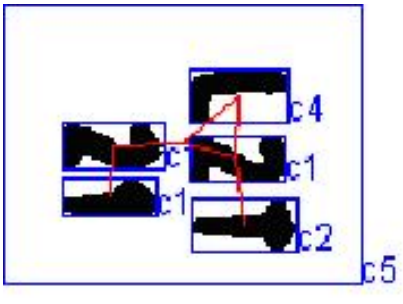
After labelling the connected components (nodes in the graph), we now describe the way these nodes are linked. The edges can be labelled or not (if unlabeled, the significance is Boolean: we have or don't have a relation between two



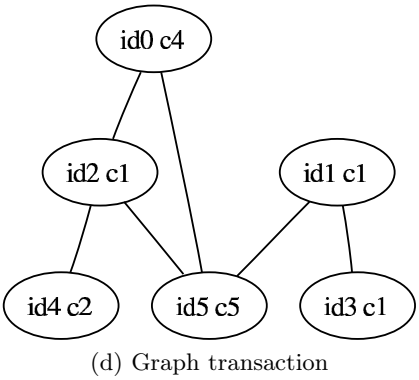
(a) Initial image



(b) Connected components labelling



(c) Graph construction



(d) Graph transaction

Fig. 1. An image and its associated graph transaction

connected components) and there can be relations of spatial proximity, based on "forces" [15], orientation or another criterion. In our current implementation the distance between centers of connected components is used (see Fig. 1). If the distance between two connected component centers is smaller than a threshold, then an edge will link the two components (nodes).

3 Graph Mining

"The main objective of graph mining is to provide new principles and efficient algorithms to mine topological substructures embedded in graph data" [5].

Mining frequent patterns in a set of transaction graphs is the problem of finding in this set of graphs those subgraphs that occur more times in the transactions than a threshold (minimum support). Because the number of patterns can be exponential, the complexity of this problem can also be exponential. An approach to solve this problem is to start with finding all frequent patterns with one element. Then, these patterns are the only candidates among which we search for frequent pattens with two elements, etc. in a level-by-level setting. In order to reduce the complexity, different constraints are used: the minimum support, the subgraphs are connected, and do not overlap.

The first systems emerged from this field are SUBDUE and GBI [5]. These approaches use greedy techniques and hence can overlook some patterns. The SUBDUE system searches for subgraphs in a single graph using a minimum description length-based criterion. Complete search for frequent subgraphs is made in an ILP framework by WARMR [5]. An important concept is that of maximal subgraph. A graph is said to be maximal if it does not have a frequent supergraph [16]. The graph-mining systems were applied to scene analysis, chemical components databases and workflows. A system that is used to find frequent patterns in graphs is FSG (Frequent Subgraph Discovery) that "finds patterns corresponding to connected undirected subgraphs in an undirected graph database" [17].

In our document image analysis context we are interested in finding maximal frequent subgraphs because we want to find symbols but to ignore their parts.

The input for the FSG program is a list of graphs. Each graph represents a transaction. FSG is effective in finding all frequently occurring subgraphs in datasets containing over 200,000 graph transactions [17]. We present subsequently how we construct the transaction list starting from a set of document images. Using the procedure presented in Sect. 2, we create for each document an undirected labelled graph.

Every connected component of this graph represents a transaction. We can further simplify the graphs by removing vertices that cannot be frequent and their adjacent edges. Using FSG we extract the frequent subgraphs and we construct a bag of graphs occurring in each document. In the following paragraphs, we consider that the frequency condition is sufficient for a group of connected components to form a symbol and we will conventionally make an equivalence between the frequent subgraphs found and symbols. As we can see in the exam-

ple (Fig. 2), the proposed symbols are far from being perfect due to the image noise, connected components clustering procedure imperfections... however we can notice the correlation between this artificial symbol and the domain symbols.

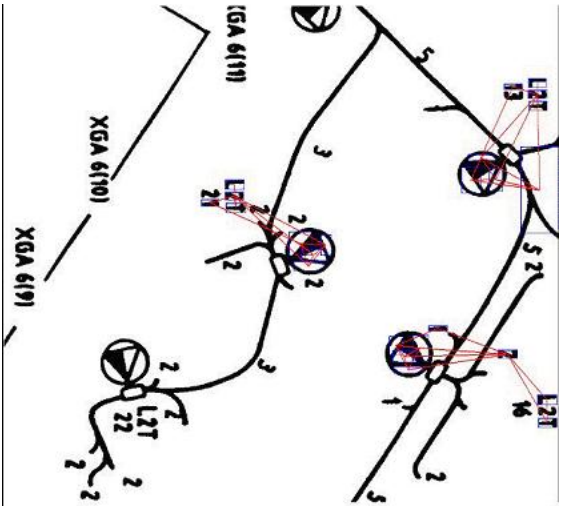


Fig. 2. Occurences of a frequent subgraph in an image

In conclusion, the subgraphs proposed as frequent are used to model a document as a bag of symbols. Because some documents may not contain any symbols, the document representation is based on two vectors containing connected components labels, and symbols labels.

$$A : (c_1, c_2, \ldots, c_n), (s_1, s_2, \ldots, s_m)$$

where c_i is the number of connected components labelled as i and s_j is the number of occurences of symbol j in document A.

4 Dissimilarity Between Document Descriptions

In this paragraph, we present the measure employed to qualify the dissimilarity between the descriptions of two document images.

A collection of documents is represented by a symbol-by-document matrix A, where each entry represents the occurrences of a symbol in a document image, $A = (a_{ik})$, where a_{ik} is the weight of symbol i in document k. Let f_{ik} be the frequency of symbol i in document k, N the number of documents in the collection, and n_i the total number of times symbol i occurs in the whole collection. In this setting, according to [18], one of the most effective weighting scheme is entropy-weighting. The weight for symbol i in document k is given by:

$$a_{ik} = \log(1 + f_{ik}).\left(1 + \frac{1}{\log N}\sum_{j=1}^{n}\frac{f_{ij}}{n_i}\log\frac{f_{ij}}{n_i}\right)$$

Now, considering two documents A and B with the associated weights $A = (a_1, a_2, \ldots, a_t)$, $B = (b_1, b_2, \ldots, b_t)$ where t is the total number of symbols, then

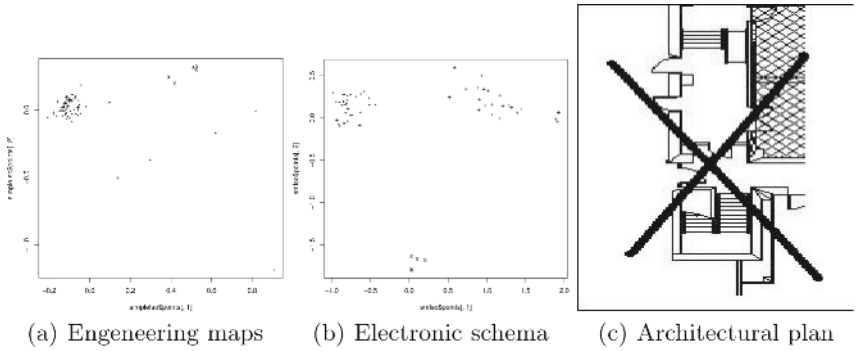
$$d(A, B) = 1 - \frac{\sum_{i=1}^{t} a_i.b_i}{\sqrt{\sum_{i=1}^{t} a_i^2 . \sum_{i=1}^{t} b_i^2}}$$

represents a dissimilarity measure based on the cosine correlation.

5 Experiments

The corpus used for evaluation contains 60 images from 3 categories: electronic (25 images) and architectural schemas (5 images) and engineering maps (30 images) (see Fig. 3). In order to present a corpus summary we employed a multidimensional scaling algorithm to represent in a two dimensional plot the dissimilarities between documents (see Fig. 4). Each document image is described with one of the following types of features: simple density and surface based characteristics (a vector with 30 components) or the connected components and symbol lists described above. In Fig. 4(a) we present the dissimilarities between images represented using simple features. In Fig. 4(b) are plotted the dissimilarities between the document images computed using the cosine correlation presented in Sect. 4. The engineering maps are plotted using '*' symbols, electronic schemas with '+' , and the architectural schemas with 'x'.

We further test the two representations in a classification context. Using a 10 fold stratified cross validation procedure and John C. Platt's sequential minimal optimisation algorithm for training a support vector classifier [19], we obtained the following results given in Tab. 3



(a) Engeneering maps (b) Electronic schema (c) Architectural plan

Fig. 3. Corpus images

Table 3. Classification results using the simple characteristics

Features	number of correctly classified instances	classification rates
only simple features	55	91.67 %
only bag of symbol representation	57	95 %
simple features and bag of symbol	58	96.67 %

We can see in Fig. 4 and the classification results (3) that the bag of symbols representation allows a better separation beetween image classes. This fact has an important influence on the quality of the query results.

A query can be an image, a list of symbols and connected components, or only one of the later lists.

$$\text{query: } (c_1, c_2, \ldots, c_n), (s_1, s_2, \ldots, s_m)$$

$$\text{query: } (s_1, s_2, \ldots, s_m)$$

$$\text{query: } (c_1, c_2, \ldots, c_n)$$

After using the graph mining algorithm on the presented corpus we obtain 52 frequent subgraphs. This subgraphs are the symbols that will be used in queries, and are numbered from 1 to 52. The description of the first 4 documents (in terms of what symbols and what are their corresponding frequencies) is subsequently presented :

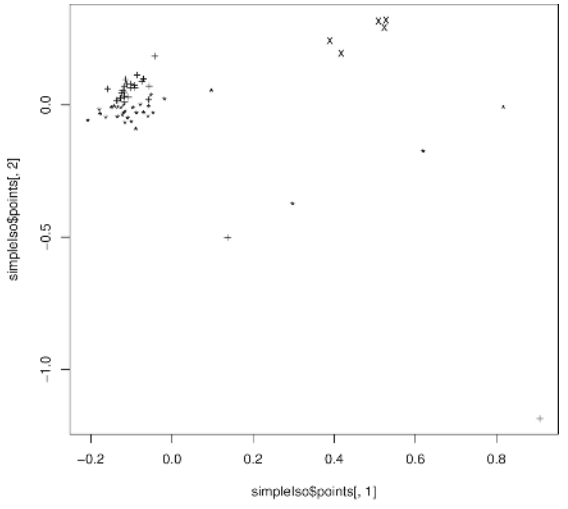
$$d_1 : \begin{array}{l}(s_1,1)(s_2,2)(s_3,3)(s_4,1)(s_5,4)(s_6,3)(s_7,2)(s_8,2)(s_9,2)(s_{13},1)(s_{14},1)(s_{16},3)\\(s_{17},2)(s_{18},1)(s_{19},4)(s_{20},6)(s_{21},1)(s_{22},4)(s_{23},2)(s_{24},4)(s_{25},2)(s_{26},2)\\(s_{35},1)(s_{36},1)(s_{37},1)(s_{41},1)(s_{45},1)(s_{46},1)(s_{49},1)(s_{51},1)(s_{52},1)\end{array}$$

$$d_2 : \begin{array}{l}(s_1,1)(s_2,3)(s_3,3)(s_4,2)(s_5,2)(s_6,1)(s_7,1)(s_{16},2)(s_{19},1)(s_{20},3)(s_{22},1)\\(s_{23},2)(s_{25},2)(s_{39},1)(s_{42},1)(s_{43},1)(s_{47},1)(s_{48},1)\end{array}$$

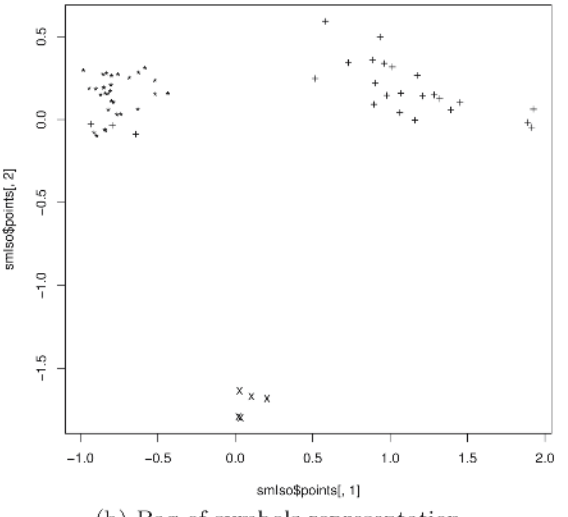
$$d_3 : \begin{array}{l}(s_1,1)(s_2,1)(s_3,4)(s_4,1)(s_7,1)(s_8,1)(s_{11},1)(s_{12},1)(s_{13},1)(s_{16},1)(s_{19},2)\\(s_{20},4)(s_{21},1)(s_{22},3)(s_{25},5)(s_{35},1)(s_{39},1)(s_{47},1)(s_{48},1)(s_{52},1)\end{array}$$

$$d_4 : \begin{array}{l}(s_1,4)(s_2,4)(s_3,3)(s_4,2)(s_5,2)(s_6,3)(s_7,1)(s_8,2)(s_9,2)(s_{11},3)(s_{12},1)(s_{16},1)\\(s_{18},1)(s_{19},4)(s_{20},4)(s_{21},2)(s_{22},1)(s_{23},3)(s_{24},4)(s_{25},2)(s_{26},2)(s_{36},1)\\(s_{37},1)(s_{39},2)(s_{40},2)(s_{41},1)(s_{42},1)(s_{44},1)(s_{46},1)(s_{47},2)(s_{48},2)(s_{49},1)\\(s_{51},2)(s_{52},2)\end{array}$$

In order to extract the formal description of a given query image we label the connected components of the query image, construct the graph, and employ graph matching to detect which symbols occur in the query image. At the end of this process the query image is described by the two lists of connected components and symbols.



(a) Simple geometric features



(b) Bag of symbols representation

Fig. 4. Document representations presented in a two dimensional space with respect to their reciprocal dissimilarities

In order to evaluate experimental results we used precision and recall measures. If A is the set of relevant images for a given query, and B is the set of retrieved images then :

$$\text{precision} = \frac{|A \cap B|}{|B|} \quad \text{recall} = \frac{|A \cap B|}{|A|}$$

As shown on Fig. 3, the corpus contains images that are scanned and contain real and artificial noise.

Table 4. Examples of queries and results

query	answer to query
$(s_1, 4)$	d_{37} dissimilarity=0.6782 d_{15} dissimilarity=0.7843 d_4 dissimilarity=0.8070 d_{13} dissimilarity=0.8450 d_{27} dissimilarity=0.8452
$(s_1, 4)(s_2, 4)(s_3, 3)(s_4, 2)$	d_2 dissimilarity=0.4233 d_7 dissimilarity=0.4722 d_{22} dissimilarity=0.4864 d_{25} dissimilarity=0.5046 d_{14} dissimilarity=0.5054
d_2 : $(s_1, 1)(s_2, 3)(s_3, 3)(s_4, 2)(s_5, 2)$ $(s_6, 1)(s_7, 1)(s_{16}, 2)(s_{19}, 1)(s_{20}, 3)$ $(s_{22}, 1)(s_{23}, 2)(s_{25}, 2)(s_{39}, 1)(s_{42}, 1)$ $(s_{43}, 1)(s_{47}, 1)(s_{48}, 1)$	d_2 dissimilarity=0.0065 d_{25} dissimilarity=0.1949 d_{22} dissimilarity=0.2136 d_{26} dissimilarity=0.2241 d_{21} dissimilarity=0.2362

Table 5. Queries recall and precision

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}
recall	0.75	0.5	0.48	0.55	0.56	0.76	0.6	0.4	0.32	0.16
precision	0.6	0.31	0.8	0.73	0.87	0.95	0.88	0.5	0.42	0.4

Table 4 gives 5 most relevant documents relative to the query.

Table 5 gives the recall and precision for 10 different queries. Queries Q1-4 represents symbol queries, i.e. as input is a list of symbols. The other queries are document images.

6 Conclusion

The research undertaken represents a novel approach for indexing document images. Our approach uses data mining techniques for knowledge extraction. It aims at finding image parts that occur frequently in a given corpus. These frequent patterns are part of the document model and can be put in relation with the domain knowledge.

Using the proposed method we reduce in an unsupervised manner the semantic gap between a user representation for a document image and the indexation system representation.

The exposed method can be applied to other graph representations of a document. In the near future, we will apply this approach to layout structures of textual document images.

Another follow up activity is to quantify the way noise affects the connected components labelling, and the manner in which an incorrect number of clusters can affect the graph mining procedure. Based on this error propagation study we

can further improve our method. Other possible improvements can be obtained if we would use a graph-based technique that can deal with error tolerant graph matching.

References

1. Antonacopoulos, A.: Introduction to Document Image Analysis. (1996)
2. Nagy, G.: Twenty years of document analysis in pami. IEEE Trans. on Pattern Analysis and Machine Intelligence **22**(1) (2000) 38–62
3. Pavlidis, T.: Algorithms for Graphics and Image Processing. Computer Science Press (1982)
4. Bagdanov, A.D., Worring, M.: Fine-grained document genre classification using first order random graphs. In: Proc. of the sixth International Conference on Document Analysis and Recognition. (2001) 79–83
5. Washio, T., Motoda, H.: State of the art of graph-based data mining. SIGKDD Explor. Newsletter **5**(1) (2003) 59–68
6. Fung, B.C.M., Wang, K., Ester, M.: Hierarchical document clustering using frequent items. In: Proc. of the SIAM Conference on Data Maining. (2003)
7. Termier, A., Rousset, M., Sebag, M.: Mining xml data with frequent trees. In: Proc. of DBFusion Workshop. (2002) 87–96
8. Doermann, D.: The indexing and retrieval of document images : A survey. Technical report, LAMP (1998)
9. Lorenz, O., Monagan, G.: Automatic indexing for storage and retrieval of line drawings. In SPIE, ed.: Storage and Retrieval for Image and Video Databases (SPIE). Volume 2420. (1995) 216–227
10. Blostein, D., Zanibbi, R., Nagy, G., Harrap, R.: Document representations. In: Proc. of the IAPR Workshop on Graphic Recognition. (2003)
11. Khotazad, A., Hong, Y.H.: Invariant image recognition by zernike moments. IEEE Trans. on Pattern Recogntion and Machine Analysis **12**(5) (1990)
12. Milligan, G.W., Cooper, M.C.: An examination of procedures for determining the number of clusters in a data set. Psychometrika **58**(2) (1985) 159–179
13. Gordon, A.D.: Classification. 2nd edition edn. Chapman & Hall (1999)
14. Kaufmann, L., Rousseeuw, P.J.: Clustering by means of medoids. In Dodge, Y., ed.: Statistical Data Analysis based on the $L1$ Norm and Related Methods, Elsevier Science (1987) 405–416
15. Tabbone, S., Wendling, L., Tombre, K.: Matching of graphical symbols in line-drawing images using angular signature information. International Journal on Document Analysis and Recognition **6**(2) (2003) 115–125
16. Yan, X., Han, J.: Closegraph: mining closed frequent graph patterns. In Press, A., ed.: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2003) 286–295
17. Kuramochi, M., Karypis, G.: An efficient algorithm for discovering frequent subgraphs. IEEE Transactions on Knowledge Data Engeneering **16**(9) (2004) 1038–1051
18. Dumais, S.T.: Improving the retrieval information from external ressources, behaviour research methods. Instrument and Computers **23**(2) (1991) 229–236
19. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods - Support Vector Learning, MIT Press (1998)

A Minimal and Sufficient Way of Introducing External Knowledge for Table Recognition in Archival Documents

Isaac Martinat[1] and Bertrand Coüasnon[2]

[1] IRISA/INSA, Campus universitaire de Beaulieu, F-35042 Rennes Cedex, France
Isaac.Martinat@irisa.fr
[2] IRISA/INRIA, Campus universitaire de Beaulieu, F-35042 Rennes Cedex, France
Bertrand.Couasnon@irisa.fr

Abstract. We present a system that recognizes tables in archival documents. Many works were carried out on table recognition but very few on tables of historical documents. These are difficult to analyze because they are often damaged due to their age and conservation. Therefore we have to introduce knowledge to compensate for missing information and noise in these documents. As there is a very important number of documents of a same type, the cost is not significant to introduce this explicit knowledge. We also want to minimalize the cost to adapt the system for a given document type. The precision of the knowledge given by the user is dependent on the quality of the document. The more the document is damaged, the more the specification has to be precise. We will show in this article how an external minimal knowledge can be sufficient for an efficient recognition system for tables in archival documents.

Keywords: Archival documents, knowledge specification, structured document analysis, table recognition.

1 Introduction

We present a system that recognizes tables in archival documents. Many works were carried out on table recognition [1, 2] but very few on archival tables. These are difficult to analyze because they are often damaged due to their age and conservation. We will only analyze tables with ruling separators between columns and rows. The rulings can be broken, skewed or curved. Another difficulty is that ink bleeds through the paper, thus rulings of flip side can be visible. For these reasons, these tables are very difficult to recognize.

The problem in recognizing archival documents is that these documents have missing information and can contain false information like flip side rulings or stains. Therefore, the user has to give knowledge to compensate these analysis difficulties. However, this knowledge has to be minimal for a fast adaptation between different document types. It has to be simple, so non-document analysis specialists can easily define it. This minimal knowledge must be sufficient to help the system to recognize these difficult documents. Therefore, we have to define a minimal and sufficient knowledge for the archival table recognition.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 206–217, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this paper, we will first present the related work on table recognition and on archival document analysis. Furthermore, we will show with the knowledge specification of the DMOS method the necessity for archival documents to give precise knowledge. Section 4 proposes for archival table recognition the necessary knowledge and explains our system uses it. We will finally show our results before to conclude on our work.

2 Related Work

2.1 Table Form Analysis

Many works were carried out on table recognition [1, 2]. We will present only the works on table and form recognition with rulings.

Handley [3] presented a method for table analysis with multi-line cells. This method first extracts from the image word boxes and rulings. Rulings whose end points are closed, are stitched together. Then for each word box, close rulings are researched and a frame is associated for each word box. This method merges word boxes with identical frames. To recognize rows and columns not separated by rulings, it then uses histogram procedure on the two axes. However, this detection is inefficient on curved documents. This method detects only broken rulings with small gaps. The method proposed in [4] detects from a binary image line segments in using erosion and dilation operations. This line segment extraction fills some breaks of form lines. They also used rules to detect bigger gaps, but these gaps are only detected in specific cases. Hori and Doermann [5] reduced the original image. In the reduced image, broken lines can be changed in solid lines but the size of detected gaps depends on factor reduction. The method proposed in [6] analyzes telephone company tables. It can recognize rulings with gaps but user has to give the maximal gap size to group segment lines.

These methods deal with broken lines but only small gaps are filled, or these gaps must be under certain conditions. Archival documents can be very damaged and can contain big gaps. Therefore these methods can not be adapted to archival documents.

2.2 Ancient Document Analysis

Few works were carried out on archival document analysis. The analysis of these documents is difficult because they are quite damaged. These documents have annotations, are torn and ink bleeds through the paper. Therefore a recognition system for archival documents needs knowledge given from the user.

He et al. [7] used a graphical interface to recognize archive biological cards. Each card contains bibliographic data and other information for one genus-group or one species-group, there are in total about ten text fields and the most of information is typewritten. The user defines boxes with this interface and labels each box. From this one a template is created, then the user can add information. With fuzzy positions, a X-Y cut method is used to analyze cards and a matching algorithm is applied between the template and the analysis. This system is

specific for archive biological cards. This method uses positions from the graphical interface and fuzzy positions to analyze documents but it is efficient only on documents of a same type which have not important variations. Esposito et al. [8] designed a document processing system *WISDOM++* that has been used on archival documents (articles, registration card). This system segments the document with a hybrid method, global analysis and local analysis. The result of this analysis can be modified by the user. Training observations are generated from these user operations. With these results, the document is then associated to a class of model documents. The method presented in [9] analyzed lists of Word War II, which do not contain rulings. For a set of documents containing the same logical structure, historians and archivists use a graphical interface to define a *template* where physical entities on a page are associated with logical information. All these methods use physical information from a model generated by a graphical interface or learned on a set of documents corrected by a user. The variations between documents of a same type depend on the matching algorithm between the image and the model. Furthermore it takes time for an user to give the model information.

For the recognition of tables with rulings, Tubbs et al. recognized 1910 U.S. census tables [10] but coordinates for each cell of the tables are given at hand in an input of 1,451 file lines. The drawback of this method is the long time spent by the user to define this description. Furthermore, the coordinate specifications do not allow variations on the documents of the defined type. Nielson et al. [11] recognized tables whose rows and columns are separated by rulings. Projection profiles are used to identify rulings. For each document a mesh is created, and individual meshes are combined to form a template with a single mesh. This method cannot process documents where rulings are skewed or curved. Individual meshes must be almost identical to be combined.

Archival documents are often damaged and recognition systems need an user specification to recognize these. The general systems presented in Sect. 2.1 cannot process these documents because they do not detect broken lines with big gaps. To help the archival document recognition, systems use an user description [10], a graphical interface [7, 9], information of other documents of the same type [11] or user corrections [8]. These works use external knowledge. However, it is often quite long to define and too precise, so these systems do not allow important variations between documents.

A system to recognize archival documents needs an external knowledge, so we propose in this article a minimal knowledge for the recognition of archival tables. We will show how this knowledge is simple, fast to give to the system, independent of physical structure if document is not too damaged and sufficient to recognize very damaged documents.

3 Knowledge Specification with DMOS Method

With the DMOS (Description and Modification of Segmentation) method we can give a description for a document type. DMOS is a generic recognition method

for structured documents [12]. This method is made of a grammatical formalism EPF (Enhanced Position Formalism) and an associated parser which is able to change the parsed structure during the parsing. With the DMOS method we can build a system for a kind of document by defining a description of the document with an EPF grammar. This grammar is then compiled to produce a recognition system. We will show how the knowledge is represented in EPF formalism and the necessity for archival documents to have a very precise description.

3.1 Knowledge Representation in EPF Formalism

With the DMOS method and the EPF formalism, a system is created much faster than to develop completly a new recognition system. EPF can be seen as an adding of several operators to mono-dimensional grammars like the principal one, the *Position operator (AT)*. For example, `A && AT(pos) && B` means A, and at position **pos** in relation to A, we find B.

The DMOS method is generic because the EPF formalism allows to define very different kinds of structured documents. This method was tested, for example, on musical scores, on mathematical formulae, on table structures [12] and on archival documents [13].

3.2 General and Specific Systems in EPF

A general system was built in EPF formalism to analyze all kinds of table-forms [14]. This system can recognize the hierarchical organization of a table made with rulings, whatever the number/size of columns/rows and the depth of the hierarchy contents in it. However we [14] showed that this general system was not able to be applied for archival documents. These documents are damaged and gaps in rulings are too large, which makes it impossible for a general system to decide if there is a gap or a normal absence of ruling. Therefore, a much more precise description is necessary to recognize these. A system was built for military forms of the 19th Century. A grammar describing these forms and the relative positions of the cells was written in EPF. It has been tested on 164,479 forms and 98.73% were well recognized with correct cell positions. There was no bad recognition. Another system was built for naturalization decrees [13]. These documents are on two columns separated by spaces. These systems are efficient on archival documents. However, even if these descriptions in EPF are faster to write than to develop a specific system, they are still quite long to define and accessible only for document analysis specialists.

4 Knowledge for Archival Table Recognition and Recognition System

Our goal is to propose a specific system for archival tables. For archival documents, the user has to give knowledge to compensate for missing information in these. DMOS is an efficient method but descriptions in EPF can be still long

to define. Furthermore, it is difficult to define a precise knowledge for damaged documents.

The proposed system is specific but it can deal with a large variety of tables and with a fast adaptation. The specified knowledge can be given by a non-specialist user. Thus it must be easy to specify, minimal but sufficient to help the system. We will show the necessary knowledge for archival tables and how our system uses it.

4.1 Necessary Knowledge Formalization

We have a very important number of documents to process. For example we have a dataset of about 130,000 census tables from 1817 to 1968. These censuses were carried out on 24 different years and often different from a year to another. Therefore, we have an important document quantity of the same type (about 5,400 images) so that the time used to give this short specification is not significant. We can ask the user to spend little time to define an external knowledge if the latter is useful for a large quantity of documents.

We want to adapt quickly the recognition system to a large variety of tables. The knowledge introduced by the user has to be simple, so an archivist can give this specification. Therefore document analysis parameters (gap size between two line segments to form a line, ruling minimal size . . .) cannot be used for this purpose. We want to minimize the specification given by the user but the system needs enough precise specification to be efficient. This knowledge must be minimal and sufficient to help the system for the document recognition. Thus user can give specification in relation to document quality, if document is good quality few informations are necessary but more precise informations are necessary for very damaged documents.

For a table, the minimal knowledge can be the number of rows and the number of columns. In Fig. 1, we show on the left example that this information is sufficient to help the system to recognize a synthetic document which misses information. In this example, a system not adapted to archival documents will recognize only two rows. However with the user specification that the number of rows is three, system can detect the line segment for a row separator ruling.

For a grade table of 25 students, the user gives the following specification using the number of rows and columns or the name of each column:

```
[ rowNumber 25 , colNumber 3 ] Or
[ rowNumber 25, col "last name", col "first name", col "grade" ]
```

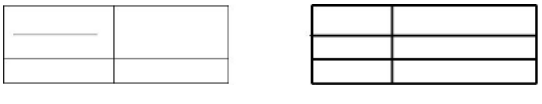


Fig. 1. left: synthetic image illustrating missing information, right: structure with 3 rows and 2 columns to recognize

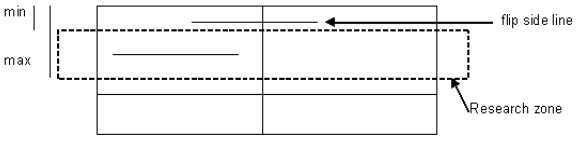


Fig. 2. Example with the same structure as previous example to illustrate the knowledge introduction more specific to detect ambiguous cases

For more damaged documents, the previous knowledge can be insufficient. A more precise specification must be given by the user to process more difficult documents. On another example (Fig. 2) with the same user specification as previously we show that the row number is not sufficient. If, on the document, ink bleeds through the paper, a false line segment is detected because of a visible flip side ruling. The following detected line segment is a row separator, but it has an equal length to the false line segment. Therefore, the system cannot decide which line segment is a row separator. However, if the user specified a minimal size of rows large enough to avoid the false line segment, the system will research the row separator ruling in a research zone that does not contain the false line segment.

For columns and rows, minimal and maximal global sizes can be given by the user. These sizes are used for every row or column. Sizes can be given in pixels or if the document density is known, sizes can also be given in centimeters or in inches. An user can give the following specification with global sizes:

```
[ rowMin 20, rowMax 150,colMin (cm 1.0), colMax (cm 8.0),
  rowNumber 25, col "last name", col "first name", col "grade" ]
```

When documents are very damaged, if these global sizes are not sufficient, the user can give specific sizes for each column/row or for a specific column/row. Column and row sizes are more constrained but they can have some variations between documents. In this example, a grade is given in digits, so the user can give a small size for this column with this following specification:

```
[ rowNumber 25, col "last name", col "first name",
  colsize (inch 1.2) (inch 2.3) "grade" ]
```

The user gives a specifications in relation to the quality of the document. He will give only the necessary knowledge. When archival documents are not too damaged, only the numbers of rows and columns are necessary. On the other hand, when documents are very damaged, the user can specify more precise knowledge to help the system make the right choice when it recognizes a document.

4.2 System Defintion

To build a document analysis system, we need to choose constraints. This choice is difficult because if we choose too many constraints, documents will be undersegmented. However, if we choose too few constraints, documents will be

oversegmented. For example, to detect a broken ruling, we have to choose the gap size between two line segments to decide if they belong to the same ruling. If the size is too small, few broken rulings will be detected, but if the size is too big, false rulings could be detected. The knowledge given by the user helps the system to decide which ruling has to be detected.

Our recognition is made of three steps, the first one is the detection of table borders, the second one is the column detection from right border to left border and the last one is the row detection from top border to bottom border. The two last steps use the user specification and they allow to adapt constraints for the recognition.

4.3 Use of External Knowledge

We have shown the advantages of the DMOS method: its efficiency and its associated EPF formalism. The EPF formalism allows a document analysis specialist to define quite quickly a recognition system. Therefore we used this method to define the proposed system for archival tables. The latter takes in argument a knowledge easy to define for a non-specialist of document analysis, for example an archivist.

Number of Rows and Columns. The system tries to detect the number of rows N and columns according to the user specification. As for rows, the system from the top border detects the row separators. Gap size is fixed to a small value, thus the system can not oversegment the table. However, if the bottom border is detected and the number of detected rows is less than N, the document is undersegmented, some rows were not detected. Gap size is then increased and the length ratio is decreased to allow the system to detect more broken lines. Length ratio is the ratio between the top border and the detected line. The system tries again the recognition until N rows are detected or if constraints are too weak, i.e the gap size is too big or the length ratio is too small. This method is written easily with several rules in the EPF formalism, we removed some arguments to simplify the writing. We will show the other arguments to explain how sizes are used by the system.

```
findRows Gap LengthR TopLine N ::=
  not (findRowSep Gap TopLine N ListDetectedLines) &&
  ''(NewGap is Gap + IncrGap, LengthR2 is LengthR - DecrRatio) &&
  findRows NewGap LengthR2 TopLine N.
```

`findRowSep` searchs N row separators from the top border. This rule fails when the bottom border is detected and the number of recognized rows is less than N. To check that a false row was not detected, the last line of the list of detected lines must be the bottom border. Otherwise, the system stops and informs the user. It is defined for columns as for rows.

Sizes. For the recognition of rows and columns, the system uses sizes given by the user when sizes were specified. If the user did not give sizes, minimal size is 0 and maximal size is the image size (width for columns and height for rows).

```
findColSep GlobalMin GlobalMax RightLine [Col|ListeCols] ::=
    getSize Col GlobalMin GlobalMax Min Max &&
    findLineV Gap Min Max RightLine LeftLine &&
    ''( sameSize RightLine LeftLine LengthRatio ) &&
    findColSep GlobalMin GlobalMax RightLine ListeCols.
```

`getSize` returns the *Min* and *Max* sizes in relation to the user specification. `findLineV` finds a broken vertical line nearest to the left of RightLine at a distance between GlobalMin and GlobalMax and `sameSize` is true if the Ratio of *LeftLine* and *RightLine* is greater than LengthRatio. By recursivity, the system detects then the other vertical lines.

If the user specifies sizes for a specific column, these sizes are used to search the next vertical line.

```
findLineV Gap Min Max RightLine LeftLine ::=
   AT (nearLeft Gap Min Max RightLine) &&
   brokenLineV Gap LeftLine && ''(parallel RightLine LeftLine).
```

With the *AT* operator, we defined the research zone to find *LeftLine* from *RightLine*, *Min* and *Max* values define the zone width, and zone height is defined with Gap value.

The EPF formalism has allowed us to define quickly a system using external knowledge. This description show how the system uses the knowledge given by the user.

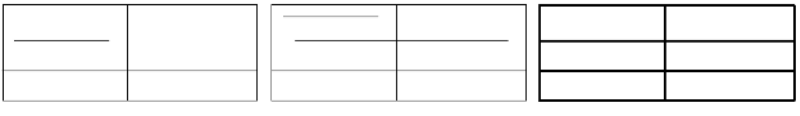


Fig. 3. left: synthetic image illustrating missing information, middle: synthetic image illustrating false ruling, right: structure with 3 rows and 2 columns to recognize

4.4 System Efficiency

We show in Fig. 3 how the constraint adaptation makes our system efficient. With a weak constant constraint of minimal ruling size, the system would recognize the middle example with four rows instead of three rows. With a strong enough constant constraint, the left example would be recognized with only two rows. Whereas our system, on the left example, begins the recognition with strong constraints and does not detect three rows so it will try again with less constraints until it recognizes the structure specified (`[rowNumber 3, colNumber 2]`). On the middle example, our system begins the recognition with strong constraints, the shortest ruling is not recognized as a row separator so the system will correctly recognize the structure. Therefore, it is very important to begin the recognition with strong constraints and to reattempt with less strong constraints only if the document is not well recognized.

5 Results

We show on some documents how the user specification helps the recognition system.

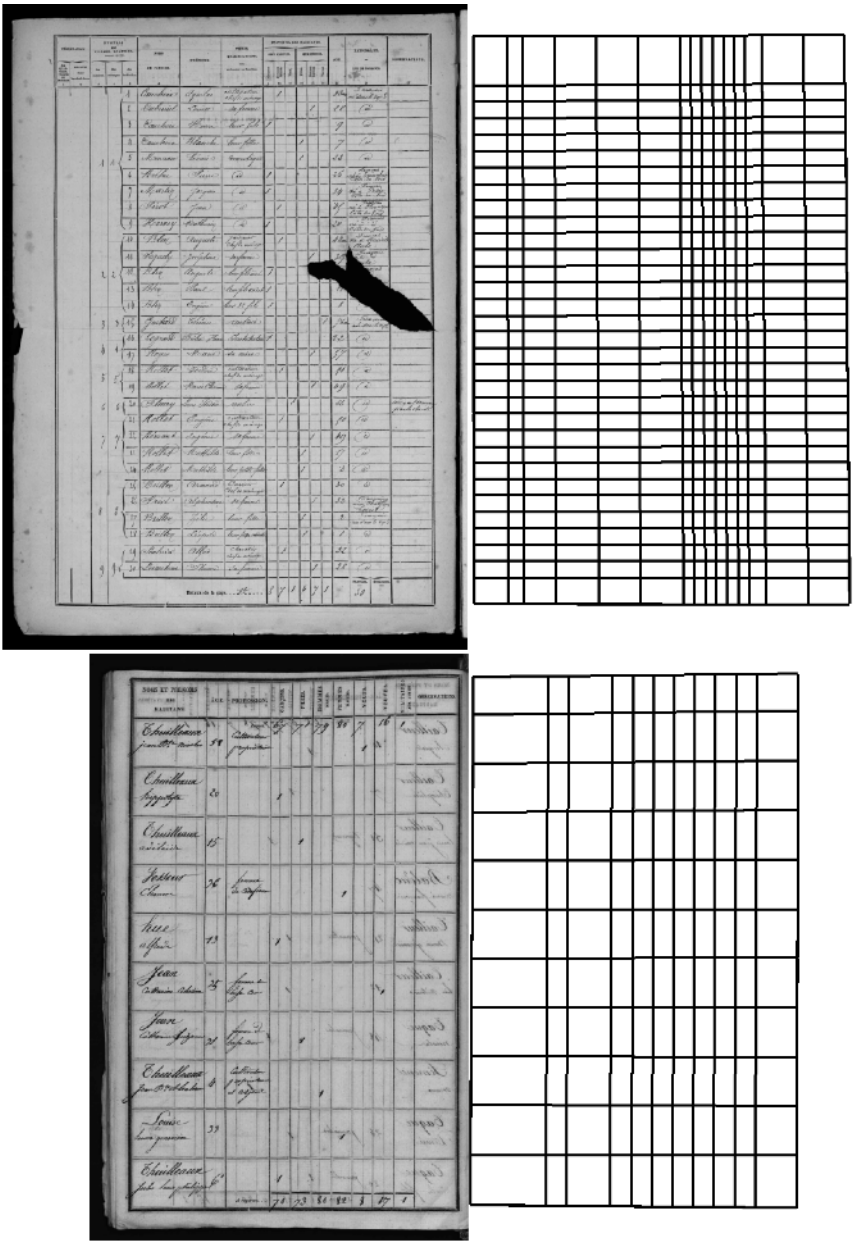


Fig. 4. Example on archival documents, on right the recognized structure

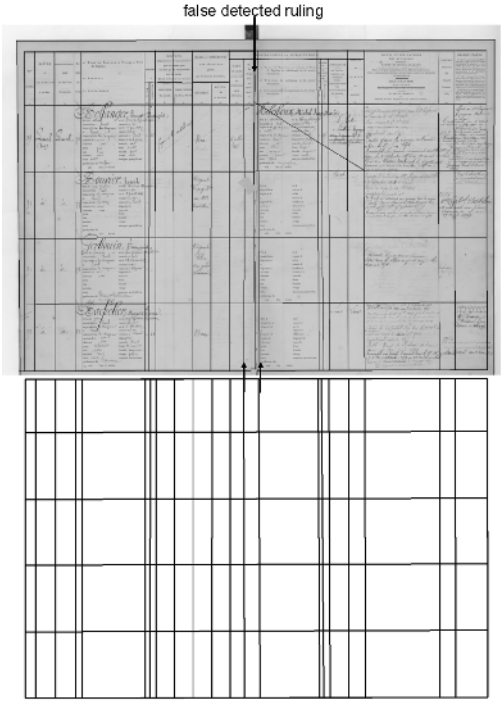


Fig. 5. Example on a two page document where with a specific column size a false detected ruling is avoided, on bottom the recognized structure

The document in the top of Fig. 4 is very damaged. The following specification:
`[ rowMin 80, rowNumber 32, colNumber 15]` allows to recognize this document. The column number is sufficient to detect columns even if the paper is very torn. At the first step, the last columns are not detected, the gaps in column separators are too big. Thus the system tries again several times the column recognition by increasing the gap size value until the right number of columns is detected. As for rows, we need to give a minimal size to avoid detecting flip side rulings. Therefore, the system will research in zones which do not contain them.

In the document on the bottom of Fig. 4, vertical flip side rulings are visible. To avoid detecting these rulings we have to give general column sizes and specific sizes with the following specification:

```
[ rowNumber 11, colMin 100, colMax 500,
colsize  200 500 "names", col "age", colsize 200 500 "profession",
col "boys", col "girls", col "bridegroom", col "bride",
col "widower", col "widow",col "military", col "observations" ].
```

Figure 5 shows a result on an archival document of two pages. A false ruling can be detected with the separation of these two pages. Therefore, to avoid this

problem, the user can specify the minimal size for the column containing the false ruling. Therefore when this column is detected, the system from the right column separator searchs the left column separator to a distance greater than the distance between the right separator and the false ruling. In this case, the user cannot give a general minimal size for all columns because on this document, there are very small columns that would not be detected.

On our first tests, we tested our system on 62 tables with the same specification and we checked 1922 cells. Only 2 adjacent cells were not well detected. Handwriting was present on the row separator and a false segment was detected from this handwriting. Table recognition evaluation is not easy so we need much more time to check results on a much more important number of documents which have been recognized. We have demonstrated on these results how the minimal knowledge that we proposed is easy to define and useful for the recognition system.

6 Conclusion

We have shown in this article how archival tables are very difficult to process because they can be very damaged. An external knowledge is necessary to help the recognition system to analyze these. This knowledge allows the system to recognize a structure which misses information and containing false information. To adapt this system quickly and to facilitate the introduction of this knowledge, we defined a minimal one. We have also presented how this minimal knowledge is sufficient and how that is easy for a user to give this specification. We presented on some results how our system is able to recognize very difficult documents.

Our future work is to design a much more general language, simple to use and sufficient to recognize all kinds of archival documents with tabular structures. We will seek to define a minimal and sufficient knowledge for more complicated tables: tables whose rows and columns can be separated by spaces, tables with recursive structure and forms.

Acknowledgments

This work has been done in cooperation with the *Archives départementales des Yvelines* in France, with the support of the *Conseil Général des Yvelines.*

References

1. Lopresti, D.P., Nagy, G.: A tabular survey of automated table processing. In: Selected Papers from the Third International Workshop on Graphics Recognition: Recent Advances and Perspectives. Volume 1941 of LNCS., Springer (2000) 93–120
2. Zanibbi, R., Blostein, D., Cordy, J.R.: A survey of table recognition. International Journal of Document Analysis and Recognition (IJDAR) **7**(1) (2004) 1–16
3. Handley, J.C.: Table analysis for multiline cell identification. In: Proceedings of SPIE – Volume 4307 Document Recognition and Retrieval VIII. (2000) 34–43

4. Xingyuan, L., Gao, W., Doermann, D., Oh, W.G.: A robust method for unknown forms analysis. In: 5th International Conference on Document Analysis and Recognition (ICDAR 1999), Bangalore, India (1999) 531–534
5. Hori, O., Doermann, D.S.: Robust table-form structure analysis based on box-driven reasoning. In: 3th International Conference on Document Analysis and Recognition (ICDAR 1995), Montreal, Canada (1995) 218–221
6. Chhabra, A.K., Misra, V., Arias, J.F.: Detection of horizontal lines in noisy run length encoded images: The fast method. In: Selected Papers from the First International Workshop on Graphics Recognition, Methods and Applications. Volume 1072 of LNCS., Springer (1996) 35–48
7. He, J., Downton, A.C.: User-assisted archive document image analysis for digital library construction. In: 7th International Conference on Document Analysis and Recognition (ICDAR 2003), Edinburgh, UK (2003) 498–502
8. Esposito, F., Malerba, D., Semeraro, G., Ferilli, S., Altamura, O., Basile, T.M.A., Berardi, M., Ceci, M., Mauro, N.D.: Machine learning methods for automatically processing historical documents: From paper acquisition to xml transformation. In: 1st International Workshop on Document Image Analysis for Libraries (DIAL 2004), Palo Alto, CA, USA (2004) 328–335
9. Antonacopoulos, A., Karatzas, D.: Document image analysis for world war 2 personal records. In: 1st International Workshop on Document Image Analysis for Libraries (DIAL 2004), Palo Alto, CA, USA (2004) 336–341
10. Tubbs, K., Embley, D.: Recognizing records from the extracted cells of microfilm tables. In: ACM Symposium on Document Engineering. (2002) 149–156
11. Nielson, H., Barrett, W.: Consensus-based table form recognition. In: 7th International Conference on Document Analysis and Recognition (ICDAR 2003), Edinburgh, UK (2003) 906–910
12. Coüasnon, B.: Dmos: A generic document recognition method, application to an automatic generator of musical scores, mathematical formulae and table structures recognition systems. In: 6th International Conference on Document Analysis and Recognition (ICDAR 2001), Seattle, WA, USA (2001) 215–220
13. Coüasnon, B., Camillerapp, J., Leplumey, I.: Making handwritten archives documents accessible to public with a generic system of document image analysis. In: 1st International Workshop on Document Image Analysis for Libraries (DIAL 2004), Palo Alto, CA, USA (2004) 270–277
14. Coüasnon, B.: Dmos, a generic document recognition method: Application to table structure analysis in a general and in a specific way. International Journal of Document Analysis and Recognition (IJDAR) (To be published)

Database-Driven Mathematical Character Recognition

Alan Sexton and Volker Sorge

School of Computer Science, University of Birmingham, UK
{A.P.Sexton, V.Sorge}@cs.bham.ac.uk
http://www.cs.bham.ac.uk/~aps|~vxs

Abstract. We present an approach for recognising mathematical texts using an extensive LaTeX symbol database and a novel recognition algorithm. The process consists essentially of three steps: Recognising the individual characters in a mathematical text by relating them to glyphs in the database of symbols, analysing the recognised glyphs to determine the closest corresponding LaTeX symbol, and reassembling the text by putting the appropriate LaTeX commands at their corresponding positions of the original text inside a LaTeX picture environment. The recogniser itself is based on a novel variation on the application of geometric moment invariants. The working system is implemented in Java.

1 Introduction

Automatic document analysis of mathematical texts is highly desirable to further the electronic distribution of their content. Having more mathematical texts, especially the large back catalogues of mathematical journals, available in rich electronic form could greatly ease the dissemination and retrieval of mathematical knowledge. However, the development of sophisticated tools necessary for that task is currently hindered by the weakness of optical character recognition systems in dealing with the large range of mathematical symbols and the often fine distinctions in font usage in mathematical texts. Research on developing better systems for mathematical document analysis and formula recognition requires high quality mathematical optical character recognition (OCR). As one approach to this problem, we present in this paper a database-driven approach to mathematical OCR by integrating a recogniser with a large database of LaTeX symbols in order to analyse images of mathematical texts and to reassemble them as LaTeX documents.

The recogniser itself is based on a novel application of geometric moments that is particularly sensitive to subtle but often crucial differences in font faces while still providing good general recognition of symbols that are similar to, but not exactly the same as, some element in the database. The moment functions themselves are standard but rather than being applied just to a whole glyph or to tiles in a grid decomposition of a glyph, they are computed in every stage of a recursive binary decomposition of the glyph. All values computed at each level of the decomposition are retained in the feature vector. The result is that the feature vector contains a spectrum of features from global but indistinct at the high levels of the decomposition to local but precise at the lower levels. This provides robustness to distortion because of the contribution of the high level features, but good discrimination power from those of the low levels.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 218–230, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Since the recogniser matches glyphs by computing metric distances to given templates, a database of symbols is required to provide them. We have developed a large database of symbols, which has been extracted from a specially fabricated document containing approximately 5300 different mathematical and textual characters. This document is originally based on [7] and has been extended to cover all mathematical and textual alphabets and characters currently freely available in LaTeX. It enumerates all the symbols and homogenises their relative positions and sizes with the help of horizontal and vertical calibrators. The single symbols are then extracted by recognising all the glyphs a symbol consists of as well as their relative positions to each other and to the calibrators. Each entry in the database thus consists of a collection of one or more glyphs together with the relative positions and the code for the actual LaTeX symbol they comprise. The basic database of symbols is augmented with the precomputed feature vectors employed by the recogniser.

To test the effectiveness of our OCR system, we analyse the image of a page of mathematics, and reproduce it by locating the closest matching LaTeX symbols and constructing a LaTeX file which can then be formatted to provide a visually close match to the original image. At this stage there is no semantic analysis or syntactic parsing of the results to provide feedback or context information to assist the recognition. As a result, the source produced is merely a LaTeX picture environment that explicitly places, for each recognised character, the appropriate LaTeX command in its correct location. However, it is our position that the information obtained in order to do this successfully is an appropriate input to the higher level analysis required for further document analysis — especially as we can provide, for each glyph, a sequence of alternative characters in diminishing order of quality of visual match. Moreover, the database-driven analysis offers us a way to effectively deal with symbols composed of several, disconnected glyphs, by easily finding, analysing, and selecting all symbols from the database that contain a component glyph that matches the glyph in question.

There has been work on collecting a ground truth set of symbols for mathematics for training and testing purposes. Suzuki et al [13] have compiled a database of symbols, with detailed annotations, from a selected set of mathematical articles. The just under 700,000 characters in their database include many different instances of the same characters, each with true, rather than artificially generated degradation. The actual number of different symbols is much smaller. Our database only contains non-degraded ideal symbols. However, each symbol is generated by a different LaTeX command and so there are relatively few copies of the same glyph in the database. In practice, there is still some duplication in our database because (a) font developers often create new fonts by copying and modifying existing fonts, sometimes leaving some symbols unchanged, and (b) two different multi-glyph symbols often contain copies of one or more of the same component glyphs, e.g. "=" and "≡". Thus Suzuki et al's database is especially suitable for test purposes and for OCR system training on the types of mathematics that appears in the necessarily limited (but large) set of symbols contained in the articles it was extracted from, whereas our database is less suitable for testing purposes but has significantly more breadth in that it contains most supported LaTeX symbols. In particular, we can deal with the rapidly growing number of symbols used in diverse scientific disciplines such as computer science, logics, and chemistry.

A large number of methods for symbol recognition have been studied. See [6] for a high level overview. In our system, which is a development from previous work on font recognition [11], we augment the basic database of symbols with precomputed feature vectors. This database serves as the template set of our character recogniser. The recogniser itself is based on a novel variation on geometric moment invariants. There has been much work on various approaches to character recognition, and geometric moment invariants have been popular [15].

The paper is structured as follows: We present our recogniser and the database of glyphs in Section 2 and 3, respectively. We give an example for the recognition of an involved mathematical expression by our algorithm in Section 4, and conclude in Section 5.

2 A Novel Algorithm for Mathematical OCR

Our algorithm for symbol recognition does not depend on a segmentation of the image into full characters. Instead we segment the image into individual connected components, or glyphs, where each symbol may be composed of a number of glyphs. The motivation for this is that because of the 2-dimensional layout in mathematical texts, and in technical diagrams, we do not necessarily have the luxury of having the symbols neatly lined up on baselines. Hence segmentation into symbols is much less reliable in such texts. Instead, our approach is to identify individual glyphs and compose separate glyphs together to form symbols as indicated by the presence of appropriate glyph components in corresponding relative positions.

We assume that preprocessing operations such as deskewing, binarisation etc. have already been applied. The algorithm proceeds by

- extracting glyphs from the image;
- calculating a feature vector for the glyph based on recursive image partitioning and normalised geometric moments;
- for each glyph, producing a list of potentially matching glyphs from the glyph database ordered by metric distance from the target glyph and identifying an initial best match for the glyph.

2.1 Extracting Glyphs

Since our feature vector is based on normalised geometric moments of sub-rectangles of the glyph image, we need an appropriate data structure to enable convenient and efficient calculation of such values. Also we need to separate the image into a list of glyph representations, where each glyph is a single connected collection of pixels. We base our moment calculations on that given by Flusser [2], where boundaries of glyphs are used. To take advantage of that, our glyph representation is a list of horizontal line segments, with each segment being represented as the start and end horizontal positions of the row together with the vertical position that the segment occurs at.

Using this representation, the glyphs can be extracted in a single scan down the image. To do so, a set of open glyphs and a set of closed glyphs is maintained. A closed glyph is one which cannot have any further horizontal line segments added to it (because

no such segment could possibly touch any existing line segment in the glyph). An open glyph is one which has at least one line segment on the horizontal row immediately above the one about to be scanned and hence a line segment could be found on the current row which may touch the glyph. In detail the algorithm proceeds as follows:

```
for each horizontal scan line of the image
  for each line segment on the line
    find the set of open glyphs that the line segment touches
    if the set is empty
      create a new open glyph containing the line segment
    else if the set contains only one open glyph
      add the line segment to the open glyph
    else
      merge all open glyphs in the set and add the line segment to the resulting open glyph
  for each open glyph that did not have a line added to it and was not merged
    remove it from the set of open glyphs and add it to the set of closed glyphs
add any remaining open glyphs to the set of closed glyphs; return the set of closed glyphs
```

The above algorithm copes with glyphs with holes e.g. "8", and those which are open at the top, such as "V" or "$\bigcup$". Note that it identifies each of "Θ", "$\oplus$" and "$\preccurlyeq$" as having two separate glyphs. In particular, this means that, for example, symbols inside frames, such as "$\boxed{a}$", can be handled correctly.

2.2 Calculating the Feature Vector

A common approach to statistical symbol recognition is to calculate some properties of a previously segmented symbol, e.g., moments of various orders and types, topological or geometric properties such as numbers of holes, line intersections etc., and to produce a feature vector by collecting the values obtained. The resulting feature vector then can be used, for example, in a metric distance or a decision tree based process to find the best match in a database of symbols. In order to improve classification accuracy using detailed features of the symbol, some systems decompose the image into, typically, a 3×3 grid of tiles and base the feature vector on calculations on the individual tiles.

Our method also decomposes the image, but instead of into a uniform grid, it decomposes it recursively into sub-rectangles based on the centres of gravity (first order moments) in horizontal and vertical dimensions, such that the number of positive (i.e., black) pixels is equal on either side of the divide. Furthermore, gross features of the image are represented at higher levels of the decomposition while finer details are still captured in the lower levels.

Figure 1 shows an example how a glyph is decomposed. We name the rectangles produced as $R_{i,j}$ where $i \geq 0$ indicates the level of splitting, and j is the component of a split where $0 \leqslant j \leqslant 2^i - 1$. The first two steps and the final result of the splitting are depicted explicitly. $R_{0,0}$ is the entire glyph image. The splitting is binary so $R_{i,j}$ will split into two sub-rectangles $R_{i+1,2j}$ and $R_{i+1,2j+1}$, where these will be the top and bottom (left and right) parts, respectively, if the splitting was using the vertical (horizontal) component of the centroid of the image. The initial splitting is vertical and each level of splitting then alternates between horizontal and vertical. The component of the centre of gravity used to split $R_{i,j}$ we call $y_{i,j}$ if i is even (and the split is therefore

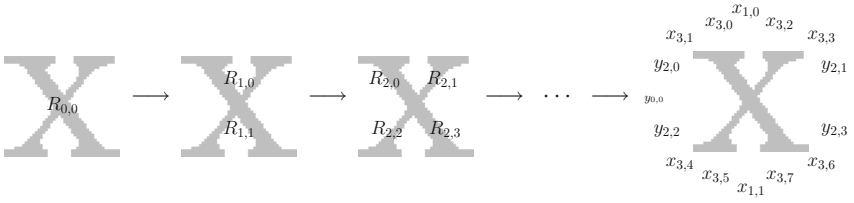


Fig. 1. Decomposition split points of "x" at 11 pt

vertical), otherwise $x_{i,j}$ (and the split is horizontal). Note that the regions at any one level are disjoint but are nested within the level above.

For each rectangular region $R_{i,j}$ of a glyph, we calculate 4 feature vector elements: either the vertical or horizontal component of the centroid, $y_{i,j}$ or $x_{i,j}$, (which is later used for further splitting of this rectangle), scaled to a number between 0 and 1, and the three second order scaled central moments, η_{20}, η_{11} and η_{02} [12] for $R_{i,j}$.

In general, the above elements are scale independent to the limits of discretisation errors. We would like to add an extra element based on the aspect ratio. The raw aspect ratio, however, would dominate the vector for tall and narrow or short and wide glyphs, so we use the hyperbolic tan function on the aspect ratio to smoothly map it into a value between 0 and 1. The element is added at the front of the feature vector.

To see this in practice, compare the final decomposition diagram in Figure 1, with the first line of feature vector elements in Table 1 in Sect. 3. The first feature vector element, fv_0, is the adjusted aspect ratio just described. In this case, the glyph is 45 pixels wide and 39 pixels tall so $\tanh(39/45) = 0.70$ to 2 decimal places.

The next four elements are derived from $R_{0,0}$, i.e., the rectangle described by the outer bounding box of the entire glyph. The vertical centre of gravity is indicated by the $y_{0,0}$ line. This is very slightly above the geometric centre of the image and so is shown, in Table 1, as 0.49 for fv_1 (in line with Java image processing, the origin of the image is the top left corner and the y coordinates increase down the image). fv_2, fv_3 and fv_4 correspond to $\eta_{2,0}, \eta_{1,1}$ and $\eta_{0,2}$ for $R_{0,0}$. The next four elements, $fv_5, \ldots fv_8$, are the corresponding elements for $R_{1,0}$, the top half rectangle of the image. Here fv_5 is marked in the figure as $x_{1,0}$, which, because of the greater thickness of the upper left arm of the glyph, is to the left of the middle of that rectangle with a value of 0.46. The vector elements for the lower half, $R_{1,1}$ follow next. Then comes, in order, the top left, the top right, the bottom left and the bottom right rectangle and so on recursively.

Extraction of the glyph from the document has produced a representation based on lists of line segments, which is suitable for efficient calculation of moments via boundary analysis [8, 2]. Furthermore, calculation of the moments for the sub-rectangles does not require complicated or costly construction of new line segment lists but instead is carried out by simply limiting the values of the line segments to that which would appear in the required rectangle when executing the calculation.

In our current implementation, we are using 4 levels of splitting, resulting in 15 regions, from which we extract feature vector elements and hence the vector currently contains 61 elements. We are experimenting with feature selection techniques to choose a suitable smaller subset of the features for actual metric function evaluation.

2.3 Initial and Alternative Matching Glyphs

Given a database of glyphs with associated precomputed feature vectors, we use the standard euclidean metric for computing glyph similarity. We collect a list of the best matches which can then be filtered to remove obvious impossibilities, apply feedback constraints from higher levels of the document analysis process and impose priorities on choosing between a collection of database glyphs that are all of similar metric distance from the target glyph. The best glyph resulting is returned as the best match but the list is retained so that higher levels of the system could potentially reject the first choice and recalculate the best match based on extra feedback information, e.g. contextual information returned from a semantic analysis.

Obvious impossibilities occur when a possible matching glyph is one component of a multi-glyph symbol but the other components of the same symbol are not found at the corresponding relative location in the target document. Feedback constraints from higher level processing could include, for example, that a particular character appears to be a letter in a word which, by dint of dictionary lookup, is likely to be one of a very small number of letters in a particular font. Priorities that can be applied include giving higher priority to recognising a symbol at a particular font size, over one at a different font size but scaled to match the target glyph.

At this level our approach is purely syntactic. This leaves us with the semantic problem of recognising symbols consisting of disconnected glyphs. While we could leave this to a later stage, in practice we believe this unnecessarily complicates the task of higher level processing. We instead can take advantage of the database information to notice when we have a good match on a component of a symbol and directly search for the remaining components.

We currently use a double track approach: (a) If the glyph matches with a symbol that consists of that one glyph alone we can simply pick it (the result may not be the correct symbol from a semantic point of view but the formatted output should be visually indistinguishable). (b) In the case that the best match for a recognised glyph is a glyph in the database that belongs to a symbol that is composed of multiple glyphs we cannot simply take that symbol since it might introduce glyphs into the result that have no counterpart in the original document. In this case we can consider two possible conflict resolution strategies:

1. We search all closely matching glyphs for one that is the only glyph of its associated symbol.
2. We search all closely matching glyphs for one whose sibling glyphs in its symbol are also matched in the appropriate relative position.

While approach 1 might not necessarily deliver the best matching glyph, it definitely will not introduce superfluous information into the document. But in some cases it will not be possible to find a symbol that matches acceptably well with the original glyph and approach 2 might be preferable (and in general, approach 2 is, of course, more correct from a semantic perspective), which forces a search over sets of glyphs of the particular area under consideration. In our current (first) implementation we give preference to approach 1 by allowing for a small error threshold when matching glyphs and giving a preference to matching single glyph symbols over multi-glyph symbols within that threshold. If this fails, however, we do resort to approach 2.

For example, consider the symbols "$\oplus$" and "$\preccurlyeq$" from Section 2. The former can be composed by two separate symbols, one for "$\bigcap$" and one for the inlaid "$+$". For the latter, however, there is no appropriate match such that we can compose the "$\preccurlyeq$" symbol from two separate, single glyph symbols. While we can find a match for "$\prec$" that is the only glyph of its symbol, the only matches available for the curly upper bar in "$\preccurlyeq$" belong to multi-glyph symbols. Therefore, the algorithm searches for a symbol whose glyphs match as many other glyphs as possible surrounding the curly upper bar in the input image. In the case of our example the algorithm would indeed come up with "$\preccurlyeq$" as the closest matching symbol.

3 The Symbol Database

The templates for the system are held in a database of approximately 5,300 symbols, each in 8 different point sizes, that is augmented with precomputed feature vectors for the character recognition process. We shall briefly describe the design of the database (for a more detailed description on how the database is constructed, see [9]) and explain its main characteristics with an example.

In detail, the database currently consists of a set of *LaTeX formatted documents* (one per point size for 8, 9, 10, 11, 12, 14, 17 and 20 points), which are rendered to tiff format (multi-page, 1 bit/sample, CCITT group 4 compression) at 600dpi. The documents are originally based on [7] and have been extended to cover all mathematical and textual alphabets and characters currently freely available in LaTeX. However, the database can be easily extended for more symbols by adding them to the documents. The documents enumerate all the symbols and homogenise their relative positions and sizes with the help of horizontal and vertical calibrators. The single symbols are then extracted by recognising all the glyphs a symbol consists of as well as their relative position to each other and to the calibrators. We store them in a suitable directory structure with one tiff file per glyph, and no more than 100 symbols per directory, together with an index file containing the requisite extra information such as bounding box to base point offsets, identification of sibling glyphs in a symbol, precomputed feature vector, etc.

In addition to these documents we have an *annotation text file* that is automatically generated from the LaTeX sources during formatting and that contains one line for each symbol described in the LaTeX documents, which associates the identifier of the symbol with the LaTeX code necessary to generate the symbol together with the information on what extra LaTeX packages or fonts, if any, are required to process the code and whether the symbol is available in *math* or *text* mode. Thus we can view each entry in the database as a collection of one or more glyphs together with the indexing information and the code for the actual LaTeX symbol they comprise.

The character recognition extracts single glyphs from the document under analysis and then tries to retrieve the best matches from the database. Since all this works on the level of glyphs only, we take a closer look at the information on single glyphs that is stored in the index files. This information consists essentially of three parts: (1) basic information on the overall symbol, (2) basic information on the glyph, and (3) the precomputed feature vector containing all possible moments described in Section 2.

Information of type (1) and (2) is mainly concerned with bookkeeping and contains elements such as width, height, absolute number of pixels of a symbol or glyph, re-

spectively as well as indexing information. The feature vector (3) is then the actual characterising information for each glyph. It is pre-computed with a uniform length for all glyphs in the database at database creation time. In the current version of the database the vector contains 61 different features. But despite the length of the vector not all features must necessarily be used by the recogniser. In fact the algorithm can be parameterised such that it will select certain features of the vector and restrict the matching algorithm to compute the metric only with respect to the features selected. This facilitates experimenting with combinations of different features and fine-tuning of the character recognition algorithm without expensive rebuilds of the database.

Table 1. Feature vectors for the symbols "x", "x", and "$\times$"

	fv_0	fv_1	fv_2	fv_3	fv_4	fv_5	fv_6	fv_7	fv_8	fv_9	fv_{10}	fv_{11}	fv_{12}	fv_{13}	fv_{14}
x	0.7	0.49	0.18	0.02	0.3	0.46	0.34	0	0.1	0.5	0.37	-0.01	0.11	0.33	0.15
x	0.75	0.49	0.18	0	0.27	0.48	0.35	0	0.1	0.48	0.39	0	0.12	0.44	0.16
$\times$	0.76	0.49	0.32	0	0.32	0.5	0.63	0.01	0.14	0.5	0.63	-0.01	0.14	0.47	0.29

	fv_{15}	fv_{16}	fv_{17}	fv_{18}	fv_{19}	fv_{20}	fv_{21}	fv_{22}	fv_{23}	fv_{24}	fv_{25}	fv_{26}	fv_{27}	fv_{28}	fv_{29}
x	0.1	0.16	0.34	0.22	-0.19	0.26	0.62	0.23	-0.2	0.27	0.6	0.15	0.12	0.18	0.49
x	0.16	0.21	0.44	0.16	-0.16	0.21	0.52	0.17	-0.19	0.24	0.52	0.19	0.2	0.24	0.36
$\times$	0.28	0.29	0.49	0.29	-0.28	0.29	0.48	0.29	-0.28	0.29	0.46	0.29	0.28	0.29	0.25

	fv_{30}	fv_{31}	fv_{32}	fv_{33}	fv_{34}	fv_{35}	fv_{36}	fv_{37}	fv_{38}	fv_{39}	fv_{40}	fv_{41}	fv_{42}	fv_{43}	fv_{44}
x	0.3	0.02	0.03	0.76	0.1	0.08	0.15	0.53	0.21	-0.02	0.04	0.19	0.18	-0.17	0.22
x	0.14	0.07	0.09	0.73	0.11	0.09	0.12	0.57	0.13	-0.07	0.09	0.23	0.11	-0.09	0.12
$\times$	0.17	0.12	0.13	0.7	0.18	0.15	0.16	0.72	0.16	-0.13	0.14	0.26	0.18	-0.14	0.15

	fv_{45}	fv_{46}	fv_{47}	fv_{48}	fv_{49}	fv_{50}	fv_{51}	fv_{52}	fv_{53}	fv_{54}	fv_{55}	fv_{56}	fv_{57}	fv_{58}	fv_{59}	fv_{60}
x	0.75	0.18	-0.18	0.24	0.39	0.21	-0.03	0.05	0.19	0.1	0.08	0.13	0.5	0.22	0.03	0.05
x	0.75	0.11	-0.09	0.14	0.36	0.13	-0.09	0.11	0.22	0.12	0.1	0.13	0.6	0.14	0.09	0.11
$\times$	0.72	0.17	-0.13	0.15	0.27	0.18	-0.14	0.14	0.24	0.17	0.13	0.14	0.7	0.18	0.14	0.16

As an example of feature vectors we compare the symbols "x", "x", and "$\times$", given in the annotation text file as the LaTeX commands `\textrm{\char'170}`, `\textsf{\char'170}`, and `$\times$`, respectively. Each of the symbols only consists of one glyph, whose feature vectors are given in Table 1. Note that the single component values are given with two digit precision only, while in the actual database the numbers are given with full double float precision.

If we now, for instance, consider the first fifteen features, we can observe that there is very little difference between the values for "x" and "x". However, both differ quite considerably from "$\times$" in features $fv_2, fv_6, fv_{10}, fv_{14}$ (i.e., in the $\eta_{2,0}$ moments for the whole glyph, the top and bottom halves of the glyph and the top left quarter of the glyph). This indicates that "$\times$" has essentially the same symmetries as the other two symbols but that the pixels in the corresponding sub-rectangles are more spread out horizontally around the respective centres of gravity for the "$\times$" symbol than for the other two. We can find the first distinguishing features for the two symbols "x" and "x" in the vector component fv_{13}. This is a first order moment corresponding to the centre of gravity $y_{2,0}$ in Figure 1. It basically reflects the impact of the top left hand serif in "x", which pushes the centre of gravity upwards and therefore results in a smaller value than for the other two symbols.

If we now want to compare the three symbols with each other using the entire feature vector, we can compute the following Euclidean distances for the three pairs of symbols:

$$|\mathrm{x} - \mathsf{x}| = \sqrt{\sum_{i=0}^{60} (fv_i(\mathrm{x}) - fv_i(\mathsf{x}))^2} \approx 0.47066 \qquad |\mathrm{x} - \times| \approx 0.83997 \qquad |\mathsf{x} - \times| \approx 0.62672$$

The numbers indicate that "x" is a closer match for "x" than "$\times$". However, the "$\times$" is naturally still closer to the sans serif "x" than to the "x" with serifs. However, none of the three symbols is actually a close match for any of the others, since while the distances between symbols can theoretically range between 0 and $\sqrt{61} \approx 7.81$, overall a distance $\leqslant .15$ is generally considered a close match by the recogniser when using this set of features.

4 An Example Recognition

In order to evaluate the effectiveness of our approach we have essentially two methods to assemble documents: Firstly, we take the closest matching glyph image from the database, possibly apply some scaling to it, and place it at the position in the new document that corresponds to the position of the recognised glyph in the original file. While this has the advantage that we can directly use single glyphs recognised and retrieved from the database and therefore do not have to deal with symbols consisting of several disconnected glyphs, it has the disadvantage that the resulting file is in a picture file format that cannot be used for further processing.

The second method is to assemble an actual LaTeX source file that formats to the recognised text. For this, the glyphs in the original document are identified and an appropriate symbol is chosen from the database. The LaTeX command for that symbol is then put at the correct position in the output document within a LaTeX picture environment whose measurements correspond to the bounding box given by the original document. The restrictions imposed by LaTeX on the `unitlength` of the picture environment can affect the exact placement of characters, since commands can only be put at integer raster points in the environment. Scaling is applied by specifying and selecting a particular font size for the LaTeX command, which also imposes some restrictions with respect to the available font sizes.

We demonstrate the results of our algorithm with an example from a paper, [4], we have experimented with that offers a large number of complex mathematical expressions. The particular expression we are interested in is given in Figure 4 as it appears in the paper. Note that it is in its original form in 10 point size font. As comparison, the results of the OCR algorithm are displayed in Figures 2 and 6, where the former contains the assembled tiff file and the latter the formatted LaTeX generated expression. Since the results are difficult to distinguish with the naked eye, we have combined images 4 and 2 using *exclusive-or rendering* in Figure 3. The similar combination of images 4 and 6 is given in Figure 5. Here, all pixels that show up in only one of the two images appear as black pixels.

The difference in the rendering is more severe for the generated LaTeX expression than for the assembled tiff file. This is due to the fact mentioned earlier, that the characters cannot be placed exactly at the right positions but only approximately at the next

$$[\![A \to B]\!] = S \to [\![A]\!] \to (S \times [\![B]\!])$$
$$\cong \prod_{w' \in \mathcal{W}} (Sw' \to [\![A]\!] \to \sum_{w'' \in \mathcal{W}} (Sw'' \times [\![B]\!]))$$

Fig. 2. Generated mathematical expression image

Fig. 3. Difference between Figure 4 and Figure 2 using XOR rendering

$$[\![A \to B]\!] = S \to [\![A]\!] \to (S \times [\![B]\!])$$
$$\cong \prod_{w' \in \mathcal{W}} (Sw' \to [\![A]\!] \to \sum_{w'' \in \mathcal{W}} (Sw'' \times [\![B]\!]))$$

Fig. 4. Original mathematical expression image

Fig. 5. Difference between Figure 4 and Figure 6 using XOR rendering

$$[\![A \to B]\!] = S \to [\![A]\!] \to (S \times [\![B]\!])$$
$$\cong \prod_{w' \in \mathcal{W}} (Sw' \to [\![A]\!] \to \sum_{w'' \in \mathcal{W}} (Sw'' \times [\![B]\!]))$$

Fig. 6. LaTeX generated mathematical expression

possible integer raster point. Since the algorithm computes the character positions from the centre outwards, in the LaTeX expression the symbols in the middle have the most overlap and the discrepancy increases towards the outside.

But also the generated tiff image does not match the original expression exactly. There are essentially three types of differences which are best explained when looking at the LaTeX code of the original expressions:

```
\newcommand{\seman}[1]{ [\![ {#1} ]\!] }
\begin{eqnarray*}
  \seman{A  \rightarrow B} & = &
  S \rightarrow \seman{A} \rightarrow (S \times \seman{B}) \\
  & \cong &
  \prod_{w'\in {\mathcal{W}}}(Sw' \rightarrow \seman{A} \rightarrow
  \sum_{w''\in {\mathcal{W}}}(Sw'' \times \seman{B}))
\end{eqnarray*}
```

Firstly, we observe that the author of [4] did not use pre-designed symbols for the semantic brackets but rather defined them via a new, handcrafted macro as an overlap of the regular square brackets. Nevertheless, the recogniser finds a suitable replacement for the characters, for instance, the `\textlbrackdbl` command in the **textcomp** package for the left semantic brackets. The distance between the original expression

and the symbol in the database is 0.0689. This discrepancy is essentially caused by the slightly leaner vertical bars in the `\textlbrackdbl` command, which can indeed be observed in the respective feature vectors. While nearly all features differ by a minimal fraction, only, the features fv_{33} and fv_{45}, corresponding to the vertical splits $x_{3,1}$ and $x_{3,4}$, respectively, have a difference of roughly 0.03.

Secondly, the difference in the "$\in$" is caused by the recogniser retrieving the command `\ABXin` from the **mathabx** package in 17 point size, which, however, does not scale as well to the required font size of 9 points as the original "$\in$" symbol would.

Finally, the remaining differences are essentially due to scaling to a point size for which no corresponding characters in the database exist. Our current database does not contain a 7 point set of symbols. However, the "w" and the calligraphic "$\mathcal{W}$" in the subscript of the sum and product symbol are actually in 7 point size. The closest match in the database yields the 9 point versions. These have to be scaled to size 7, which leads to slight discrepancies in the rendering. In the generated LaTeX expression this scaling is achieved, for instance, by LaTeX command `{\fontsize{7}{0}\selectfont $\mathnormal{w}$}`.

5 Conclusion

We have presented a novel algorithm for mathematical OCR that is based on a combination of recursive glyph decomposition and the calculation of geometric moment invariants in each step of the decomposition. The current implementation of the algorithm yields nearly optimal results in recognising documents that are already compiled from actual LaTeX source files. We are currently experimenting with scanned images of documents, in particular, we have started experimenting with articles from the Transactions of the American Mathematical Society [14]. Within the repository of the JSTOR archive [3], images of all the back issues of this journal — starting 1900 — have been made available electronically. While the results of these experiments are already encouraging, more experimentation with feature selection and fine tuning of the recognition algorithm is needed to achieve a robust top quality recognition.

We compared the results from our recogniser with those from two other recognisers we call *Box* and *Grid*. Both use the same aspect ratio feature and the same moment functions as our recogniser, but Box includes the moment functions only on the entire glyph and Grid includes them for each of the 9 tiles of a 3×3 grid subdivision of the glyph. Preliminary results indicate that the Box recogniser performs worst, presumably due to the lack of sensitivity to details of the glyphs. The Grid recogniser suffers from the arbitrary nature of features of empty or near-empty cells in the grid (e.g., the upper right cell of an 'L' character) — a disadvantage that our system is not subject to.

The effective limit on the recursive decomposition of the glyphs to extract feature vectors is the increasing discretisation errors that arise as we try to decompose rectangles with fewer and fewer positive pixels. In practice, for any rectangle, the calculation of the geometric moments will gather some discretisation error, as discussed in [5]. This error can be reduced, at some computation cost, by being more precise in how one translates from the proper continuous integral expression for the moment calculation to the discrete version for a binary image. However, another form of discretisation error

appears as we split into sub-rectangles based on rounding the split point to the nearest pixel boundary. We are currently investigating the costs and benefits of applying a more accurate approach to the discretisation problem here.

The success of our approach depends on the availability of a large high quality database of symbols generated from a large set of freely available LaTeX fonts. In its current version, the database contains, among its approximately 5,300 symbols, 1,600 mathematical symbols and 1,500 characters from different mathematical alphabets. The remaining symbols are mostly regular textual characters, accents, as well as additional scientific symbols, such as chemical or meteorological symbols. Since we keep copies of each symbol at 8 different point sizes, we are currently storing about 42,400 different symbols in total. Since many symbols are composed of more than one glyph, and we actually store glyphs rather than symbols in the database (but with sufficient information to reconstruct the full symbols as needed), we are actually storing about 59,000 glyphs. Nevertheless, the database is easily extensible and is therefore also suitable for recognising scientific texts other than mathematics.

Analysing a document involves extracting the glyphs from the document and finding its nearest neighbours with respect to the metric in the database. The nearest neighbour search is searching in the full database of the 59,000 glyphs for each target glyph in the system. On a moderately powerful desktop PC running Linux, the software, in its current unoptimised state takes about 10 minutes to process a page of three to four thousand target glyphs. Many optimisations are possible to improve this speed but, to provide true scalability to very large databases of symbols, we intend to use an SM-tree [10], a high performance variant of the M-tree metric file access method [1]. However, for our current work, we are using a naïve internal memory algorithm which is slower but adequate for non-production use and easier to experiment with.

References

1. P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proc. of the 23rd VLDB Conference*, p.426–435, 1997.
2. J. Flusser. Fast calculation of geometric moments of binary images. In M. Gengler, editor, *Pattern Recognition and Medical Computer Vision*, p.265–274. ÖCG, 1998.
3. The JSTOR scholarly journal archive. `http://www.jstor.org/`.
4. P. Levy. Possible world semantics for general storage in call-by-value. In J. Bradfield, editor, *Proc. of CSL'02*, volume 2471 of *LNCS*, p.232–246. Springer, 2002.
5. W. Lin and S. Wang. A note on the calculation of moments. *Pattern Recognition Letters*, 15(11):1065–1070, 1994.
6. J. Lladós, E. Valveny, G. Sánchez, and E. Martí. Symbol recognition: Current advances and perspectives. *LNCS* 2390, p.104–127, 2002.
7. S. Parkin. The comprehensive latex symbol list. Technical report, CTAN, 2003. available at `http://www.ctan.org`.
8. W. Philips. A new fast algorithm for moment computation. *Pattern Recognition*, 26(11):1619–1621, 1993.
9. A. Sexton and V. Sorge. A database of glyphs for ocr of mathematical documents. In Michael Kohlhase, editor, *Proc. of MKM-05*, LNCS. Springer Verlag, 2005. In print.
10. A. Sexton and R. Swinbank. Bulk loading the M-tree to enhance query performance. In *Proc. of BNCOD-21*, volume 3112 of *LNCS*, pages 190–202. Springer Verlag, 2004.

11. A. Sexton, A. Todman, and K. Woodward. Font recognition using shape-based quad-tree and kd-tree decomposition. *Proc. of JCIS-2000*, p.212–215. Assoc. for Intel. Machinery, 2000.
12. M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis and machine vision*. International Thomson Publishing, 2nd edition, 1998.
13. M. Suzuki, S. Uchida, and A. Nomura. A ground-truthed mathematical character and symbol image database. In *Proc. of ICDAR-2005*, p.675–679. IEEE Computer Society Press, 2005.
14. Transactions of the American Mathematical Society. Available as part of JSTOR at `http://uk.jstor.org/journals/00029947.html`.
15. D. Trier, A. Jain, and T. Taxt. Feature extraction methods for character recognition - a survey,. *Pattern Recognition*, 29(4):641–662, 1996.

Recognition and Classification of Figures in PDF Documents

Mingyan Shao and Robert P. Futrelle

Northeastern University, Boston, MA 02115, USA
myshao, futrelle@ccs.neu.edu

Abstract. Graphics recognition for raster-based input discovers primitives such as lines, arrowheads, and circles. This paper focuses on graphics recognition of figures in vector-based PDF documents. The first stage consists of extracting the graphic and text primitives corresponding to figures. An interpreter was constructed to translate PDF content into a set of self-contained graphics and text objects (in Java), freed from the intricacies of the PDF file. The second stage consists of discovering simple graphics entities which we call *graphemes*, e.g., a pair of primitive graphic objects satisfying certain geometric constraints. The third stage uses machine learning to classify figures using grapheme statistics as attributes. A boosting-based learner (LogitBoost in the Weka toolkit) was able to achieve 100% classification accuracy in hold-out-one training/testing using 16 grapheme types extracted from 36 figures from BioMed Central journal research papers. The approach can readily be adapted to raster graphics recognition.

Keywords: Graphics Recognition, PDF, Graphemes, Vector Graphics, Machine Learning, Boosting.

1 Introduction

Knowledge mining from documents is advancing on many fronts. These efforts are focused primarily on text. But figures (diagrams and images) often contain important information that cannot reasonably be represented by text. This is especially the case in the Biomedical research literature where figures and figure-related text make up a surprising 50% of a typical paper. The importance of figures is attested to in the leading Open Access Biomedical journal, *PLoS Biology* which furnishes a "Figures view" for each paper.

The focus of this paper is on figures which are diagrams, rather than raster images such as photographs. Our group has worked on diagram-related topics for some years, including work on diagram parsing, spatial data structures, ambiguity, text-diagram interrelations, vectorization of raster forms of diagrams, and summarization. This paper deals with graphic recognition in the large, describing a system that begins with the electronic versions of papers and leads to a classifier trained by machine learning methods that can successfully classify diagrams from the papers. This will then allow knowledge bases to be built for organized browsing and diagram retrieval. Retrieval will normally involve related text and should be able to retrieve diagrams from queries that use diagram examples or system-provided exemplars.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 231–242, 2006.
© Springer-Verlag Berlin Heidelberg 2006

To apply machine learning, we first convert the original electronic format of the diagram into machine-resident objects with specified geometric parameters. Then we design algorithms to generate attribute statistics for each diagram that will successfully characterize a diagram. There are thus three sequential stages in the processing/analysis chain: *Extraction* of the figure-related graphics from papers, *attribute computation*, and *machine learning*.

In online papers in PDF format, diagrams may exist in raster or vector format. Most published diagrams are in raster format. However, BioMed Central (BMC), a leading Open Access publisher, has to date published about 14,000 papers, of which approximately 40% contain vector formatted figures. In the preliminary research reported here, we have used a small number (36) of BMC vector figures. Although this paper focuses on diagrams available in vector format, the approach is equally applicable to raster formats. They would require an additional preprocessing step, vectorization, a sometimes imperfect process for deriving a vector representation [1, 2, 3].

It might seem straightforward to extract graphic objects from PDFs, which are already in vector format. This is not the case. PDF is a page-space, geometry-based language with various graphics/text state assignments and shifts that must be untangled. PDF has no logical structure at any high level, such as explicitly delimited paragraphs, captions, or figures. Even white space in text is not explicitly represented, other than by a position shift before the next character is rendered. A small number of studies have attempted to extract vector information from PDFs, typically deriving an XML representation of the original [4]. For our analysis work, we use in-memory Java objects.

The document understanding community has been focused on text, perhaps overly focused. For example, Dengel [5] in a keynote devoted to "all of the dimensions" of document understanding, doesn't even mention figures. Much of the work on graphic recognition for raster images has been devoted to vectorization of technical drawings and maps [1]. It rarely goes on to extract structure, much less to apply machine learning techniques to the results. One piece of research on chart recognition from raster images, for bar and pie charts, used hand-crafted algorithms for recognition [6].

For vector figures in CAD and PDF, hybrid techniques have been used which rasterize the vector figures and then apply well-developed raster-based document analysis algorithms. This settles the issue of where on the page the various items appear, but the object identity of the items is lost [7, 4]. Our approach is different, because we render (install) the object references in a *spatial index*, a coarse raster-like spatial array of objects [8, 9, 10]. This combines the best of both worlds; it allows us to efficiently discover sets of objects that obey specified spatial constraints.

The spatial indexing approach can operate at the level of full document analysis to locate and separate out graphics on pages irrespective of the placement and order of the vector commands in the underlying files, be they PDF, SVG, or the result of graphics recognition (vectorization).

There is some work on vector-based figure recognition. A system was developed for case-based reasoning that did matching of new diagrams to a CAD database [11]. Graph matching was used in which the graphs had geometrical objects at the nodes and geometric relations on the arcs.

For PDFs, a brief but useful description of PDF file structure can be found in [12]. The Xed system converts PDF to XML [4]. This is one of the few papers we could find that shows the results of extracting geometric state and drawing information from PDF. Such a result would have to be converted back to in-memory objects, as we do, before further analyses could be done. We have no requirement for XML in our work, since Java objects can be serialized to files and visualized using Java 2D. Their paper describes four similar tools, only one of which, the commercial system, *SVG Imprint*, appears to generate geometric output; the other three produce raster output for figures or entire pages only.

2 Graphics Recognition System for PDF

We accomplish graphics recognition for vector figures in PDF in the three stages as shown in Fig 1. The first stage consists of extracting the graphic and text primitives corresponding to figures. The mapping from PDF to the rendered page can be complex, so an interpreter was constructed to translate the PDF content into self-contained graphics and text objects, freed from the intricacies of the PDF file. Our focus is on vector-based figures and their internal text. Heuristics were used in this study to locate the figure components on each page. The target form for the extracted entities is Java objects in memory (or serialized to files). This allows us to elaborate them as necessary and to do the processing for the next two stages.

The second stage consists of discovering simple graphics entities which we call *graphemes*. A grapheme is a small set of graphic primitives satisfying specified geometric constraints [13] and Fig. 4. We can also consider graphemes in a larger sense as including point statistics such as the number of polygons in a figure, or statistical distributions such as a histogram of line lengths. A number of different grapheme types can be defined in order to extract enough information from a diagram to classify it. In certain cases, graphemes may contain many primitives. Examples include a set of tick marks on an axis or a set of small triangles used as data point markers in a data graph. Such large sets are described as obeying *generalized equivalence relations* [8, 14]. Discovering geometrical relations between objects is aided markedly by a preprocessing stage in which primitives are rendered (installed) in a *spatial index* [9, 10].

The third stage uses machine learning to classify figures using grapheme statistics as descriptive attributes. In this paper we report on supervised learning studies. Statistics for 16 different grapheme types were collected for 36 diagrams extracted from BioMed Central papers. The diagrams were manually pre-classified and used for training and hold-out-one evaluation. A boosting algorithm, LogitBoost from the Weka toolkit [15], was used for multi-class learning. LogitBoost was able to achieve 100% classification accuracy in hold-out-one training/testing. Other learning algorithms we tried achieved less than 100% accuracy. We can't expect any machine learning algorithm to achieve 100% accuracy in the scaled up work we will do involving tens of thousands of diagrams. Nevertheless, the preliminary results are encouraging. Using a large collection of atomic elements (graphemes) to characterize complex objects (entire diagrams) is analogous to the "bag of words" approach which has been so successful in text document categorization and retrieval. Once trained, the learning system can classify new diagrams presented to it for which the grapheme statistics have been computed.

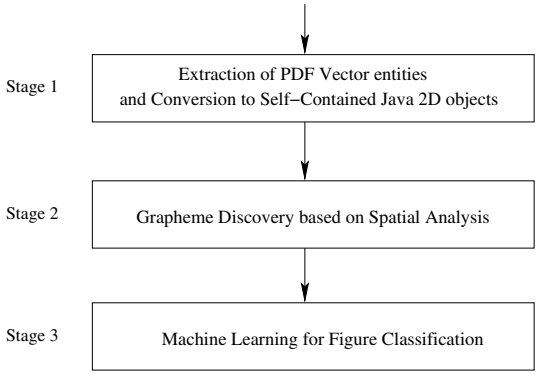


Fig. 1. Stages of our PDF vector figure recognition system. The first stage consists of extraction of the PDF vector entities in the file and their conversion to *self-contained objects*, Java instances compatible with Java 2D. The second stage involves the discovery of simple items in the figure, *graphemes*, a typical one being two or three primitives obeying geometric constraints such as an arrowhead, or a large set of simply related objects such as a set of identically appearing (congruent) data point markers. The third stage is to use the statistics of various graphemes found in a figure as a collection of attributes for machine learning.

Combining extraction, grapheme discovery, and machine learning for diagram classification is a new approach that bodes well for the future.

3 Extraction of Figure-Related PDF Entities

3.1 Features of PDF Documents and Their Graphics

A PDF document is composed of a number of pages and their supporting resources (Fig. 2). Both pages and resources are numbered objects. Each PDF page contains a resource dictionary and at least one content stream. The resource dictionary keeps a list of pairs of a resource object number and a reference name. A resource object may be a font, graphics state, color space, etc. Once defined, resource objects can be referenced anywhere in the PDF file.

The content streams define the appearance of PDF documents. They are the most essential parts of PDF; they use resources to render text and graphics. A content stream consists of a sequence of instructions for text and graphics. Text instructions include text rendering instructions and text state instructions. Text rendering instructions write text on a page. Text state instructions specify how and where text will be rendered to a page, such as location, transform matrix, word space, text rise, size, color, etc.

Graphics instructions include graphics rendering instructions and graphics state instructions. Graphics rendering instructions draw graphics primitives such as line, rectangle, and curve. Graphics state instructions specify the width, color, join style, painting pattern, clipping, transforms, etc. PDF also provides a graphics state stack so that local graphics states can be pushed or popped to change the graphics state temporarily.

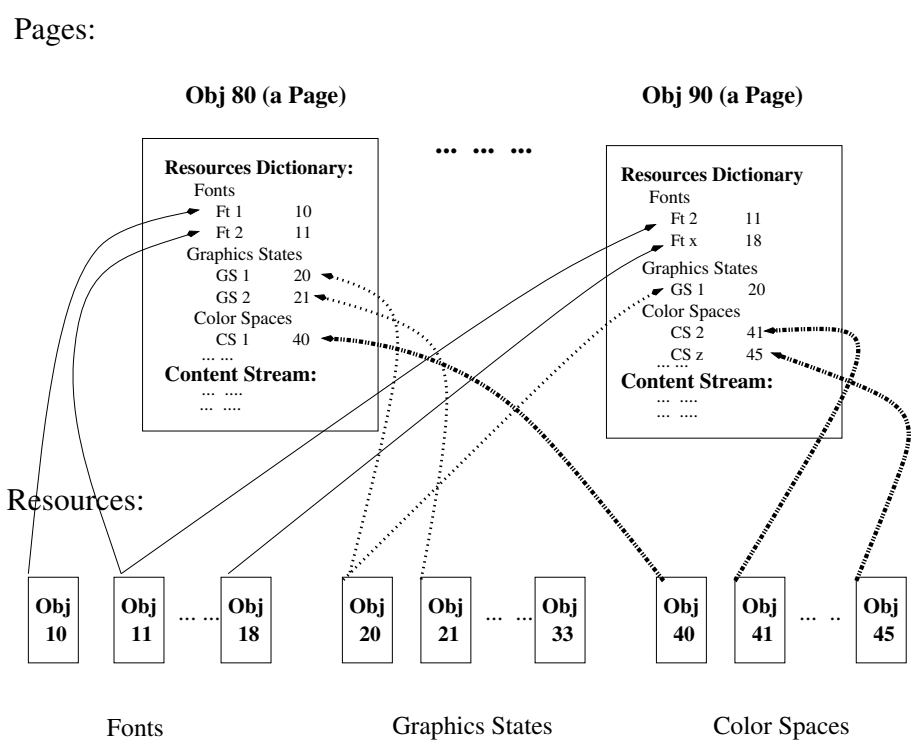


Fig. 2. A simplified PDF structure example. A PDF file is composed of pages and resources such as font, graphics state, and color space. Both pages and resources are defined as objects with a sequence number, a UID. In this example, page 1 is object #80, and font 1 is object #10. These sequence numbers are used as reference numbers when the object is referenced in another object. In this example, object #10 that defines a font is referenced in a page (object #80) object's resource dictionary as "Ft1 10" in which 10 is the font object's sequence number. Once the resource objects are defined, they are globally available, i.e., they can be referenced by any pages in the same PDF file. For instance, object #20 is referenced by two page objects: object #80 and #90. For a useful brief description of PDF structure, see [12].

3.2 Extraction Strategies

To extract graphics, we first translate PDF documents into a format that we can manipulate in software. We apply the open source package, Etymon PJX [16], to translate entire PDF documents into a sequence of Java objects corresponding to PDF objects or instructions. Etymon PJX defines a class for each member of the set of basic PDF commands. It parses the PDF file to create a sequence of object instances, corresponding to the commands in the PDF file, including the argument values given to each command. Thus, for a PDF document, we get Java objects for pages, resources, fonts, graphics states, content streams, etc. Next, we need to determine which Java objects should be extracted. These objects should be the graphics and text inside of figures, as opposed to blocks of text outside the figures proper. The extraction procedure is complicated due

to the structural nature of the PDF content stream, and the lack of a simple mapping between positions in the PDF file content stream and positions on the page.

The PDF content stream is a sequential list of instructions. The sequence is important because the sequence of resources (graphics states and text states) defines the local environment in which the graphics and text are rendered. The values of resources can be changed in the sequence, affecting only the instructions that follow. This property makes extraction complicated because to extract either graphics primitives or text inside graphics with all of their related state parameters, we need to look back through the instruction sequence to find the last values of all the parameters needed.

Despite the fact that the content stream is sequential, the instruction sequence in the content stream is not necessarily in accord with their positions on the page. The content stream instruction sequence and positioning on a page are distinct issues in PDF. A PDF document may apply different strategies to write content streams, all leading to the same appearance, though their instructions may be arranged in different orders. Except in specific cases, we cannot apply content stream position information to aid extraction. The drawing order does affect occlusion when it occurs, but occlusion was not a part of this study.

Extraction of Figures. Since some PDF pages only contain pure text or a few simple figures such as tables, which are not dealt with in this study, we can apply the statistics of line primitives to eliminate such a page — if a page has only a few line primitives, then this page does not contain any figure we need to extract. If there are more than a certain number of non-line primitives such as curves or rectangles, we can conclude that this page must contain one or more figures. If there are neither curves nor rectangles in a page, we can still conclude that a PDF page has figures if the count of line primitives is large enough. A similar strategy was used in our preliminary analyses to determine which papers in the BMC collection contained vector-based figures.

Once we conclude that the graphics in a PDF page contains figure material, we extract both graphics rendering instructions and their supporting graphics states. Graphics states can be specified in either the content stream or in separate objects. Graphics state instructions in content streams can be easily extracted as normal instructions, while graphics states in separate objects are extracted using reference and resource dictionaries.

Extraction of Text Within Figures. After extracting all of the graphics elements in a figure, the text inside the figures needs to be extracted. As explained in Section 3.1, the sequence of text and graphics instructions is not necessarily in accord with the sequence in the rendered page. This makes it difficult to decide which part of text instructions in content stream renders the text inside of graphics. The PDF articles published by BioMed Central (BMC) use an Adobe FrameMaker template that results in the PDF content stream structure shown in Fig. 3.

Once the figures and their text have been extracted, we can create PDFs for viewing and validation. This is done by using Etymon PJX tools to generate PDF from the extracted subset of Java objects. This PDF should contain the figures and their text, nothing more nor less.

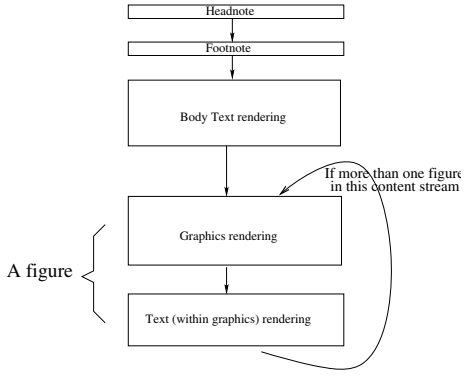


Fig. 3. The content stream structure of BioMed Central (BMC) PDF pages. The content stream of all the BMC PDF pages is organized in the following sequence: head-note, footnote, body-text, graphics instructions (including rendering instructions, graphics state instructions, graphics state references), and text inside the graphics. Graphics, if any, are rendered at the end of each content stream, and text inside the graphics follows the graphics rendering instructions. This structure help us to locate and extract the text inside graphics. The use of this BMC-specific structure is in no way a limitation on our methods. Spatial indexing can be used to locate the objects forming the graphics content in a page irrespective of the content stream sequence.

3.3 An Interpreter to Create Self-contained Objects

The results of the extraction step are Java objects of graphics/text drawing instructions, graphics/text states, etc. This sequence of Java objects exactly mirrors the PDF instructions in the content stream. PDF rendering instructions usually depend on the local environment defined by state instructions. Thus, rendering a graphics object requires the current graphics state and rendering text requires the current font definition. In principle, the entire preceding content stream must be read to get the state parameters needed to render graphics or text.

We have implemented an interpreter to translate these interdependent Java objects into *self-contained objects*. Each self-contained object, either a graphics primitive or text, contains a reference to a state object describing its properties. To enhance modularity, multiple self-contained objects may reference the same state object.

In PDF, the graphic state stack is used to temporarily save the local graphics state so that it will not affect the environment that follows. We deal with this problem by implementing a stack in our interpreter to simulate the PDF state stack so that the local graphics state and the pushed prior state(s) are preserved. Then every self-contained object, no matter how its graphics state is defined, by internal graphics state instructions, external graphics state objects, or via the graphics state stack, references the correct state.

Our interpreter reads all extracted objects and translates and integrates them into self-contained objects that extend Java 2D classes so that they can be manipulated independently from the PDF specification.

4 Spatial Analysis and Graphemes

Up to this point, we have described the extraction of graphics primitives. The ultimate utility of the extracted primitives is for the discovery of the complex shapes and constructions that they comprise, and beyond that to use them in systems that index and retrieve figures and present them to users in interactive applications. A thorough analysis of a figure can involve visual parsing, for example to discover the entire structure of an x,y data graph with its scale lines and annotations as well as data points and data lines, and so forth [17, 9]. Here we describe an alternate approach based on *graphemes*, which is simple compared to full parsing, but still quite useful. A grapheme is typically made up of only two primitives; examples are shown in Fig. 4.

Graphemes allow us to classify figures using a variety of machine learning techniques, as we will see in Section 5. Classification, in turn, can enable indexing and retrieval systems to be built.

A particular grapheme class is described as a tuple of primitives, usually just a pair, that obey constraints on the individual primitives as well as geometrical constraints that must hold among them. For example the *Vertical Tick* tuple in Fig. 4 can be described as a pair of lines, L_1 and L_2 that obey the constraints described in Algorithm 4.1.

Algorithm 4.1. VERTICAL_TICK(L_1, L_2)

Comment: Check if a pair of lines (L_1, L_2) construct a Vertical_Tick

if $\begin{cases} short(L_1); \\ vertical(L_1); \\ long(L_2); \\ horizontal(L_2); \\ below(L_1, L_2); \\ touch(L_1, L_2); \end{cases}$

then $Vertical\_Tick \leftarrow L_1, L_2$

Comment: If L_1 is a short vertical line and L_2 a long horizontal line, L_1 is below L_2, and they touch at one end of L_1, then they form a Vertical_Tick.

Graphemes such as *Vertical_Tick* can be discovered by simplified versions of the Diagram Understanding System developed earlier by one of us [9, 18]. One difficult aspect of such analyses is exemplified by the predicates *short()* and *long()* in Algorithm 4.1. This is dealt with by a collection of strategies, e.g., line length histogram analyses, or comparing lengths to the size of the smallest text characters for *short()*.

4.1 Spatial Indexes Aid Grapheme Parsing

The parsing algorithms that define graphemes operate efficiently because a preprocessing step is used to install the primitives in a *spatial index*, allowing constraints such as *below()* and *touch()* to be evaluated rapidly.

A spatial index is a coarse 2D-array of cells (array elements) isomorphic to the 2D metric space of a figure [8, 9, 10, 18] . Each graphics primitive is rendered into the spatial index so that every cell contains references to all graphics primitives that occupy or pass

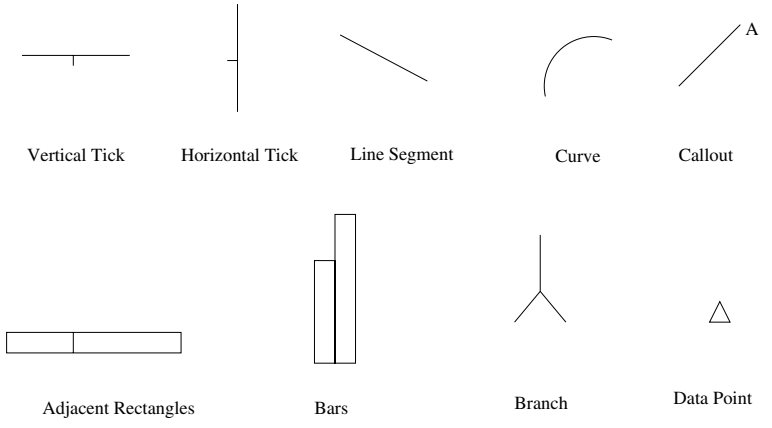


Fig. 4. Some grapheme examples: Vertical Tick, Horizontal Tick, Line, Curve, Callout, Adjacent Rectangles, Bars, Branches, and Data Point

through the cell. Each primitive contains its position in the original PDF drawing sequence in order to faithfully represent occlusions that can occur accidentally or by design.

The spatial index provides a efficient way to deal with spatial relations among graphics primitives, and enables us to deal with various graphics objects such as lines, curves, and text in a single uniform representation. For example, the *touch()* predicate for two primitives simply checks to see if the intersection of the two sets of cells occupied by the primitives is non-empty.

5 Machine Learning for Graphics Classification and Recognition

We analyzed vector graphics figures in PDF articles published by BMC, and defined the following five classes as shown in Fig. 5.

- A *data point figure* is an x, y data graph showing only data points;
- A *line figure* is an x, y data graph with data lines (may also have data points);
- A *bar chart* is an x, y data graph with a number of bars of the same width;
- A *curve figure* is an x, y data graph with only curves;
- A *tree* is a hierarchical structure made of some simple graphics such as rectangles or circles that are connected by arrows or branches.

5.1 Results: Machine Learning of Diagram Classes Using Graphemes

To the extent that distinct classes of figures have different grapheme statistics (counts for each grapheme type), we can use machine learning techniques to distinguish figure classes. We have used supervised learning to divide a collection of figures into the five classes described in Fig. 5.

We extracted figures from PDF versions of articles published by BioMed Central. We examined 7,000 BMC PDFs and found that about 40% of them contain vector graphics.

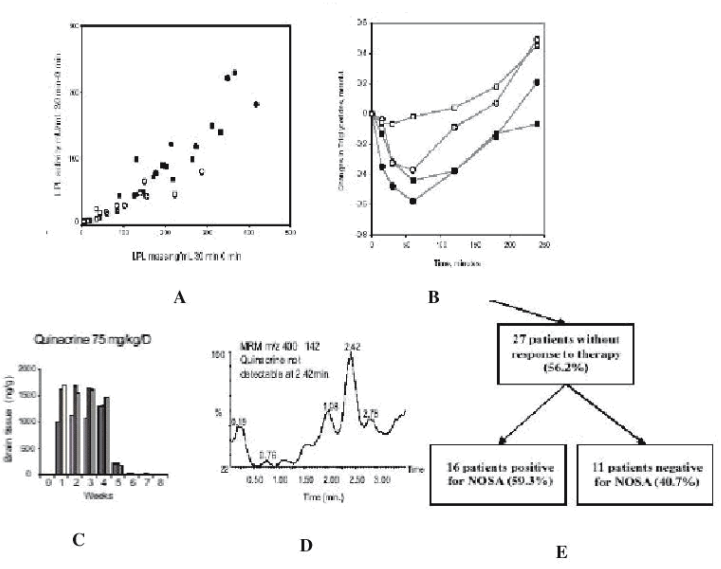


Fig. 5. Five figure classes: **A**: Data graph - points. **B**: Data graph - lines. **C**: Bar chart. **D**: Data graph - curves. **E**: Tree/Hierarchy. The figures used were drawn from BMC papers.

This high percentage, compared to other publishers, appears to be because of specific guidance and encouragement in the BioMed Central instructions for authors.

We extracted vector data from 36 diagrams. A total of 16 different grapheme classes were used as attributes, all geometrical in nature. The counts of grapheme instances in particular diagrams varied from 0 to 120, the latter value being the number of data points in one of the data graph diagrams. Two multi-class learners in the Weka 3, Java-based workbench were used, the Multilayer Perceptron, and LogitBoost. In hold-out-one testing, the perceptron was 94.2% accurate. Its failure on some cases was not unexpected. LogitBoost is a member of the new class of boosting algorithms in machine learning and was able to achieve 100% accuracy on this particular set of 36 diagrams. This excellent result is a testament both to the power of graphemes as indicators of diagram classes and to the power of modern boosting methods. In the future, we will extend these results by analyzing much larger collections of diagrams. As the size and complexity of these collections increases, the accuracy will most certainly be below 100%.

6 Conclusions

This paper has described the design, implementation, and results for a system that can extract and analyze figures from PDF documents, and classify them using machine learning. The system, made up of three analysis stages, was applied to the content of diagrams from research articles published by BioMed Central.

Stage 1. *Extraction* of the subset of PDF objects and commands that comprise vector-based figures in PDF documents. The process required building an interpreter that led to a sequence of self-contained Java 2D graphic objects mirroring the PDF content stream.

Stage 2. *Graphemes* were discovered by analysis of the objects extracted in Stage 1. Graphemes are defined as simple subsets of the graphic objects, typically pairs, with constraints on element properties and geometric relations among them.

Stage 3. *Attribute vectors* for multi-class learners were generated using statistics of grapheme counts for 16 grapheme classes for 36 diagrams, divided into five classes. The best of these learners, LogitBoost from the Weka 3 workbench, was able to achieve 100% accuracy in hold-out one tests.

6.1 Future Work

Besides purely geometrical graphemes, it will be useful to create attributes based on various statistical measures in the figures such as histograms of line lengths, orientations, and widths, as well as statistics on font sizes and styles.

We will include additional classes of vector-based PDF papers that are not created with the standardized FrameMaker-based structure that BMC papers have. The spatial indexing techniques we have described will allow us to locate the figures and figure-related text in such papers irrespective of their position in the PDF content stream sequence.

The approach described here has focused on vector-based diagrams. The great majority of figures published in electronic form are raster based, typically JPEGs. Vectorization of these figures [1, 3, 2, 19], even if imperfect, can generate a vector-based representation of the figure that will allow graphemes to be generated. This in turn will allow systems to be built that can take advantage of figure classification. Such systems could, in principle, deal with all published figures, though most successfully when operating on line-drawn schematic figures, that is, diagrams.

Grapheme-based approaches can form a robust foundation for building full-fledged knowledge-based systems that allow intelligent retrieval of figures based on their information content. In practice, indexing and retrieval of figures will be aided by including figure-related text as a component. We intend to use graphemes as one component of the new diagram parsing system we are developing, which will substantially extend the capabilities of our earlier systems [9, 18]. The fully parsed diagrams that result will allow the construction of more fine-grained knowledge-based systems. These will allow user-level applications to be built that include interactions with diagram internals, linkage between text descriptions and diagram content, and more.

This paper extends our earlier results [10] that also used spatial indexing and machine learning techniques to classify vector-based diagrams. Our papers on a variety of aspects of diagram understanding can be found at http://www.ccs.neu.edu/home/futrelle/papers/diagrams/TwelveDiagramPapersFutrelle1205.html

References

1. Ablameyko, S., Pridmore, T.: Machine interpretation of line drawing images : technical drawings, maps, and diagrams. Springer (2000)
2. Tombre, K., Tabbone, S.: Vectorization in graphics recognition: To thin or not to thin. In: Proceedings of 15th International Conference on Pattern Recognition. Volume 2. (2000) 91–96

3. Lladós, J., Kwon, Y.B., eds.: Graphics Recognition, Recent Advances and Perspectives, 5th InternationalWorkshop, GREC 2003, Barcelona, Spain, July 30-31, 2003, Revised Selected Papers. In Lladós, J., Kwon, Y.B., eds.: GREC. Volume 3088 of Lecture Notes in Computer Science., Springer (2004)
4. Hadjar, K., Rigamonti, M., Lalanne, D., Ingold, R.: Xed: A new tool for extracting hidden structures from electronic documents. In: First International Workshop on Document Image Analysis for Libraries (DIAL'04). (2004) 212–224
5. Dengel, A.: Making documents work: Challenges of document understanding. In: Proceedings ICDAR'03, 7nd Int'l Conference on Document Analysis and Recognition, Edinburgh, Scotland (2003) 1026–1035 Key Note Paper.
6. Huang, W., Tan, C.L., Leow, W.K.: Model-based chart image recognition. In: GREC'03. (2003) 87–99
7. Chao, H., Fan, J.: Layout and content extraction for PDF documents. In: Document Analysis Systems (DAS). (2004) 213–224
8. Futrelle, R.P.: Strategies for diagram understanding: Object/spatial data structures, animate vision, and generalized equivalence. In: 10th ICPR, IEEE Press. (1990) 403–408
9. Futrelle, R.P., Nikolakis, N.: Efficient analysis of complex diagrams using constraint-based parsing. In: ICDAR'95. (1995) 782–790
10. Futrelle, R.P., Shao, M., Cieslik, C., Grimes, A.E.: Extraction, layout analysis and classification of diagrams in PDF documents. In: ICDAR'03. (2003) 1007–1014
11. Luo, Y., Liu, W.: Interactive recognition of graphic objects in engineering drawings. In: GREC'03. (2003) 128–141
12. Hardy, M., Brailsford, D., Thomas, P.: Creating structured PDF files using xml templates. In: In Proceedings of the ACM Symposium on Document Engineering (DocEng'04), Milwaukee, USA, ACM Press (2004) 99–108
13. Futrelle, R.P.: Ambiguity in visual language theory and its role in diagram parsing. In: VL'99. (1999) 172–175
14. Futrelle, R.P., Kakadiaris, I.A., Alexander, J., Carriero, C.M., Nikolakis, N., Futrelle, J.M.: Understanding diagrams in technical documents. IEEE Computer **25** (1992) 75–78
15. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. 2nd edn. Morgan Kaufmann,San Francisco (2005)
16. Etymon: (Pjx 1.2) http://www.etymon.com/epub.html.
17. Chok, S.S., Marriott, K.: Automatic generation of intelligent diagram editors. ACM Trans. Comput.-Hum. Interact. **10** (2003) 244–276
18. Futrelle, R.P.: (http://www.ccs.neu.edu/home/futrelle/diagrams/demo-10-98/) The Diagram Understanding System Demonstration Site.
19. Shao, M., Futrelle, R.P.: Moment-based object models for vectorizaiton. In: IAPR Conference on Machine Vision Applications (MVA2005). (2005) 471–475

An Incremental Parser to Recognize Diagram Symbols and Gestures Represented by Adjacency Grammars*

Joan Mas, Gemma Sanchez, and Josep Llados

Computer Vision Center, Computer Science Department.
Edifici O, Universitat Autònoma de Barcelona,
08193 Bellaterra (Barcelona), Spain
{jmas, gemma, josep}@cvc.uab.es
http://www.cvc.uab.es

Abstract. Syntactic approaches on structural symbol recognition are characterized by defining symbols using a grammar. Following the grammar productions a parser is constructed to recognize symbols: given an input, the parser detects whether it belongs to the language generated by the grammar, recognizing the symbol, or not. In this paper, we describe a parsing methodology to recognize a set of symbols represented by an adjacency grammar. An adjacency grammar is a grammar that describes a symbol in terms of the primitives that form it and the relations among these primitives. These relations are called *constraints*, which are validated using a defined cost function. The cost function approximates the distortion degree associated to the constraint. When a symbol has been recognized the cost associated to the symbol is like a similarity value. The evaluation of the method has been realized from a qualitative point of view, asking some users to draw some sketches. From a quantitative point of view a benchmarking database of sketched symbols has been used.

Keywords: Adjacency grammar, incremental parser, syntactical symbol recognition.

1 Introduction

Symbol Recognition is an active research discipline of Graphics Recognition. Symbol Recognition engines are at the heart of different Graphics Recognition workflows. They are present in processes like re-mastering engineering drawings, indexing in document databases, or freehand sketching. Symbols are heterogeneous entities. They range from simple 2D line-based structures to complex objects consisting of text and graphic elements with particular color, texture and shape attributes.

The problem of symbol recognition may be stated in terms of different approaches. In this paper we focus on two of them. Depending on the methodological basis, it can be solved by statistical or structural approaches. On the other hand, depending on the input mode used to draw symbols, we can distinguish between on-line or off-line modes.

* This work has been partially supported by the spanish project CICYT TIC2003-09291 and catalan project CeRTAP PVPC.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 243–254, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Firstly, as other pattern recognition problems, two major methodological trends can be used to recognize a symbol: statistical and structural. Graph-based approaches are generally used when symbols consist of line structures, while grammars are suitable to recognize symbols or combinations of symbols arranged according to a kind of diagrammatic notation or language like music [1], dimensions [2] or logic diagrams [3]. On the other hand, statistical-based methods seem more useful when symbols are small entities [4] or have heterogeneous components like logos [5].

The problem of symbol recognition can also be classified according to the mode used to draw the symbols (on-line or off-line). On-line recognition involves a kind of digital pen interface. Systems using pen devices as input can take advantage of dynamic information as speed, pressure or order of strokes. A number of illustrative on-line graphics recognition applications can be found in the literature. Most of them consist in sketching interfaces that involve a recognition of symbols of a diagrammatic language. SILK [6] is a system to design a GUI and it recognizes the widgets and other interface elements as the designer draws them. Jorge et al. have used different types of grammars as adjacency grammars [7] or fuzzy relational grammars [8] to recognize sketched shapes. A particular application where sketched symbols are used to index in a database is *SmartSketch Pad* [9] that has been designed for PDAs.

In this paper we deal with the problem of symbol recognition applied to a sketching framework. This framework has two working modes, one allowing the design of sketches from the beginning and the other modifying an existing sketch. The former involves either an off-line or an on-line input mode depending on whether the sketch is scanned from a paper or it is drawn by a digital pen device. The latter involves an on-line input mode. Since either on-line and off-line input modes are considered, we have designed a general recognition approach able to tackle with this duality.

From the point of view of the methodology we use a structural approach to solve the problem. This methodology is selected since we use the structure of the patterns as a characteristic, taking relations between the subpatterns, components or elements, instead of numerical-valued features as in the case of statistical methodology. More specifically we use a syntactical approach. This approach is based on the formal language theory.

Symbol recognition in on-line mode requires to tackle with three difficulties. First, the invariance to the order of the strokes. Different users may draw the strokes of the same symbol in a different order. Since our framework is not restricted to a particular user we do not fix any order in the way they draw the symbol. On the other hand, an on-line recognition system can not know when a symbol has been completely drawn. Our method recognizes symbols in an incremental way analyzing stroke by stroke as long as they are introduced. Finally, a hand-written input means a distortion in it. In summary, given such a framework the proposed solution solves the problems of: any order in the sequence of the input strokes, incremental input of these strokes and distortions on them.

To solve the problem of different order in the sequence of the input, our syntactic approach considers the input as a set instead of a sequence. The solution proposed to solve that problem is an adjacency grammar. The rules defining such a grammar describe symbols as a set of primitives and the relations among them. The grammar has

also to cope with distortions in these relations. To allow these distortions, a degree in the relation among primitives is computed. To conclude, given a sequence of strokes and a grammar defining a symbol, a parser is required to accept the sequence as a predefined symbol or gesture or to reject it. Since the parser needs to recognize symbols while entering strokes, an incremental parser is used.

The rest of this paper is organized as follows. First in section 2 adjacency grammars are introduced, and the instance of the grammar used in this work is presented. In section 3 the parsing process used to recognize sketched symbols is described. Section 4 is devoted to results. Finally conclusions and discussion are presented in section 5.

2 An Adjacency Grammar to Represent Symbols and Gestures

An adjacency grammar is used to define a symbol by a set of primitives and the adjacency constraints among them. This grammar allows the specification of two dimensional structures with one dimensional string. Following the approach presented in [7], formally an adjacency grammar is defined by a 5-tuple $G = \{V_t, V_n, S, C, P\}$ where:

- $\mathbf{V_t}$**:** denotes the alphabet of terminal symbols. In this work, $V_t = \{$*segment*, *arc*$\}$, *arc* refers to open or closed arcs.
- $\mathbf{V_n}$**:** denotes the alphabet of non-terminal symbols.
- **S** $\epsilon \mathbf{V_n}$**:** is the start symbol of the grammar.
- **C:** is the set of constraints applied to the elements of the grammar. In our case the set of constraints are presented in Fig. 1 enumerated as: C = {incident, adjacent, intersects, parallel, perpendicular, contains}.
- **P:** are the productions of the grammar defined as:

$$\alpha \rightarrow \{\beta_1, \ldots, \beta_j\} \; if \; \Gamma(\beta_1, \ldots, \beta_j) \tag{1}$$

Where $\alpha \epsilon V_n$ and all the $\beta_j \epsilon \{V_t \cup V_n\}$, constitute the possibly empty multiset of terminal and non-terminal symbols. Γ is an adjacency constraint defined on the attributes of the β_j. In our case the attributes are $\{X_0, Y_0, X_f, Y_f, X_c, Y_c, R, Bound\}$ where:

- $\mathbf{X_0}, \mathbf{Y_0}$**:** denotes the starting point of a segment.
- $\mathbf{X_f}, \mathbf{Y_f}$**:** denotes the end point of a segment.
- $\mathbf{X_c}, \mathbf{Y_c}$**:** denotes the center of an arc.
- **R:** denotes the radius of an arc.
- **Bound:** denotes the bounding-box of the symbol.

Notice that since the recognition is order free, it is not relevant which terminal point of the stroke is attributed as starting or end point.

A valid example of a grammar production is $\alpha \rightarrow \{\beta_1, \beta_2, \beta_3\} \epsilon P$, where we should consider all 6 possible permutations of β_1, β_2 and β_3 as equally valid substitution for α. Since the order in which the primitives appear is not taken into account, we can use this grammar on both on-line and off-line input modes.

To illustrate the grammatical formalism, look at the example of Fig. 2. Look for example at the production *SYMBOL5*, the set of primitives that describes this symbol is formed by a non-terminal symbol of type *QUAD*, and two terminal symbols of type

(a) Incident (b) Adjacent (c) Intersects (d) Perpendicular (e) Parallel (f) Contains

Fig. 1. Examples of constraints

segment. The relations defined among these primitives are the intersection of the two tokens of type *segment* and the inclusion of these tokens in the token represented by the non-terminal symbol *QUAD*. The grammatical rule that describes a symbol of type *QUAD* is also shown in Fig. 2 and represents a square. Since it does not exist a constraint describing the length ratio between two primitives, rectangles may be also considered as the non-terminal *QUAD*.

Further attention should be made to the definition of constraints shown in Fig. 1. Notice, according to the rules of Fig. 2, that constraints are formulated between either two primitives or a non-terminal and a primitive. Production named as *SYMBOL3* is an example of that. Constraint $Parallel(Segment1, QUAD, \&cost)$ involves that the *Segment1* is *parallel* to some of the components of *QUAD*. On the other hand, constraint $Constains(QUAD, Segment1, \&cost)$ implies that *Segment1* is contained on the bounding-box of the non-terminal *QUAD*. We can also have a constraint $Contains$ defined as $Contains(a, b, c, \&cost)$ that takes into account if the symbol or primitive c is included into the bounding-box generated by the union of the bounding-boxes of a and b. Production called as *SYMBOL51* of Fig. 2 has a constraint $Contains$ defined such that.

This syntactic approach is used for symbol recognition in a sketching interface framework. Since sketches have an inherent inaccuracy and distortion, they do not contain perfect instances of symbols. A sketch is just a rough expression of an idea. For this reason a distortion model has to be considered and added to the grammar. Before modelling distortion tolerance, we have studied the types of distortions that may appear in hand-drawn symbols. It has resulted in classifying them in three classes:

- *Distortions on segments:* Related to primitives labelled as *segments* and affects restrictions on the grammar as *parallel* and *perpendicular*. Figure 3(b) shows some examples of hand-drawn instances where this distortion is accomplished.
- *Distortions on junctions:* Refers to the connection or junctions among the primitives. Affecting both of the kind of primitives and directly related to constraints as *incident* and *adjacent*. In Fig. 3(c) appears some instances that have this distortion.
- *Misclassification of primitives:* This group of distortions includes the addition of extra primitives and the incorrect identification of a primitive. Figure 3(d) shows some examples on this distortion. Instances having a distortion of this kind are not suitable of being well recognized by our system.

The syntactic approach presented in this paper copes with the distortions of the first two groups. To solve the problem of the distortion tolerance we have extended the grammar with the addition of cost functions to the constraints. This cost measures the degree under which a constraint is accomplished. The cost is normalized between 0 and 1. For example, referring to the constraint *Parallel*, this constraint would take a

```
QUAD := {Segment1,Segment2,Segment3,Segment4}
        Adjacent(Segment1,Segment2,&cost) & Adjacent(Segment2,Segment3,&cost)
        & Adjacent(Segment3,Segment4,&cost) & Adjacent(Segment4,Segment1,&cost)
        & Parallel(Segment1,Segment3,&cost) & Parallel(Segment2,Segment4,&cost) &
        Perpendicullar(Segment1,Segment2,&cost) & Perpendicullar(Segment2,Segment3,&cost)
        & Perpendicullar(Segment3,Segment4,&cost) & Perpendicullar(Segment4,Segment1,&cost)

TRIANGLE := {Segment1,Segment2,Segment3}
            Adjacent(Segment1,Segment2,&cost) & Adjacent(Segment2,Segment3,&cost)
            & Adjacent(Segment3,Segment1,&cost).

SYMBOL 3 := {QUAD,Segment1}
            Contains(QUAD,Segment1,&cost) & Parallel(Segment1,QUAD,&cost).

SYMBOL 5 := {QUAD,Segment1,Segment2}
            Contains(QUAD,Segment1,&cost) & Contains(QUAD,Segment2,&cost)
            & Intersects(Segment1,Segment2,&cost).

SYMBOL 33 := {QUAD,Arc}
             Contains(QUAD,Arc,&cost) & Closed(Arc,&cost).

SYMBOL 46 := {TRIANGLE,Segment}
             Contains(TRIANGLE,Segment,&cost).

SYMBOL 47 := {SYMBOL46,Arc}
             Contains(Arc,SYMBOL46,&cost) & Closed(Arc,&cost).

SYMBOL 51 := {Arc,Segment1,Segment2}
             Intersects(Segment1,Segment2,&cost) & Contains(Segment1,Segment2,Arc,&cost).
```

Fig. 2. Example rules of the adjacency grammar corresponding to symbols represented in Fig. 4

cost of 1 where the two strokes that are evaluated, form an angle of 90 degrees. Once an input has been evaluated, it has a cost calculated from the costs obtained by all the constraints referring the symbol represented. This cost measures how similar is the input to the definition made by the grammatical rule. Depending on wether constraint has been evaluated the cost is calculated depending on the angle between the primitives, refers to constraints *parallel* and *perpendicular*, or the distance between points of the primitives, involves constraints *adjacent*, *incident* and *intersects*. The final cost for any symbol is calculated as follows:

$$Cost = \frac{\sum C_i}{\#Constraints} \tag{2}$$

where C_i is the cost associated to any constraint.

3 Incremental Parser

Once the grammar model has been presented, let us describe the recognition process. It consists in a parsing approach that process the input, i.e. hand drawn strokes, guided by the grammar rules.

Among all the parser paradigms, we have chosen an incremental parser. This parser is a bottom-up parser which implies that the recognition of the symbol starts from the primitives and goes up to the start symbol. The main reason to choose this parser is

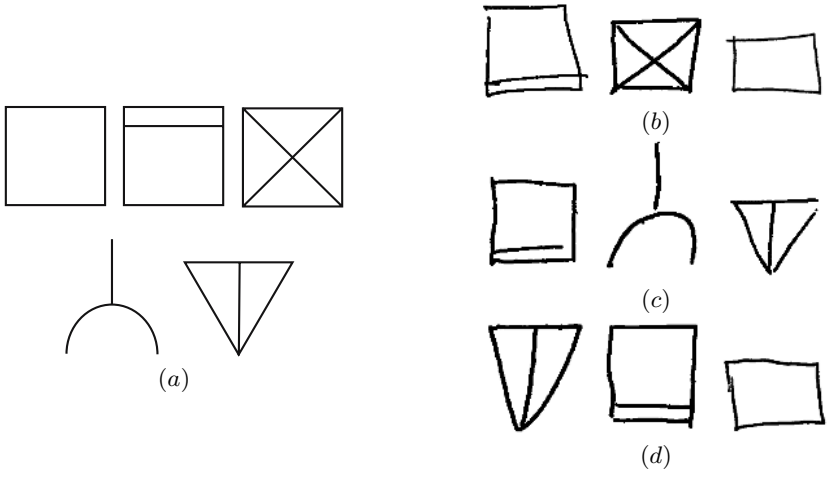


Fig. 3. Distortions occurred in online symbols, (a) Ideal instances (b) Distortions on segments, (c) Distortions on junctions and (d) Misclassification of primitives

that it analyzes the input in an incremental way, i.e. it works as long as the primitives are being entered to the system, without requiring the drawing to be completed to start the recognition. Other authors considering incremental parsers are Chok in [10], and Costagliola in [11].

The parser presented in this work is characterized by two requirements: First, we have to divide the symbols that we are going to recognize into disjoint classes where two symbols that belong to the same group can not be one a subpart of the other. If we look at the rules described in Fig. 2 we can observe that we should create three groups of symbols. The first group should contain the symbols {*QUAD, TRIANGLE, SYMBOL51*}. The second group would be formed by the symbols {*SYMBOL3, SYMBOL5, SYMBOL33, SYMBOL46*}. The last group is the one formed by symbol {*SYMBOL47*}. In Fig. 4, we can observe that *SYMBOL3* contains the symbol *QUAD* as a subpart, that makes these two symbols appear in different groups. In the same case, we encounter *SYMBOL46* and *TRIANGLE*. On the contrary, if we compare *SYMBOL3* and *SYMBOL5*, we can observe that these two symbols share a subpart but none of both is a subpart of the other. For this reason these symbols are in the same group. These groups are specified by a human actor.

The second requirement is the definition of a list. This list is formed by pairs as:

$$L = \{(S_1, P_1), \ldots, (S_n, P_n)\} \tag{3}$$

where S_i refers to a symbol and P_i refers to a primitive.

Each pair defines an incompatibility between a symbol and a primitive. An incompatibility such as we define in this parser is just the set of primitives that are incompatible with the reduction of a symbol. For example, if we look at the productions of Fig. 2, we can observe that the symbol represented as *QUAD*, should not have an incompatibility with any kind of primitive. On the other hand, there should be an incompatibility

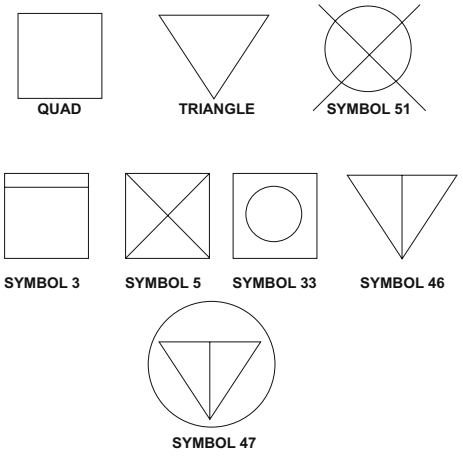


Fig. 4. Set of symbols defined by the adjacency grammar

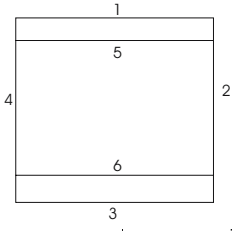


Fig. 5. Input sequence of the parser

between the symbol represented as *SYMBOL3* and a primitive labelled as *segment*, this incompatibility avoids ambiguity in the parser decisions. When the parser has reduced a symbol and the next primitive that enters to the system provokes an incompatibility the parser has to go back to a previous step and add the new primitive to the top. For example, if we define an incompatibility between *SYMBOL3*, see Fig. 4, and a primitive of type *segment*. If the parser analyzes an input sequence like the shown in Fig. 5, the parser would found an incompatibility in the list due to the sixth primitive, and would have to go back to a previous step.

Once we have defined the requirements of the parser, we are going to explain how it works. The parser takes one primitive and tries to apply some of the rules of the first group. If there is not a valid rule the parser takes another primitive and tries to apply the rules again. In the case that a valid reduction is found, the parser inserts the left-hand of the production on the parser tree and takes another primitive. The process of reduction is preceded by the calculation of the cost value of any candidate rule. In the case of multiple candidate rules we will select the rule that minimizes this cost. Before trying to apply a rule of the next level, the parser checks an incompatibility among the items recognized and the new primitive. If the incompatibility is found, the parser has to take away the symbol recognized from the parser tree, and try to recognize it again.

The process continues until all the primitives are read. Each symbol is returned when the parser recognizes it.

One of the main drawbacks of this parser process is the complexity. Looking at the following pseudo-code we can assure that the parser has a complexity of order $\bigcirc(n^3)$. Being n the maximum number between the productions of the grammar and the number of primitives entered to the system.

An approximative pseudocode of the parser algorithm is:

```
While notEmpty(SetofPrimitives)
  el = extractelement(SetofPrimitives)
  TestCompatibility(el,pt)
  continue = true;
  While (notEmpty(SCC) && continue)
    SC = extract(SCC)
    While (notEmpty(SC) && continue)
        rule = extractRule(SC)
        if(ApplyRule(rule,el,pt)) continue = false;
    EndWhile
  EndWhile
 EndWhile
```

where:

- ***SCC*** represents the list of groups of the symbols.
- ***SC*** represents a group of symbols.
- ***rule*** represents a grammatical rule.
- ***pt*** represents the parse-tree.
- ***TestCompatibility(el,pt)*** checks for an incompatibility between the element and the symbol on the top of the parse-tree in the list of incompatibilities. If an incompatibility is found the symbol on the top of the tree is eliminated and the parser goes to a previous step.
- ***ApplyRule(rule,el,pt)*** Tries to apply the rule with the union of the element and the elements of the top of the tree. In case of success the left hand symbol of the production is inserted as the top of the parse-tree.

Figure 6 shows the parser evolution for an input like the shown in Fig. 5 and having an incompatibility between symbol ***SYMBOL3*** and primitive ***segment*** in the incompatibility list. The grammatical rules are the shown in Fig. 2.

The parser starts to work with the input of a primitive of type *segment*, as it can not find any rule that makes a reduction, the primitive is added to the top of the parse-tree and another primitive enters to the system. Until the entry of the fourth primitive, the parser can not apply any reduction. Therefore, the parser inserts the first three primitives to the top of the tree. See Fig.6 steps $(a, b$ and $c)$. The entry of the fourth primitive generates the reduction of the primitives at the top of the parse-tree, adding as a top, the non-terminal symbol specified by the left-hand of the rule accomplished. In this case symbol *QUAD*. This corresponds to Fig. 6 step (d). Figure 6(e) shows the reduction of the top of the parse-tree, provoked by the entry of the next primitive to the system. The

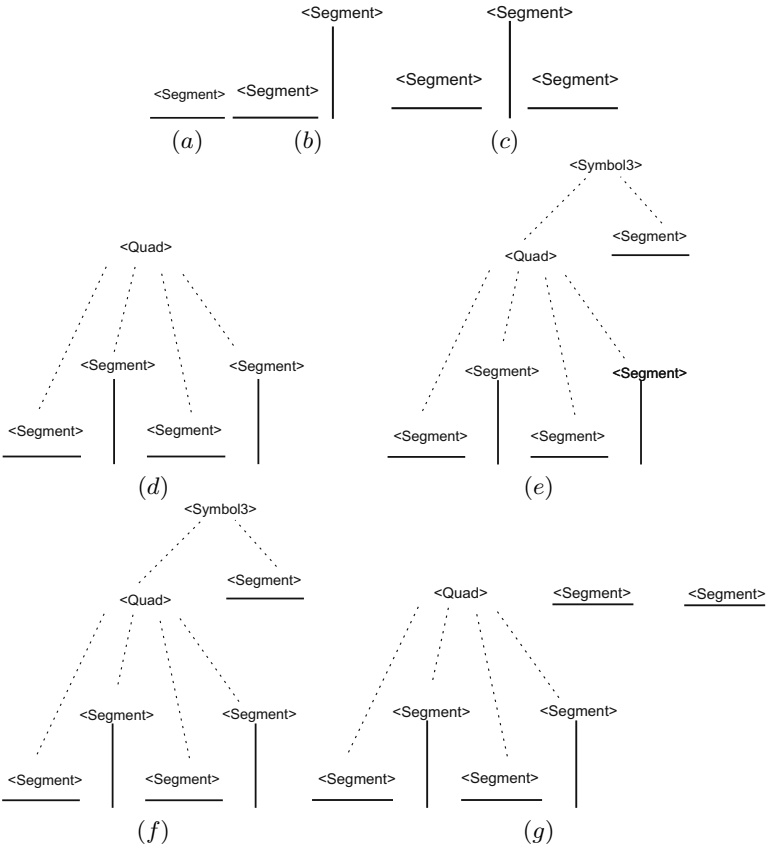


Fig. 6. Evolution of the parser with the input of Fig.5

left-hand symbol of the production *SYMBOL3* becomes the next top of the parse-tree. Having the parser in the state of Fig. 6 step (f) and entering to the system a primitive segment that provokes an incompatibility with the symbol at the top of the parse-tree and the primitive. The parser extracts the symbol from the top and add the new primitive to the new top that belongs to the substraction of symbol *SYMBOL3*, leaving the parse-tree as shown in Fig. 6 step (g). As there is no rule that generates a reduction with the tokens at the top of the parse-tree and there is any primitive to enter to the system, the parser will stop at this point.

4 Experimental Results

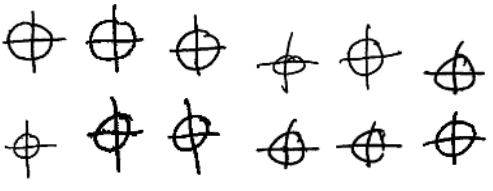
Our parser has been integrated on an architectural sketch recognition application. The aim of the framework is the recognition and 3D-reconstruction of an architectural plan. We have two kinds of symbols to deal with: diagram symbols, as doors, tables, walls, etc. and gestures that allows the user to edit the drawing, deleting, rotating or moving the elements of the draw. This framework has been used to evaluate the performance

Table 1. Results with the online database using Adjacency Grammars with distortion tolerance

	SYMBOL3	SYMBOL5	SYMBOL33	SYMBOL51	SYMBOL52	SYMBOL55	SYMBOL57	SYMBOL46	SYMBOL47	SYMBOL27
PERSON1	100	100	92.31	100	94.12	100	100	100	100	54.55
PERSON2	81.82	100	100	100	81.82	90.91	90.91	90.91	100	100
PERSON3	100	81.82	90.91	90.91	90.91	100	90.91	90.91	90.91	63.64
PERSON4	90.91	100	80	80	100	100	86.67	81.82	63.64	100
PERSON5	82.61	95.65	95.65	95.65	100	100	95.65	100	81.82	90.91
PERSON6	100	90.91	100	100	100	81.82	81.82	81.82	100	81.82
BY SYMBOL	92.55	94.73	93.14	94.42	94.47	95.45	90.99	90.91	89.39	81.82
TOTAL	91.78									

Table 2. Compared results

	SYM.3	SYM.5	SYM.33	SYM.51	SYM.52	SYM.55	SYM.57	SYM.46	SYM.47	SYM.27	**TOTAL**
No Incremental	97.1	94.6	72.4	98.7	74.1	96.2	83.3	97	89.4	74.2	87.7%
Incremental	92.55	94.73	93.14	94.42	94.47	95.45	90.99	90.91	89.39	81.82	91.78%



(a) (b)

Fig. 7. Instances of symbol51: (a) Well recognized instances and (b) Unrecognized instances

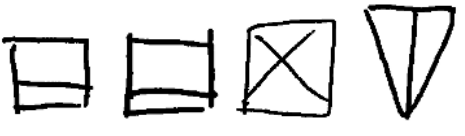


Fig. 8. Instances recognized with a high-value cost

of our method. The reader is referred to [12] for further information of this application framework. From a qualitative point of view, a number of users have been asked to draw different sketches and satisfactory results have been obtained. On the other hand, to quantitatively evaluate the method, a benchmarking database consisting of 700 sketched symbols has been used, obtaining an overall recognition rate of 91.78% as can be seen in the total cell of Table 1.

In Table 1 we can see the results obtained of the recognition of symbols with the grammar and the incremental parser. In this case, we have chosen the instances of 6 people and apply the grammar that contains the definition of 10 symbols. The values in the cells of Table 1, refer to the success ratio by a person in any symbol. Comparing with the results obtained in [13], as shown in Table 2, we can observe an improvement in the overall recognition rate. The problems with some constraints explained in [13],

have been reduced by means of the definition of costs on constraints instead of the range of truth used in the definition. Looking at Table 2 we can observe that some recognition rates by symbol have decreased, this is because in the parser proposed on [13] the author does not take into account incompatibilities among symbols and primitives. The instances that have not been well recognized with the parser have distortions belonging to the third group presented in section 2.

Figure 7(a) show some instances that have been well recognized by the system, opposite in Fig. 7(b) we found some instances that have not been well recognized. Looking at Fig. 7(a) we may seen the instances well recognized, as we can seen this instances are far from perfect compared by the ideal instance of Fig. 4, i.e. the circle was not centered with the segments, the two segments have different lengths, etc. Instances in Fig. 7(b) correspond to instances where an error of misclassification of primitives appears. This error is related to the addition of an extra primitive by the user at time to draw or the detection of an arc instead of a segment and vice versa. In Fig. 8 we may seen some instances that have been well recognized by the system with a high cost value. The ideal instances can be seen in Fig. 4. High distortion on the adjacency constraint may be the main reason to such high values.

5 Conclusions

In this paper we have presented a syntactic methodology to recognize symbols represented by an adjacency grammar. This parser paradigm is conditioned to two requirements. The definition of groups of symbols that are not subparts of other symbols in the same group and the definition of an incompatibility list formed by symbols and primitives. An inaccurate definition of some of these requirements would make the parser to work on an erroneous way. Nevertheless taking into account that a parser is driven by a grammar, it makes necessary a good specification of the symbols forming the grammar as much in the elements forming the rules as the constraints between the elements forming that rules.

Looking at the results of Table 1, we can consider that the adjacency grammar accomplish the aim of describing symbols representing two dimensional shapes in a one dimensional way. Also we can consider that the parser paradigm is a good choice given that the percentage of symbols recognized is of a 91.78%. This percentage involves the recognition of 652 instances over the 700 tested.

We can also denote that some symbols are more likely to be bad drawn by the user. If we look at Table 2 at columns *SYMBOL3* or *SYMBOL51*, we can observe that the recognition rate has decreased. The insertion of additional strokes by the user when handwriting is done, may cause this decreasing on the recognition rate. This fact implies that we need to develop some method of error tolerance.

Concerning to the parser, we can denote that this parser works well with this kind of grammars and the fact that works on an incremental way makes it useful to work with on-line input methodologies. The main drawback of this parser is the complexity. The complexity of this parser is of order $\bigcirc(n^3)$. Being n the maximum number between the number of productions and the number of primitives entered to the system. To solve this problem, in the future work we should consider some pruning techniques or lazy evaluation.

References

1. Fahmy, H., Blonstein, D.: A graph grammar programming style for recognition of music notation. Machine Vision and Applications **6** (1993) 83–99
2. Collin, S., Colnet, D.: Syntactic analysis of technical drawing dimensions. Int. Journal of Pattern Recognition and Artificial Intelligence **8**(5) (1994) 1131–1148
3. Bunke, H.: Attributed programmed graph grammars and their application to schematic diagram interpretation. IEEE Trans. on PAMI **4**(6) (1982) 574–582
4. Adam, S., Ogier, J., Cariou, C., Gardes, J., Mullot, R., Lecourtier, Y.: Combination of invariant pattern recognition primitives on technical documents. In Chhabra, A., Dori, D., eds.: Graphics Recognition - Recent Advances. Springer, Berlin (2000) 238–245 Vol. 1941 of LNCS.
5. Doermann, D., Rivlin, E., Weiss, I.: Applying algebraic and differential invariants for logo recognition. Machine Vision and Applications **9** (1996) 73–86
6. Landay, J., Myers, B.: Sketching interfaces: Toward more human interface design. IEEE Computer **34**(3) (2001) 56–64
7. Jorge, J., Glinert, E.: Online parsing of visual languages using adjacency grammars. In: Proceedings of the 11th International IEEE Symposium on Visual Languages. (1995) 250–257
8. Caetano, A., Goulart, N., Fonseca, M., Jorge, J.: Javasketchit: Issues in sketching the look of user interfaces. aaai spring symposium on sketch understanding (2002)
9. Liu, W., Xiangyu, J., Zhengxing, S.: Sketch-based user interface for inputting graphic objects on small screen device. In Blostein, D., Kwon, Y., eds.: Graphics Recognition: Algorithms and Applications. Springer, Berlin (2002) 67–80 Vol. 2390 of LNCS.
10. Chok, S.S., Marriott, K.: Automatic construction of user interfaces from constraint multiset grammars. In: VL '95: Proceedings of the 11th International IEEE Symposium on Visual Languages, Washington, DC, USA, IEEE Computer Society (1995) 242–249
11. Costagliola, G., Deufemia, V.: Visual language editors based on lr parsing techniques. In: Proceedings of 8th International Workshop on Parsing Technologies. (2003) 79–80
12. Sánchez, G., Valveny, E., Lladós, J., Mas, J., Lozano, N.: A platform to extract knowledge from graphic documents. application to an architectural sketch understanding scenario. In Marinai, S., Dengel, A., eds.: Document Analysis Systems VI. World Scientific (2004) 349–365
13. Mas, J., Sanchez, G., Llados, J.: An adjacency grammar to recognize symbols and gestures in a digital pen framework. In: Proceedings of Second Iberian Conference on Pattern Recognition and Image Analysis. (2005) 115–122

Online Composite Sketchy Shape Recognition Using Dynamic Programming

ZhengXing Sun, Bo Yuan, and Jianfeng Yin

State Key Lab for Novel Software Technology, Nanjing University, P. R. China, 210093
szx@nju.edu.cn

Abstract. This paper presents a solution for online composite sketchy shape recognition. The kernel of the strategy treats both stroke segmentation and sketch recognition as an optimization problem of "fitting to a template". A nested recursive optimization process is then designed by means of dynamic programming to do stroke segmentation and symbol recognition cooperatively by minimizing the fitting errors between inputting patterns and templates. Experimental results prove the effectiveness of the proposed method.

1 Introduction

As people have been using pen and paper to express visual ideas for centuries, the most convenient, efficient and familiar way to input graphics is to draw sketches on a tablet using a digital pen, which is named as sketch-based user interface [1][2]. Owing to the fluent and lightweight nature of freehand drawing, sketch-based user interface is becoming increasingly significant for exploratory and/or creative activities in graphical computing [3]. However, the poor accuracy of online sketchy shape recognition engines is always frustrating, especially for the composite shapes and newly added users, because sketch is usually informal, inconsistent and ambiguous.

In the process of composite sketchy shape recognition, two core phases should indispensably be included [4][5][6], which are respectively named as stroke segmentation and symbol recognition. Almost all of existing methods do them in a sequential manner. That is, the inputting strokes are firstly segmented into a set of geometric primitives in the phase of stroke segmentation, and the configuration of primitives is then recognized as a symbol in the phase of symbol recognition [4]. Obviously, while the recognition results in this process are dependent on both stroke segmentation and symbol recognition, the effectiveness of stroke segmentation would be premised as a precondition for the accuracy of symbol recognition. Existing research experiments do generally stroke segmentation simply with some local features of strokes such as curvature and pen speed [4][7], whereas they have put much more emphases on exploring the robust symbol recognition methods [4][5][6]. However, it could be impossible for these experiments to evaluate if the segmentation results of strokes would be acceptable to symbol recognition. On the one hand, almost all of them use empirical thresholds of local features to test the validity of an approximation of strokes. In fact, strokes of freehand drawing vary with different users, even the same user at different time. This ultimately leads to the problem of a

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 255 – 266, 2006.
© Springer-Verlag Berlin Heidelberg 2006

threshold being too tight for one user while too loose for another. On the other hand, all of them do stroke segmentation independently no matter what the symbol recognized might be. Actually, the user is purposeful with the intended symbol in mind when expressing his/her ideas with the particular sketchy shape based on both the current observations and the past experiences, though he/she draws one stroke after another. That is to say, stroke segmentation for any users is much more related to the ultimate intended symbols besides the local features and temporal information. This eventually results in the problem of the local features being only suited to some shapes but not to others.

Accordingly, this paper will present an improved solution for online composite sketchy shape recognition, which is described as follows. As the kernel idea of our solution, we firstly regard both stroke segmentation and symbol recognition as the optimization problem of "fitting to a template". The purpose of this strategy is twofold. On the one hand, templates can be used as both a guide to the process of stroke segmentation together with local features and an evaluation reference of an approximation of strokes. On the other hand, the process of both stroke segmentation and symbol recognition would be unified in the process of "fitting to a template". In order to implement our proposed strategy, we define a nested recursive optimization process to do cooperatively stroke segmentation and symbol recognition by minimizing the fitting errors between inputting patterns and the defined shape models of the symbol (templates). This results in the combination of stroke segmentation and symbol recognition. Meanwhile, dynamic programming is adopted to implement this combined process and reduce the computing complexity, which is an efficient way of solving problems where you need to find the best decisions one after another [8].

The remainder of this paper is organized as follows: The related works in sketch recognition are summarized in Section 2. In Section 3, the main idea of our proposed strategy is outlined. In Section 4, details of our method will be discussed. Section 5 will present our experiments. Conclusions are given in the final Section.

2 Related Works

A variety of recognition techniques have been proposed for sketch recognition. They can be mainly classified into three categories: feature-based methods, graph-based methods and machine learning methods.

Feature-based methods make use of some features of sketchy shapes [2][9][10]. The benefit of feature-based approaches is that stroke segmentation is not necessary, while their drawback is that they can only recognize simple shapes, such as rectangles and circles. For example, Rubine [9] defined a gesture characterized by a set of eleven geometric attributes and two dynamic attributes. However, it is only applicable to single-stroke sketches and sensitive to drawing direction and orientation. Fonseca et al [10] proposed a method of symbol recognition using fuzzy logic based on a number of rotation invariant global features. Because their classification method relies on aggregate features of pen strokes, it might be difficult to differentiate between similar shapes.

As one of the most prominent approaches to object representation and matching, graph-based methods have been recently applied to hand-drawn pattern recognition

problems [4][5][6]. In these methods, input patterns are first decomposed into basic geometric primitives and then assembled into a graph structure. Pattern detection is then formulated as a graph isomorphism problem. For these methods, a hypothesis must be abided that all of inputting strokes have felicitously been segmented. As a structural (or topological) representation of composite graphic objects, these methods are theoretically suitable for all composite shapes with variant complexities. The practical limitations for them are their computational complexity. Furthermore, their performance degrades drastically when applied to drawings that are heavily sketchy.

Inspired by the success of machine learning methods in pattern recognition, researchers have recently took them to freehand drawing recognition. Sezgin et al [11] view the observed pattern as the result of a stochastic process governed by a hidden stochastic model, and identified the hidden stochastic model according to its probability of generating the output. In our previous researches, we have also made attempt to solve the problems of user adaptation for online sketch recognition based on machine learning method such as Support Vector Machine (SVM) [12] and Hidden Markov Model (HMM) [13]. Nevertheless, they have traditionally employed statistical learning methods where each shape is learned from a large corpus of training data. Due to their need for large training sets, these methods are not easily extensible to diverse applications and are normally useful only for the patterns they are originally trained for.

In the early researches of stroke segmentation, the local curvatures of strokes were used for the detection of corner (segment) points, which usually help decompose the original stroke into basic primitives such as lines and arcs. Usually, curvature information alone is not a reliable way to determine such points. Temporal information such as pen speed has been recently explored as a means to uncover users' intentions during sketching [7]. The speed based methods have been proven much more reliable to determine the intended segment points. Sezgin [7] and later Calhoun [4] used both curvature and speed information in a stroke to locate segment points. Saund [14] has used more perceptual context, including local features such as curvature and intersections, as well as global features such as closed paths. However, all of them use empirical thresholds to test the validity of an approximation. As an improvement, Heloise et al [15] have used dynamic programming to approximate recursively a digitized curve with a given number of line and arc segments based on templates. The main difference between their method and ours is that they put attention only to stroke segmentation for simple shapes.

3 Strategy Overview

The flowchart comparison between the traditional methods and our strategy for online composite sketchy shape recognition is shown in **Fig. 1**, both designed to work for recognizing single isolated symbols. Both of the two processes contain four sub-processes: *ink pre-processing*, *stroke segmentation*, *symbol recognition* and *user mediation*. As the user draws, the *ink pre-processing* is firstly used to eliminate the noise that may come from restriction of input condition and input habits, and to normalize and uniform the positional distribution of sample points. The *stroke segmentation* process involves searching along the stroke for "segment points" that

divide the stroke into different geometric primitives. The candidate segment points of each stroke are generated using both the motion of the pen and the curvature of the resulting ink [16]. Once the segment points have been identified, some geometric primitives are selected to match closely to the original ink and provide compact descriptions of the strokes that facilitate sketch recognition. Subsequently, the *symbol recognition* takes the geometric primitives composing the candidate symbols as input and returns some better definitions ranked by the geometric and topological similarity measure between the candidates and templates. Knowledge about the particular domain of the symbols must be used to prune the list of candidate symbols if possible. *User mediation* is designated for users to evaluate implicitly or explicitly the results of the symbol recognition, and to refine the recognition results if necessary [17].

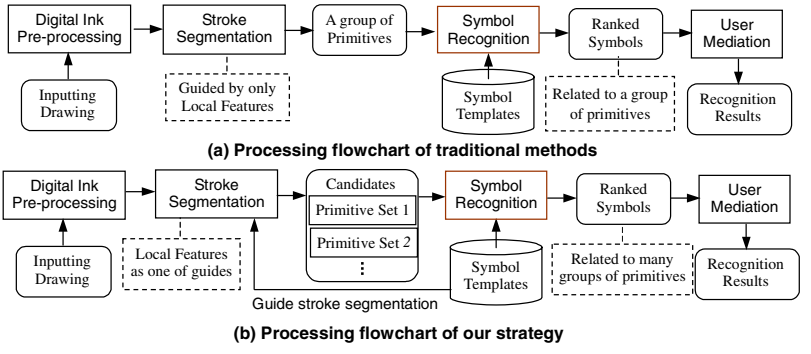


Fig. 1. Flowchart comparison between the traditional methods and our strategy

Most of traditional composite sketch recognition methods do stroke segmentation independently according to some local features of strokes no matter what the symbols recognized might be, as shown in **Fig. 1**(a). In these approaches, a set of primitives resulting from stroke segmentation is generally exclusive, and the results of symbol recognition are heavily sensitive to the segmentation results. This leads to a problem that a little mistake in stroke segmentation will cause a fatal error in symbol recognition.

The main improvement of our strategy for composite sketchy shape recognition is that both stroke segmentation and symbol recognition are treated as the problem of fitting with template, as shown in **Fig. 1**(b). Besides the local features of strokes, we make use of the primitives of shapes in templates to guide stroke segmentation, where the "segment points" is detected according to the fitting errors between the primitives of segmented strokes and the primitives of symbols in templates. Some groups of primitives of segmented strokes with smaller fitting error are finally given as the candidate results. Symbol recognition can then search in terms of the fitting errors between the shape in these candidates and the symbols in templates, and return some of them with small fitting errors as recognition outputs.

Because the number of the candidate results of stroke segmentation is exponential in the size of candidate segment points, an exhaustive search is obviously a poor strategy. Therefore, we design a nested recursive search process and adapt dynamic

programming approach to optimize the search process, where the inner do stroke segmentation and the outer do symbol recognition.

4 Sketchy Shape Recognition Based on Dynamic Programming

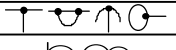
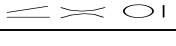
4.1 Sketchy Shape Representation and Fitting Error Calculation

During freehand drawing, the contents what users stressed would be the spatial or topological relationships between primitives of the sketch. Accordingly, a symbol can be described as: $S=(P,TP,AP,R,TR,AR)$, where, P is a set of primitives, TP is the types of primitives; AP is the attributes of primitives; R is a set of topological relationships between primitives; TR is the types of topologic relations, AR is the attributes of topological relationships.

There are mainly two types of primitives in our researches: line and ellipse segment, while arc segment is treated as an instance of ellipse, that is: $TP=(TP_l,TP_e)$. The attributes of primitives are defined by $AP=(C_0,C_1,C_2)$, where, for line segment, C_0 and C_1 are the start and end points of segment respectively and $C_2 = 0$; for ellipse segment, C_0 is the center point of the ellipse, C_1 and C_2 are the long and short axes respectively; for arc segment, C_0 is the center point of the arc, C_1 and C_2 are start and end angle of arc counted in clockwise respectively.

We consider six kinds of topologic relationships in our researches, including *adjacency*, *cross*, *half-cross*, *tangency*, *parallelism* and *separation*, as shown in **Table 1**. These relationships and their attributes can be represented as $TR=(R_a,R_c,R_h,R_t,R_p,R_s)$ and $AR=(AR_a,AR_c,AR_h,AR_t,AR_p,AR_s)$ respectively. The attributes AR_a,AR_c,AR_h and AR_t are defined by the angle between the two primitives at the common point in clockwise, as shown in **Fig. 2**(a) and (b). If one of the two primitives is an arc segment, the angle can be calculated in terms of a tangent line or a local chord at a common point on the arc, as shown in **Fig. 2**(c). The attribute AR_p is defined by the ratio of the length of the superposition part of the two primitives to the average length of them as shown in **Fig. 2**(d). The attribute AR_s sets to null.

Table 1. Six kinds of topologic relationships

Sign	Types	Examples	Descriptions
R_a	*Adjacency*		Two segments connect at least one end point.
R_c	*Cross*		Two segments intersect each other.
R_h	*Half-cross*		A segment ends at another.
R_t	*Tangency*		A segment is tangent to another.
R_p	*Parallelism*		Two segments are parallel or concentric.
R_s	*Separation*		Two segments do not intersect each other.

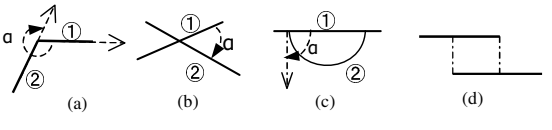


Fig. 2. Attributes definition of topological relationships between two primitives

A general representation of a symbol is an attributed relationship graph, such as spatial relationship graph (*SRG*) [6]. However, graph matching is a well-known NPC problem, especially for composite sketchy shape recognition. In our research, we define an ordered topological relationship chains (*OTRC*) to represent the composition of a freehand drawing. An ordered topological relationship chain is a temporal ordered list of nodes, each of which records the attributes of a primitive and its relationships with its neighboring primitives in the composition of the symbol. This representation can largely reduce the computing complexity of matching, which will not limit users' drawing orders. Furthermore, the temporal sequence can be used in user modeling to record statistically drawing orders of every type of shapes for a particular user to improve the recognition efficiency [17], as our experiments have shown that there is generally a fixed temporal order for primitive shapes when the same user draws the same shapes.

In our strategy for composite sketchy shape recognition, the *fitting error f*(*SM*,*T*) between the inputting patterns (*SM*) and the templates (*T*) must be calculated. There are two types of fitting errors: the *topological matching error* $f_r(R^{SM},R^T)$ between the inputting patterns and templates and the *approximating error* $f(P^{SM},P^T)$ to approximate the segments of stoke.

We define some rules for the *topological matching error* $f_R(R^{SM}, R^T)$ as fallows:

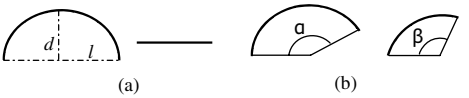
$$f_r(R_1,R_2)=\begin{cases}1, \ If\ \ TR_1 \neq TR_2;\\ \left(\left|\alpha_1-\alpha_2\right|\right)/2\pi, \ If\ \ TR_1=TR_2\equiv R_a;\\ \left(\left|\alpha_1-\alpha_2\right|\right)/(\pi/2), \ If\ \ TR_1=TR_2\equiv R_h,R_c,R_t;\\ \left(\left|L_1-L_2\right|\right)/(L_1+L_2), \ If\ \ TR_1=TR_2\equiv R_p;\\ 0.5, \ If\ \ TR_1=TR_2\equiv R_s.\end{cases} \tag{1}$$

where α is the angle between two interrelated primitives and L is the length of superposed segment between two parallel primitives, as shown in **Fig. 2**.

The *approximating error* $f(P^{SM},P^T)$ contains two parts. The first is *the primitive matching error* $f_p(P^{SM},P^T)$, which is defined as a similarity measure between the segments of stoke and the primitives of symbols in the templates. We define this error as following rules:

$$f_P(P_1,P_2)=\begin{cases}0, \ If\ \ TP_1=TP_2=TP_l;\\ d/(l+d), \ If\ \ TP_1=TP_e, TP_2=TP_l;\\ \left(\left|\alpha-\beta\right|\right)/(\alpha+\beta), \ If\ \ TP_1=TP_2=TP_e.\end{cases} \tag{2}$$

where, ***d*** and ***l*** are the height and chord of the arc respectively, as shown in **Fig. 3**(a); α and β are the centric angles of the two arcs respectively, as shown in **Fig. 3**(b).



(a) (b)

Fig. 3. Calculation of the fitting error between two primitives

The second is *the primitive approximating error* $f_a(P^{SM})$ that is defines as a approximating measure to approximate segments of stoke with basic primitives, and treated as a distortion factor for the primitive matching error. Calculation of this error varies with the type of primitives used for approximating: if $TP=TP_l$, $f_a(P^{SM})$ is the error of total least square fitting on the ink points; if $TP=TP_e$, $f_a(P^{SM})$ is the error of elliptical fitting [18]. They are finally normalized to ranges between 0 and 1.

Accordingly, the *fitting error* $f(SM,T)$ between inputting patterns and template can be defined as follow:

$$f(SM,T)=\sum_{i=1}^{N_p} f_a\left(P^{SM}\right)\times f_p(P^{SM}(i),P^T(i))+\sum_{i=1}^{N_p-1} f_r(R^{SM}(i),R^T(i)) \tag{3}$$

where, N_P are the number of primitives and their relationships; P^{SM} and P^T, R^{SM} and R^T are the primitives and spatial relationships between primitives in a inputting pattern and a templates used for matching respectively.

4.2 Stroke Segmentation Using Templates

We regard stroke segmentation as a problem of "fitting to a template", which is an optimization problem to select a set of segment points from the candidates of input strokes to fit to the template with minimal fitting error.

Given a freehand drawing *SM* and a template *T*, *SM* is consisted of a sequence of strokes $\{S_i\}$, each stroke contains a set of temporal ordered candidate segment points $\{P_{ij}|\}$; a template *T* is represented as a set of primitives $\{t(i)\}$. The number of segment points needed to be identified is: $k=NT-NS$ (in general, $NS\leq NT\leq NB-1$, where *NT* is the number of primitives of a symbol in templates, *NS* is the number of strokes and *NB* is the total numbers of ordered candidate segment points including the start and end points of strokes, NB_i is the number of ordered candidate segment points for the i^{th} stroke.). The problem of stroke segmentation using templates can then be defined as to select *k* segment points from the ordered candidate segment points to segment the drawing into $k+1$ segments such that the drawing represented by these segments is fit with several candidate symbols in the templates with minimal fitting errors. A brute-force approach would do an exhaustive search on all the combinations of *k* segment points from the *NB* candidate segment points. However, the number of combinations is exponential to the size of *NB*. So, we simplify the search process by means of Dynamic Programming (DP) techniques [8].

Dynamic programming tends to break the original problem to sub-problems and chooses the best solution in the sub-problems, beginning from the smaller in size. The best solution in the bigger sub-problems is found by using the best ones of the smaller sub-problems through a retroactive formula, which connects the solutions [8]. Accordingly, we define firstly the optimal substructure for the segmentation problem. For a selected segment point, an optimal segmentation contains the optimal segmentation of the input stroke(s) up to this point. In other words, to find an optimal segmentation of a set of strokes with template *T* composed of *NT* numbers of ordered primitives, one assumes that the optimal solution for fragmenting everything up to the selected segment point with a template $T\{t(1), t(2),\ldots, t(NT-1)\}$ has been computed, and the piece from the selected segment point to the end is then fit with $T\{t(NT)\}$.

A recursive solution is then designed based on the above optimal substructure. Let $d(n,m,k,t)$ be a minimal fitting error to approximate every point up to the m^{th} point in

the n^{th} stroke with template t, and let $f(S_n,i,m,t(j))$ be the fitting error resulting from fitting the segment from the i^{th} point up to the m^{th} point in the n^{th} stroke using $t(j)$. The best segmentation for a set of strokes with NS strokes using K segment points and a template T would thus be $d(NS, NB,K,T)$. The recursive definition of $d(n,m,k,t)$ is expressed as follows [15]:

$$d(n,m,k,t)=\begin{cases}\left[\sum_{i=1}^{n-1} f(S_i,1,NB_i,t(i))\right]+f(S_n,1,m,t(n)), & \text{if } k=0;\\ \min_{k<i<m}\{f(S_n,i,m,t(NT))+d(n,i,k-1,t(j\mid j=1,\cdots,NT-1))\}, & \text{if } n=1,k>0;\\ \min\begin{cases} f(S_n,1,m,t(NT))+d(n-1,NB_{n-1},k,t(j\mid j=1,\cdots,NT-1)) \\ \min_{k<i<m}\{f(S_n,i,m,t(NT))+d(n,i,k-1,t(j\mid j=1,\cdots,NT-1))\}\end{cases}, & \text{if } n>1,k>0.\end{cases} \tag{4}$$

where, the approximating error can be defined as follow:

$$f(S_n,i,m,t(j))=f_a(S_n,i,m)\times f_p(S_n,i,m,t(j)) \tag{5}$$

When $k=0$, each of the strokes is fit with the corresponding primitive in the template and the segment from P_{i1} to P_{im} in the n^{th} stroke is compared with the i^{th} primitive in the template. When $n=1$ and $k>0$, a choice has to be made on which point $P_{ni}(i>k)$ to be the breakpoint, otherwise, the number of segment points required would exceed the number of candidate segment points available. When $n>1$ and $k>0$, in addition to checking the best breakpoint to use in S_n, the previous stroke (S_{n-1}) must also be checked because it is possible that the best breakpoint may lie in any of the previous strokes. Due to the optimal substructure, the optimal segmentation for the last point in the previous stroke S_{n-1} is all that must be checked.

For the unordered templates, we design a nested recursive search process based on dynamic programming for stroke segmentation, where the inner takes charge for all cases of the order of primitives in a template, the outer carries out all possible segmenting segments for a stroke.

The algorithm has a run time complexity of $O(K\times NB^2)$, where K is the number of segment points and NB is the total number of candidate segment points. The space requirement is $O(K\times NB)$ for keeping a table of solutions to the sub-problems. However, because the total number of candidate segment points is far less then of ink points of a sketchy symbol, our algorithm is much faster than the algorithm in [15].

4.3 Sketchy Shape Recognition Using Dynamic Programming

Sketchy shape recognition is actually a problem of "fitting to a template". Similar to the recursive solution of stroke segmentation using template, we design a nested recursive process by means of dynamic programming to do sketch recognition and stroke segmentation cooperatively. That is, the inner circulation process searches all of the possible segment points by minimizing *the approximating error* as described in section 4.2, while the outer searches all of the possible symbols by minimizing the *fitting error*.

Accordingly, let $d(n,m,k,t,j)$ be a minimal fitting error to approximate every segment up to the m^{th} segment in the n^{th} stroke with the template t, and let $f(S_n,i,m,t(u),j)$ be the fitting error resulting from fitting the segment from the i^{th} point

up to the m^{th} point in the n^{th} stroke using $t(u)$, where j is the index of primitive in the templates. The best result for a set of strokes with NS strokes using K segment points and a template T would thus be $d(NS, NB,K,T,0)$. The recursive definition of $d(n,m,k,t,j)$ can be then expressed as follows:

$$d(n,m,k,t,j)=\begin{cases} f(S_1,1,NB_n,t[NT],j), \text{ if } NT=1,k=0; \\ \min\limits_{0<u<NT+1}\{f(S_n,1,m,t[u],j)+d(n-1,NB_{n-1},k,t-t[u],u)\}, \text{ if } NT>1,k=0; \\ \min\limits_{0<u<NT+1}\{\min\limits_{m<v<NT-k}\{f(S_n,v,m,t[u],j)+d(n,v,k-1,t-t[u],u)\}\}, \text{ if } n=1,k>0; \\ \min\limits_{0<u<NT+1}\left\{\min\left\{\begin{array}{l} f(S_n,1,m,t[u],j)+d(n-1,NB_{n-1},k,t-t[u],u) \\ \min\limits_{m<v<NT-k}\{f(S_n,v,m,t[u],j)+d(n,v,k-1,t-t[u],u)\} \end{array}\right\}\right\}, \text{ if } n>1,k>0. \end{cases} \tag{6}$$

where, the *fitting error* can be defined as follow:

$$f(S_n,i,m,t(u),j)=\begin{cases} f_a(S_n,i,m), \text{ if } \quad j=0; \\ f_a(S_n,i,m)\times\left(f_p(S_n,i,m,t(u))+f_r(S_n,i,m,t(u),j)\right), \text{if } \quad j>0. \end{cases} \tag{7}$$

When $k=0$, each of the strokes is fit with the corresponding basis in the template and the segment from P_{i1} to P_{im} in the n^{th} stroke is compared with the u^{th} primitive in the template.

When n=1, k>0, the inner circulation of recognition ransacks all of the possible segmentation, while the outer searches all of the possible patterns in the template.

When n>1, k>0, the inner looks for all of the possible segmentation. There are two different circumstances should be considered. The first case is that there are no segment points in the current stroke. Therefore, the current stroke will be only compared with a primitive of template, and the others will be compared with the substructure of the template. Another case is that there is at least one breakpoint in the first stroke. Suppose the m^{th} candidate breakpoint P_{nm} is a real breakpoint in the current stroke, we will calculate the error between the segments of stroke from the first point to P_{nm} and one of the primitives in the template, and the other segments of first stroke and the remainder strokes will be compared with the primitives in of the template one by one. The outside circulation will ransack all the possible patterns of the primitives in the template.

The algorithm has a time complexity of $O(K\times NB^2\times NT^2)$, where K is the number of segment points, NB is the total number of candidate segment points and NT is the total number of primitives for each template.

5 Experiments and Discussion

To verify our proposed method, we have done two experiments on templates with 150 types of predefined shapes as shown in **Fig. 4** (20 simples in left-top- side of **Fig. 4**, 96 moderates and 34 complexities in right-down side of **Fig. 4**), where a shape is simple if it is composed of less than three primitives and drawn by a few strokes, moderate if it is composed of 3-6 primitives and drawn by 3-6 strokes and complex if

it is composed of morn than 6 primitives and drawn by more than 6 strokes. All experiments are done on an Intel PC (with a 2.8 GHz CPU and 512MB memory) running Microsoft Windows XP Professional.

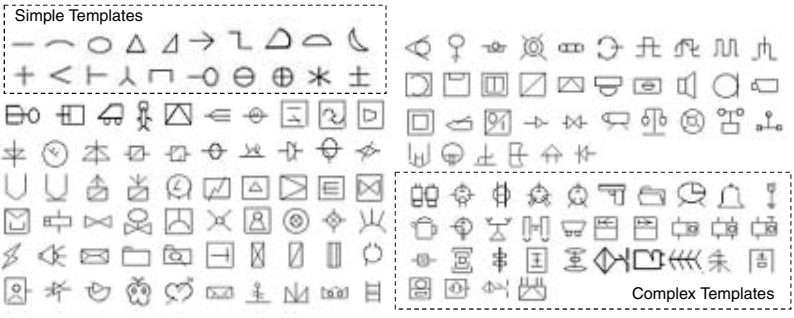
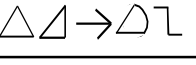
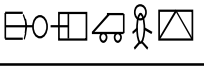


Fig. 4. Shapes in our templates for online composite sketchy shape recognition

The first experiment is to measure the effectiveness and efficiency of our method. Some of our experimental results are listed in **Table 2**, where three types of shapes with two different drawing styles are illustrated. From **Table 2** we can firstly see that all of the correct recognized results for three drawings are ranked in the front of top 5 results with minimal fitting error. Furthermore, although the number of strokes and segment points may be different for the two drawing styles of a symbol, the number of recognized primitives is same. This means that our algorithm is not sensitive to drawing styles, because stroke segmentation would be guided well by the templates in our proposed strategy. It must be indicated that the top 5 results listed in fourth row of **Table 2** are mainly with the same number of primitives especially for moderate and complex shapes because there are few shapes with similar topological relationships in our templates as shown in **Fig. 4**. Thirdly, the computing time is proportionate to the complexity of drawings, as shown in the final row of **Table 2**, through we have taken a strategy of "swapping storage space for run time" by defining an ordered chain. This is because a stroke can be segmented in more different ways and a drawing may also

Table 2. Experimental results of three typical sketches

Type of Shape	Typical sketches	Number of strokes, segment points and primitives	Top 5 results with min. fitting errors	Computing Cost (ms)
Simple Shapes		1, 2, 3		7.8
		1, 2, 3		7.9
Moderate Shapes		6, 2, 6		123
		3, 5, 6		183
Complex Shapes		9, 0, 9		226
		6, 4, 9		398

be recognized as more different types of symbols as the complexity of symbols increase. The computing complexity could be decreased either if there are small numbers of templates or if templates of ordered primitives are used as discussed in section 4.2. However, our method is still acceptable in practice, because we aim mainly at conceptual design where a user usually express his/her ideas only with not too many numbers of templates, which are much less than the 150 templates in our experiments.

The second experiment we have done is to test the adaptability of our method for different users. We select four types of shapes as shown in **Fig. 5**(a) and ask four users to draw 100 examples for every type of shapes. The average recognition accuracy of every type of shape for each user is shown in **Fig. 5**(b). From **Fig. 5**(b) we can see that all of the average accuracies are higher than 95%. This proves further that our method is not sensitive to variation of drawing styles.

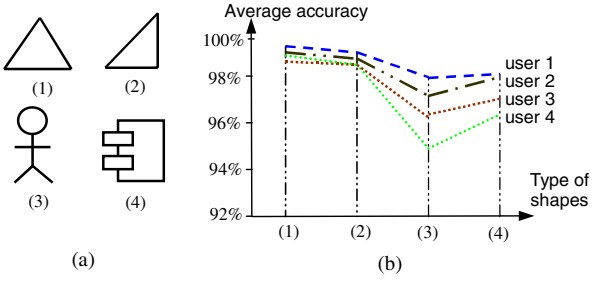


Fig. 5. Experimental results of different users

6 Conclusions

In this paper, a strategy is presented for composite sketchy shape recognition. In our strategy, both stroke segmentation and symbol recognition are uniformed as the problem of "fitting to a template" with minimal fitting error. A nested recursive process is designed to implement the optimization based on dynamic programming technology. A distinct advantage of the method is that it can be adapted to different drawing styles because stroke segmentation would be guided well by the templates. Our experimental results have proved both effectiveness and efficiency of the proposed method. The main limitation to our method is that the computing cost is proportionate to the complexity of drawings and sensitive to the size of templates. This is the direction of our further improvements.

Acknowledgement

This paper is supported by the grants from the National Natural Science Foundation of China [Project No. **69903006** and **60373065**] and the Program for New Century Excellent Talents in University of China [Project No. **NCET-04-04605**].

References

1. Landay J A and Myers B A, Sketching interfaces: toward more human interface design, IEEE Computer, vol. 34, no. 3, 2001, pp. 56-64.
2. Wenyin Liu, Xiangyu Jin and Zhengxing Sun, Sketch-based user interface for inputting graphic objects on small screen device, Lecture Notes in Computer Science, Vol. 2390, 2002, pp. 67-85.
3. Zhengxing Sun and Jing Liu, Informal user interfaces for graphical computing, Lecture Notes in Computer Science, Vol. 3784, 2005, pp. 675-682.
4. Calhoun C., Stahovich T. F., Kurtoglu T., et al, Recognizing multi-stroke symbols, In: AAAI Spring Symposium on Sketch Understanding, AAAI Press, 2002, pp.15-23.
5. Seong-Whan Lee, Recognizing hand-drawn electrical circuit symbols with attributed graph matching, Structured Document Image Analysis, 1992, pp. 340-358.
6. Xiaogang Xu, Zhengxing Sun, Binbin Peng, et al, An online composite graphics recognition approach based on matching of spatial relation graphs, International Journal of Document Analysis and Recognition, Vol. 7, No.1, 2004, pp. 44-55.
7. Sezgin T. M., Stahovich T., Davis R., Sketch-based interfaces: early processing for sketch understanding. Proceedings of the 2001 Workshop on PUI, Orlando, Florida, 2001, pp.1-8.
8. Bellman R. E., Dynamic Programming, Princeton University Press, 1957.
9. Rubine Dean, Specifying gestures by example, Computer Graphics, 1991, Vol. 25, pp. 329-337.
10. Fonseca M. J., Pimentel C. and Jorge J. A., CALI-an online scribble recognizer for calligraphic interfaces, In: AAAI Spring Symposium on Sketch Understanding, AAAI Press, 2002, pp. 51-58.
11. Sezgin T. M. and Davis R., HMM-based efficient sketch recognition, Proceedings of the 10th international conference on IUI, Jan., 2005, San Diego, California, USA.
12. Zhengxing Sun, Wenyin Liu, Binbin Peng, et al, User adaptation for online sketchy shape recognition, Lecture Notes in Computer Science, Vol. 3088, 2004, pp. 303-314.
13. Zhengxing Sun, Wei Jiang and Jianyong Sun, Adaptive online multi-stroke sketch recognition based on Hidden Markov model, Lecture Notes in Artificial Intelligences, Springer-Veralg, Vol. 3930, 2006, pp. 948-957.
14. Saund, E, Finding Perceptually closed paths in sketches and drawings, IEEE Transactions on PAMI, Vol. 25, No. 4, 2003, pp. 475-491.
15. Heloise H., Michael S., A. Richard N., Robust sketched symbol segmentation using templates, International Conference on Intelligent User interface, ACM Press, pp.156-160.
16. Zhengxing Sun, Wei Wang W, Liu J, Sketch parameterization using recursive curve approximation, Proceedings of GREC 2005, Hong Kong, China, 2005, pp.325-334.
17. Zhengxing Sun, Bin Li, Qiang Wang and Guihuan Feng, Dynamic user modeling for sketch-based user interface, Lecture Notes in Computer Science, Springer-Veralg, 2006, to be published.
18. Pilu M., Fitzgibbon A. and Fisher R, Direct least-square fitting of ellipses, IEEE Trans. PAMI, 21(5), 476-480.

Using a Neighbourhood Graph Based on Voronoï Tessellation with DMOS, a Generic Method for Structured Document Recognition

Aurélie Lemaitre[1], Bertrand Coüasnon[1], and Ivan Leplumey[2]

[1] IRISA/INRIA, Campus universitaire de Beaulieu, F-35042 Rennes Cedex, France
[2] IRISA/INSA, Campus universitaire de Beaulieu, F-35042 Rennes Cedex, France

Abstract. To develop a method for structured document recognition, it is necessary to know the relative position of the graphical elements in a document. In order to deal with this notion, we build a neighbourhood graph based on Voronoï tessellation. We propose to combine the use of this interesting notion of neighbourhood with an existing generic document recognition method, DMOS, which has been used to describe various kinds of documents. This association allows exploiting different aspects of the neighbourhood graph, separating the graph analysis from the knowledge linked to a kind of document, and establishing a bi-directional context-based relation between the analyser and the graph. We apply this method on the analysis of various documents.

1 Introduction

In the field of structured document recognition, the knowledge on relative position between the graphical elements of a document is often necessary. Voronoï tessellation of image, and the dual Delaunay graph, provide an interesting description of this concept of neighbourhood. This method is used in several papers for structure recognition of document images, in the context of specific applications: detection of lines, words, segments. We propose to exploit such information in a generic context, using an existing document recognition method, DMOS.

Indeed, in the standard version of DMOS method, the relative position of elements is given with an approximation. That is why we propose to introduce neighbourhood graph based on Voronoï tessellation, which offers a precise notion of relative position. Furthermore, using the graph with DMOS makes it possible to extract local numerical information depending on a context that is determined by symbolic information contained in DMOS method.

In a first part, we will see relative work on Voronoï diagram in the field of structured document recognition, and the associated neighbourhood graph that we have implemented. Then, we will present DMOS method and the integration of neighbourhood that has been realized. We will expose afterwards applications that have been set up in order to validate these tools. We will end by a discussion.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 267–278, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Neighbourhood Graph Based on Area Voronoï Diagram

We recall a few definitions, describe related work on document recognition, and present our implementation of a neighbourhood graph.

2.1 Definitions

Definitions of Voronoï Diagram are given in [8] and [7]. We present the basic points.

Classical Voronoï Diagram. The classical Voronoï diagram cuts up the area into influence regions of points. Let P $=\{p_1, ..., p_n\}$ be a set of points from the plan, called *generators*, and $d(p, q)$ be the Euclidean distance between points p and q. Then, the *Voronoï region* of a point p_i is given by

$$V(p_i) = \{p | d(p, p_i) \leq d(p, p_j), \forall j \neq i\}$$

It is the set of points that is nearest to this generator than any other.

The ordinary Voronoï diagram is given by the set of Voronoï region:

$$V(P) = \{V(p_i), ..., V(p_n)\}$$

We usually associate to Voronoï diagram the dual Delaunay graph that is composed of the same set of vertex P, and which contains a edge between points p and q if they are neighbours in the Voronoï diagram.

Area Voronoï Diagram. The basic Voronoï diagram has been generalized in several directions. One of the possible generalizations consists in replacing the set of points, the *generators*, by a set of connected components. A connected component is a set of black pixels that are in contact. We present an example in Fig.1(c). Such a diagram is called *area Voronoï diagram*.

2.2 Related Work on Structured Document Recognition

The area Voronoï diagram has been used a lot for structure detection of images as it enables to know the component's nearest neighbours.

Thanks to this information, various methods have been proposed to segment documents into words, lines, paragraphs, and columns. Generally, like in [4] or [6], the knowledge that is necessary for the analysis is included by learning thresholds like inter-character, inter-word and inter-text line gaps.

However, these thresholds are learnt statistically on the whole document. Thus, all the knowledge that is introduced is relative to the global image, and it is not dissociated from the exploitation of the Voronoï tessellation. Consequently, the analysis is limited to quite homogeneous documents, and the knowledge is reduced and appropriated just for one kind of document.

In order to extract more information from Voronoï diagram, we propose to separate the neighbourhood graph analysis from the necessary knowledge. That is why we use the neighbourhood graph with a generic method for structured document recognition: DMOS.

(a) Image (b) Distance image (c) Area Voronoï diagram (d) Neighbourhood graph

Fig. 1. Construction of neighbourhood graph

2.3 Implementation of a Neighbourhood Graph Based on Discrete Distance

The area Voronoï diagram construction is often based on merging classical Voronoï diagram, built with a set of points from the contours of components. Several methods are proposed in [8]. The difficulty of this approach is to select the convenient points from the contour. Another method in [7] is based on iterative expansion of convex polygons associated to connected components. However, it assumes that convex polygons englobing each component do not overlap others; this is not always the case in handwritten documents.

We have chosen to implement a neighbourhood graph, labelled with discrete distances, whose implementation has been detailed in [5]. The principle is to apply chamfer distance by propagation on the initial image. We obtain, as a result, three images that contain respectively, for each pixel:

- the discrete distance to the nearest connected component (Fig.1(b));
- the name of the nearest connected component (Fig.1(c)). Indeed, this image is the approximate area Voronoï diagram;
- the coordinates of the nearest point of the nearest component.

Thanks to these three images, we can build a neighbourhood graph (Fig.1(d)), labelled with distances, more complete than a mere Voronoï tessellation. This graph can be exploited for document analysis with method DMOS.

3 A Generic Method for Structure Recognition

3.1 DMOS Method

We presented in various papers ([1], [2]) DMOS (Description and Modification of Segmentation), a generic method for structured document recognition. This method is made of:

- the grammatical formalism EPF (Enhanced Position Formalism), which makes possible a graphical, syntactic and even semantic description of a class of documents;
- the associated parser which is able to change the parsed structure during the analysis. This allows the system to try other segmentations with the help of context to improve recognition.

This DMOS method aims at generating automatically structured document recognition systems. Thus, by only changing the EPF grammar, we produced various recognition systems: one on musical scores, one on mathematical formulae or several for archive documents [3], which proves the genericity of the method. Moreover, these grammars have been validated on large document bases (165,000 archive documents for example).

3.2 EPF Formalism

The grammatical EPF formalism is based on several operators that make possible a two-dimension document description: the position of each element is specified relatively to the others. We introduce the main operators on a simple example and then explain the way they are used to analyse the document.

Example on a Simple Grammar. The simplified grammar presented here describes a mathematical formula based on an integral, like in Fig.1(a).

Intuitively, we can describe such a document by: an integral symbol, on the left part of the image; integration bounds, on the top and bottom part of the integral; an expression, on the right of the integral.

The grammar rules will follow this intuitive description, thanks to two position operators: `AT` gives the position of an element relatively to a previous one; `AT_ABS` is an absolute position of the element. Then, the simplified main rule is:

```
integralFormula::=
    AT_ABS(leftPicture) && integralSymbol && (
    AT(topRight integralSymbol) && bound ##
    AT(bottomRight integralSymbol) && bound ##
    AT(right integralSymbol) && expression).
```

The concatenation operator in the grammar is `&&`. The operator `##` means here that each of the three last lines is relative to the first one. The rule `expression` is not detailed here. The rules `integralSymbol` and `bound` consist in extracting terminals, thanks to the operator `TERM_CMP`. We present here the simplified rule for the detection of a bound:

```
bound::= TERM_CMP noCond character.
```

No condition is required here (`noCond`) for the detection of a bound; we could specify here a condition about the kind or the size of the component.

Mechanism of Neighbourhood. We have seen on the previous part that the joint use of a position operator, `AT`, and a component detection operator, `TERM_CMP`, was necessary to detect an element. We present here the associated mechanism of neighbourhood used by the analyser.

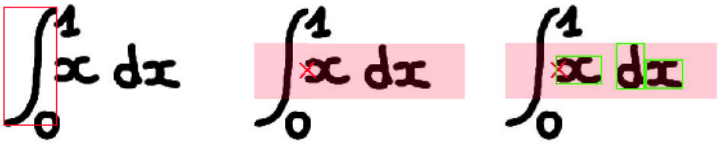
First, the operator `AT` makes it possible to choose a reference position (a point) and a research zone, depending on the last component found. Then, the operator `TERM_CMP` extracts a component:

- in the research zone;
- the one which bounding box is the nearest from the reference position;
- fulfilling the condition.

The example on figure 2 shows the application of the rules:

```
AT(right integralSymbol) && TERM_CMP noCond character.
```

In Fig.2(a), the `integralSymbol` has just been recognized, represented by his bounding box. The operator `AT` sets the reference position and the research zone corresponding to `right` of the `integralSymbol` bounding box (Fig.2(b)). Then, the instruction `TERM_CMP` makes the analyser look for the nearest component in the research zone, using the distance between reference point and bounding boxes (Fig.2(c)).



(a) Bounding box of integral

(b) Definition of the reference position and the research zone

(c) Detection of components depending on their bounding box

Fig. 2. Detection of the component x from the *integralSymbol*

3.3 Limits of this Version

This mechanism of neighbourhood is not always appropriate. Indeed, to build this neighbourhood, the elements are compared to their bounding box, which is quite vague in certain cases, and most particularly in handwritten documents.

Let us take the example of the previous grammar with another document, presented in Fig.3(a). With the same detection mechanism, the research zone and the reference pointer are set like in Fig.3(b), and the nearest component that will be detected is the d instead of the x. This is due to overlapping bounding boxes of components.

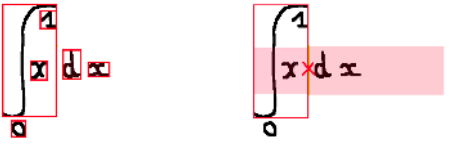
That is why we proposed to include new operators for this grammar, based on the use of the neighbourhood graph that has been presented in Sect.2.3. In this graph, the relative position of two graphical components is then given by the existence of an edge in the graph and the associated distance, which is more precise than a relative position of bounding boxes.

4 Integration of Neighbourhood Graph in DMOS

The work has consisted in inserting, in the existing formalism, interesting data that could be extracted from the graph.

We propose a new component detection mechanism and associated basic conditions based on neighbourhood graph.

Then, as DMOS makes it possible to analyse documents locally, we propose to exploit information based on the graph, depending on the context. Thus, we have set up a bi-directional communication between the analyser and the neighbourhood graph.



(a) Bounding boxes (b) Definition of the reference position and the research zone

Fig. 3. Detection of the component x from the *integralSymbol*

4.1 New Component Detection Mechanism

The first part consists in replacing the mechanism of component detection, presented in Sect.3.2, by a new one, based on a neighbourhood graph.

New Operator. We introduce a new operator, `TERM_CMP_GRAPH`, that can be used in the same conditions as `TERM_CMP`, but which mechanism is based on the neighbourhood graph. The grammar rule presented in Sect.3.2 becomes:

```
AT(right integralSymbol) && TERM_CMP_GRAPH noCond character.
```

When executing the instructions, the analyser leans on neighbourhood graph. Indeed, the operator `AT` set the research zone, like previously. However, the reference position is a component instead of a point in the previous version: the `integralSymbol` element is memorized as reference component.

With the `TERM_CMP_GRAPH` operator, we can detect an element:

- in the research zone;
- the nearest in the neighbourhood graph to the reference component;
- respecting the conditions.

New Conditions for the Detection. We introduce new conditions on required elements based on neighbourhood graph. Two examples are given.

Edge between Components. This condition assumes that the chosen element is linked with the reference component in the neighbourhood graph.

```
condExistDirectLink ReferenceComponent ComponentToAnalyse
```

Edge Distance. This condition limits the distance between two components.

```
condMaxDistance ReferenceComponent MaxDistance ComponentToAnalyse
```

succeeds if distance between `ReferenceComponent` and `ComponentToAnalyse` is inferior to `MaxDistance`.

These are just examples of the most common used conditions. However, the user can develop specific ones when necessary.

4.2 Extraction of Local Statistical Information

Thanks to symbolic knowledge contained in DMOS analyser, the context of analysis is always known. Consequently, the symbolic level can ask numerical information depending on a local context. Thus, in a context given area, we can examine only the corresponding part of the neighbourhood graph. Consequently, Voronoï tessellation can be exploited depending on the context and not necessarily on homogeneous document.

We propose to study local statistics about distances in the neighbourhood graph, thanks to the operator:

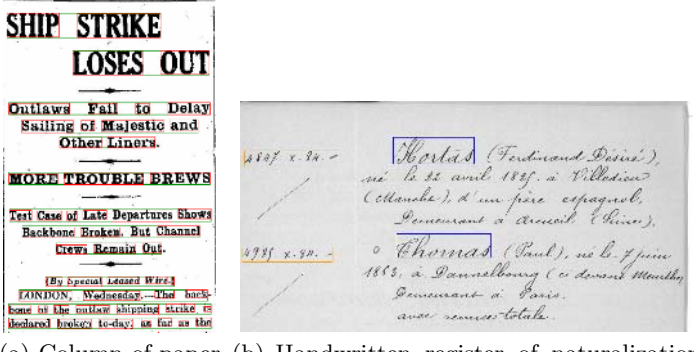
```
calculateStatDistGraph Area FavoriteDirection RequiredStatistic
```

In the selected zone `Area`, we extract distances of a chosen set of edges, depending on `FavoriteDirection`: every edge included in the zone, only the vertical or horizontal ones. Then, we calculate the chosen statistic `RequiredStatistic` that can be average, median or threshold in order to separate data into classes.

5 Application of Neighbourhood Graph Integration

These different tools have been applied for the description of various kinds of documents. The aim was to prove their genericity and to determine the cases the neighbourhood graph could be useful.

We used the statistic tools for the detection of words in printed papers and handwritten registers and studied a definition of a grammar using both bounding boxes and neighbourhood graph on handwritten register of the 19th century.



(a) Column of paper (b) Handwritten register of naturalization decree

Fig. 4. Application on two kinds of documents

5.1 Local Analysis

We applied the statistic tool to express a word and line recognition grammar. The aim was not to provide a good word detection system, and our mechanism could be improved according to the kind of document. However, its application on both printed and handwritten documents points out the interest of a *local* analysis and of the *Voronoï based* neighbourhood.

Principle of Word Detection. We consider that a line is a succession of words and a word a succession of letters, from left to right, linked in the neighbourhood graph. A distance threshold must distinguish the inter-word and inter-line gaps to express whether successive letters belong to the same word. We calculate this threshold line after line, thanks to the operator presented in Sect.4.2.

The work is split up into three parts:

1. The approximate area `Area` that contains the line is detected.
2. The threshold is extracted. We study each edge contained in the area and we ask for a threshold separating distances into 2 classes thanks to the k-average method:

   ```
   calculateStatDistGraph Area EveryLink KAverage
   ```

3. The words are extracted, thanks to the application of the rules below and taking the calculated threshold into account:

   ```
   word Threshold::=
       firstLetter MyLetter &&
       AT(rigthLine MyLetter) &&
       endOfWord Threshold MyLetter.
   endOfWord Threshold LastLetter::=
   letterOfWord Threshold LastLetter ThisLetter &&
       AT(rightLine ThisLetter) &&
       endOfWord Threshold ThisLetter.
   endOfWord _ _. %Stop case
   ```

   The recurrence ends when the next letter is too far from the previous one or when there is no more letters on the line. The detection of terminals is given by the rules below:

   ```
   firstLetter MyLetter::= TERM_CMP_GRAPH noCond MyLetter.
   letterOfWord Threshold LastLetter ThisLetter::=
       TERM_CMP_GRAPH [(condExistDirectLink LastLetter),
              (condMaxDistance LastLetter Threshold)], ThisLetter.
   ```

Interest of Local Analysis. We applied this grammar on columns of papers extracted from *International Herald Tribune* of years 1900, 1925 and 1950; those documents had been proposed for a contest at ICDAR 2001. Our base was composed of 2588 words to recognize; we managed to detect 98.53% of them.

We can see on the example presented in Fig.4(a) the interest of a local analysis, because of the large variation in police size. Indeed, with a global threshold determination, we could not find a convenient value for both title and text. In

our application, we have chosen to extract a threshold for each line, and to treat differently each text size.

The only reason for the remaining 2.47% undetected words is the case of one-word-composed lines where each letter is considered as a word. In order to solve this problem, we could extend the threshold extraction to the whole paragraph with the same character size. This could be done easily thanks to DMOS method.

Interest of Voronoï Based Neighbourhood. We applied this graph-based grammar on handwritten registers from the 19th century (Fig.4(b)). Our base was composed of 521 handwritten words on 111 lines, represented by their englobing shape.

In order to show the interest of Voronoï graph in comparison to bounding box neighbourhoods, we implemented another grammar, based on the same mechanism of words and thresholds, but with a bounding-box-based neighbourhood. We show that this grammar is less precise, especially with handwritten documents.

We consider only words that are found with a precision of 95% of the surface of their englobing shape. With Voronoï neighbourhood, 62.6% of words are recognized with this precision, whereas only 48.6% with bounding boxes. Moreover, bounding boxes detect lot of noise, because only 27.0% of detected words correspond to an expected one, whereas 54.2% of words detected with Voronoï method are interesting.

As we said previously, the aim was not to obtain a perfect word detection but to show that, with the same mechanism of thresholds between words, Voronoi-based mechanism was more precise than bounding-box neighbourhood.

5.2 Global Structure Recognition

The new operators have been implemented fulfilling the language genericity. Consequently, it makes it possible to use in a grammar either the bounding box based distances or the Voronoï tessellation neighbourhood, and thus to combine both neighbourhood in a document description. Indeed, the neighbourhood graph is useful for local analysis, when precise positioning between two components is required. However, global analysis is more efficient with bounding boxes.

Handwritten Register Structure Recognition. An example of such a combination is given with the description of handwritten registers of naturalization decree from the end of the 19th century (example in Fig.4(b)). A previous EPF grammar, presented in [3], made it possible to extract the columns, the registration numbers and the names from such a document. Indeed, we wanted to detect the numbers, in the margin area, and the names, fronting the numbers, in the body of the text. We have modified this grammar in order to introduce neighbourhood graph in relevant cases.

Alignment Detection. The detection of the document's margin, that is to say vertical alignments of characters, is based on a global research. That's why, for this global phase of the detection, we have chosen to keep using the bounding box distances.

Fig. 5. Example of overlapping bounding boxes

Number Detection. The detection of numbers consists in finding at least three horizontally aligned components in the margin area. With bounding boxes, the detection of each component is imprecise, especially when the bounding boxes are overlapping. The introduction of neighbourhood graph makes it possible to describe a number as a succession of characters, linked with an edge in the graph. Thus, we can detect each component contained in the number and overcome the difficulty of bounding boxes.

For example, in Fig.5.2, the bounding boxes of the left and right parts of the N are totally overlapping horizontally. With the standard version of the grammar, the right part of the N was not detected. With the new one, based on neighbourhood graph, we can detect each component.

Surname Detection. Concerning surnames, the previous grammar was detecting a global line of text, made of at least five aligned components. The surname was supposed to be contained in the first half of the text line. Nevertheless, this is approximate, because when the surname is very short, the analysis returns a lot of noise, whereas a long surname will be cut.

In order to improve this detection, we propose to introduce the previous word-detecting grammar (see Sect.5.1). Surname is composed of the two first words of the line, extracted thanks to local statistic information. This method is globally better for detecting the surname but the difficulty is to know how many words are contained in the surname, from one to three depending on cases.

Results. This grammar has been applied on 1130 naturalization decree pages from the end of the 19th century, which represents 3785 registration numbers and their associated surnames. The global recognition rate of number and surname areas is similar to the one corresponding to the previous grammar, that is to say around 99,02%. However, the extracted elements inside the number and surname areas are more precise, because we can detect each component of the number and only the first words for the name.

The important point was to validate the joint use of neighbourhoods, and to validate the extraction of numerical information from the neighbourhood graph, depending on the context.

6 Discussion

6.1 Interests for the Formalism

Compared with the standard version of DMOS, the introduction of a neighbourhood graph brings new faculties of expression of the knowledge. The operator

`TERM_CMP_GRAPH` compensates for the imprecision of bounding box based distance, giving precise information about the existence of a neighbourhood between two components, and their distance.

When defining a new grammar, the user keeps the possibility to use either the bounding box neighbourhood or the Voronoï based graph. Information contained in neighbourhood graph describes local relation between two components; their exploitation is really convenient to detect close components, when a precise relative position is required. This is particularly adapted for handwritten document analysis, liable to overlapping bounding box problems.

In return, in a global study of the document, like detecting margins for example, the neighbourhood graph doesn't seem to bring pertinent information. The bounding box neighbourhood still seems more relevant in that case.

6.2 A Contextual Utilization of Voronoï Diagram

The main particularity of the exploitation of Voronoï diagram is that the data contained in the graph is separated from the knowledge. This gives mainly one advantage: the grammar-based description of the kind of document makes it possible to exploit data contained in the graph according to the context of analysis. It means that, depending on the circumstances of the analysis, the user can, on one hand, choose which information should be extracted from the graph, and on the other hand, determine how this characteristic should be interpreted into symbolic information. This makes it possible to extract more information from Voronoï diagram than in classical applications.

6.3 Possible Evolutions

In this version of our work, neighbourhood graph makes it possible to position only two components. It would be sometimes interesting to know the relative position of two groups of components. For example, once a line of text has been detected as a set of components, we could gather those elements in order to be able to position a line relatively to another. This would require a hierarchical structure in order to make the graph evolve during the analysis. This evolution could bring new interesting information for a global document analysis.

7 Conclusion

This paper shows how we have extended the exploitation of a neighbourhood graph based on Voronoï tessellation, by separating the graph analysis from the expression of the necessary knowledge, thanks to the generic method DMOS.

The standard relative position mechanism used in DMOS, based on bounding boxes, was not precise enough in certain cases. That is why we have introduced a new mechanism based on Voronoï tessellation, especially convenient for overlapping bounding-box sensitive handwritten documents.

Voronoï tessellation is usually used globally on a document, which reduces its capacity of exploitation. Thanks to method DMOS, this graph can be exploited

depending on the context. Consequently, the data extracted from the neighbourhood graph can be more complete and adapted to required information.

The neighbourhood graph can be used for the description of any kind of document. For example, we applied this work on two kinds of documents: printed newspapers and handwritten naturalization decrees.

References

1. Bertrand Coüasnon. DMOS: A generic document recognition method to application to an automatic generator of musical scores, mathematical formulae and table structures recognition systems. International Conference on Document Analysis. (ICDAR'01), pages 215-220, 2001.
2. Bertrand Coüasnon. Dealing with noise in DMOS, a generic method for structured document recognition: an example on a complete grammar. Graphics Recognition: Recent Advances and Perspectives, pages 38-49, 2004.
3. Bertrand Coüasnon, Jean Camillerapp, and Ivan Leplumey. Making Handwritten Archives Documents accessible to Public with a Generic System of Document Image Analysis. International Workshop on Document Image Analysis for Libraries (DIAL'04), Pages 270-277, 2004.
4. Koichi Kise, Motoi Iwata, and Keinosuke Matsumoto. On the application of Voronoï diagrams to page segmentation. Document Layout Interpretation and its Application (DLIA'99), 1999.
5. Ivan Leplumey and Charles Queguiner. Un graphe de voisinage bas sur l'utilisation des distances discrtes. Confrence Internationale Francophone sur l'Ecrit et le Document (CIFED'2002), pages 41-50, 2002.
6. Yue Lu, Zhe Wang, and Chew Lim Tan. Word grouping in document images based on Voronoï tessellation. Workshop on Document Analysis Systems (DAS'04), 2004.
7. Yue Lu, Chew Lim Tan. Constructing Area Voronoï Diagram in Document Images. International Conference on Document Analysis. (ICDAR'05), pages 342-346, 2005.
8. Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *S*patial Tessellations: Concepts and Applications of Voronoï Diagrams. Wiley, 2000.

Primitive Segmentation in Old Handwritten Music Scores*

Alicia Fornés, Josep Lladós, and Gemma Sánchez

Computer Vision Center/Computer Science Department,
Edifici O, Campus UAB 08193 Bellaterra (Cerdanyola), Barcelona, Spain
{afornes, josep, gemma}@cvc.uab.es
http://www.cvc.uab.es

Abstract. Optical Music Recognition consists in the identification of music information from images of scores. In this paper, we propose a method for the early stages of the recognition: segmentation of staff lines and graphical primitives in handwritten scores. After introducing our work with modern musical scores (where projections and Hough Transform are effectively used), an approach to deal with ancient handwritten scores is exposed. The recognition of such these old scores is more difficult due to paper degradation and the lack of a standard in musical notation. Our method has been tested with several scores of 19th century with high performance rates.

1 Introduction

The aim of Optical Music Recognition (OMR) is the identification of music information from images of scores and its conversion into a machine legible format. This process allows the development of a wide variety of applications: edition and publication of scores never edited, renewal of old scores, conversion of scores into Braille code, creation of collecting databases to perform musicological analysis and finally, production of audio files or musical description files: NIFF (Notation Interchange File Format) and MIDI (Music International Device Interface).

Although OMR has many similarities with Optical Character Recognition (in fact OCR is a sub-task of OMR because lots of scores include text), OMR requires the understanding of two-dimensional relationships. It is nevertheless true that music scores follow strict structural rules that can be formalized by grammar rules, so context information can be extracted helping in the recognition process. A survey of classical OMR (from 1966 to 1990) can be found in [1], where several methods to segment and recognize symbols are reviewed: the detection of staff lines is performed using projections, a line adjacency graph, slicing techniques, comparing line angle and thickness. Extraction and classification of musical symbols is performed using projections, classifiers based on decision trees, matching methods and contour tracking properties. Finally, validation of scores is usually done using grammars.

* This work has been partially supported by the Spanish project CICYT TIC 2003-09291.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 279–290, 2006.
© Springer-Verlag Berlin Heidelberg 2006

OMR is a mature area for printed scores, however our work is focused on the recognition of handwritten ones: we propose a method to detect primitives in modern and old handwritten scores. In modern ones, the detection of staff lines is performed using Hough Transform and projections, whereas in old scores, a contour tracking process is required to cope with deviations in staff. Concerning graphical primitive detection, we propose similar approaches either for modern and old scores: morphological operations, Hough Transform and median filters.

This paper is organized as follows: In section 2 the structure of scores and layers of the system are shown. In section 3 our work with modern handwritten scores is presented, whereas our approach to the segmentation and classification of primitive score elements in old handwritten scores is described in section 4. In section 5 some illustrative experimental results are reported. Finally, in section 6 the concluding remarks are exposed.

2 Handwritten Scores: Structure and Layers

Whereas there is a lot of literature about the recognition of printed scores, few research works have been done in handwritten ones [2, 3]. Regarding printed ones, handwritten scores introduce additional difficulties in the segmentation and the recognition process: notation varies from writer to writer, symbols are written with different sizes, shapes and intensities; the number of touching and broken symbols increases significantly.

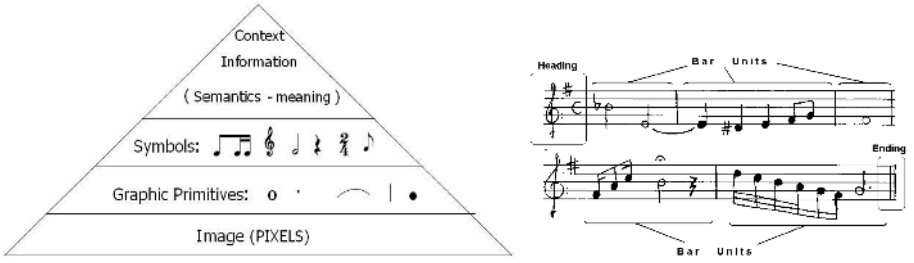
According to the approach proposed by Kato [4], an OMR system has several layers, corresponding to the abstraction levels of the processed information, see Fig. 1(a): the image layer is formed by pixels; the graphical primitive layer is formed by dots, lines, circles and curves. In the symbol layer, graphical primitives are combined to form musical symbols. In the semantic-meaning layer information, the pitch and the beat of every note is obtained, and grammar rules are used to validate it and solve ambiguities. Feedback among layers is extremely important because each level contains hypothesis of various levels of abstraction, so, if an upper layer rejects a result produced from lower layers (e.g. a certain object is not what it has been determined to be), the system must be able to correct this error and classify the object again.

The musical notation in scores consists of the following elements: staffs (when musical symbols are written down), attributive symbols at the beginning (clef, time and key signature), bar lines (that separate every bar unit) that include rests and notes (composed of head notes, beams, stems, flags and accidentals); and finally, slurs and dynamic markings. Some scores include text, so an important task is to determine which objects are text (lyrics), and which are musical symbols. In addition, some words correspond to dynamic markings, so context information should help to distinguish them.

Formal language theory provides useful tools to recognize and solve ambiguities in terms of context-based rules or semantic restrictions using attributes. Grammars are usually used to describe the score structure, see Fig. 1(b). Therefore, parsers guide the recognition and validation process. Informally speaking,

a grammar describing a score consists of three blocks **G: S** $\rightarrow$ **H[B]E**, where **H** is the heading with the attribute symbols. Then, the score is discomposed in bar units **B**. The end of the score is marked with an ending measure bar (**E**).

Our recognition strategy follows a typical OMR architecture: After preprocessing the image, a segmentation process extracts graphical primitives; then recognition and classification of musical symbols is performed. Finally, a semantic layer uses context information to validate it and solve ambiguities.



(a) (b)

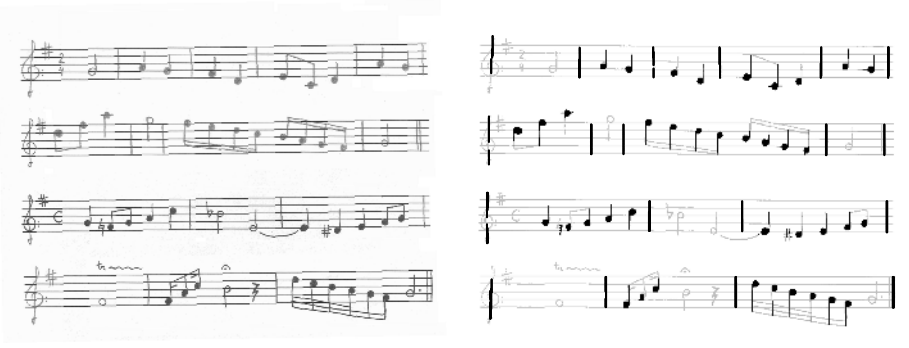
Fig. 1. (a) Levels. (b) Structure of a score.

The early-level stages are described in this paper: segmentation of score blocks and detection of primitives. As we have said before, most segmentation problems are due to distortions caused by staff lines, broken and touching symbols as well as high density of symbols. For this reason, deleting staff lines and isolating symbols are the first tasks to cope with.

3 Modern Handwritten Scores

Initially, we have been working with modern musical scores, where paper is in good condition, there is a standard of musical notation and most of staff lines are printed. Here, the approach proposed consists in the following: First, the input image (at a resolution of 300 dpi) is binarized(using the Otsu method) and deskewed (using Hough Transform to detect staff lines). After that, horizontal projections can effectively be used to detect rows likely to contain a staff line. In the staff analysis some parameters are set: width of staff lines and distance between them. Knowing these parameters, a run-length smearing process deletes staff lines trying to keep complete symbols. Finally, morphological operations reduce noise.

Concerning the primitive detection stage, vertical lines and head notes are the first graphical primitives to recognize: detection of vertical lines is also performed using the Hough Transform (allowing a skew of 20 degrees). Then, they are classified in beams (which have headnotes), bar lines (longer than beams, without headnotes and divide scores in bar units) and others (e.g. lines that are part of another kind of symbols).



(a) (b)

Fig. 2. (a) Original Image. (b) Graphical primitives detected.

Detection of filled headnotes is performed with a morphological opening (with a disk of *radius = w/3*, where w is the distance between staff lines) and using parameters of circularity, area and compactness.

Extraction of whole and half notes are more difficult because handwritten circles are often broken or incomplete, so morphological operations cause a lot of false positives and further work is required. After that, the remaining image is processed to obtain other graphic primitives.

Figure 2(a) shows the original and skewed image. Using Hough Transform the orientation of the image is detected and corrected. Thus, horizontal projections show every staff line as a maximum, and staff lines can be deleted. In Fig. 2(b) we can see the detection of graphical primitives: headnotes and vertical lines are in black color, and bar lines are shown as the thickest vertical lines. The remaining score is in grey color (staff lines are not actually present, but in this figure they are shown on purpose). As we can see, good results are achieved.

4 Old Handwritten Scores

A growing interest in the Document Analysis area is the recognition of ancient manuscripts and their conversion to digital libraries, towards the preservation of cultural heritage. Our work is focused on the recognition of old handwritten scores (18th-20th centuries) so that these scores of unknown composers could be edited and published (contributing to the preservation and dissemination of artistic and cultural heritage). Working with old scores makes the task more difficult because of paper degradation (most scores are in poor condition) and the lack of a standard notation. In addition, there are distortions caused by staff lines (which are often handwritten), broken and touching symbols as well as high density of symbols. In order to cope with these problems, an expert system will be required to learn every new way of writing, and artificial intelligence based techniques will take advantage of higher level musical information. In the following sections, the method proposed to detect and extract staff lines and graphical primitives is exposed (see the followed steps in Fig. 3).

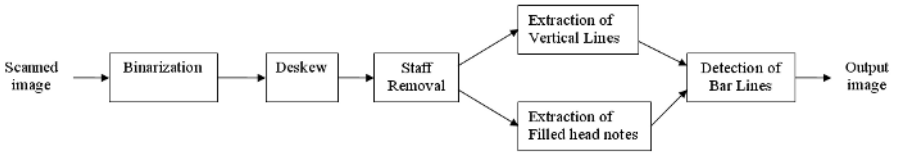


Fig. 3. Preprocessing Stages of the system

4.1 Extraction of Staff Lines

The preprocessing and segmentation phases must be adapted to this kind of scores: First of all, global binarization techniques do not work because of degradation of the scores; so adaptive binarization techniques are required (such as Niblack binarization [5]). Secondly, the detection of staff lines is more difficult due to distortions in staff (lines often present gaps in between), and contrary to modern scores, staff lines are rarely perfectly horizontal. This is caused by the degradation of old paper, the warping effect and the inherent distortion of handwritten strokes (staff lines are often drawn by hand). For those reasons, a more sophisticated process is followed (see Fig. 4).

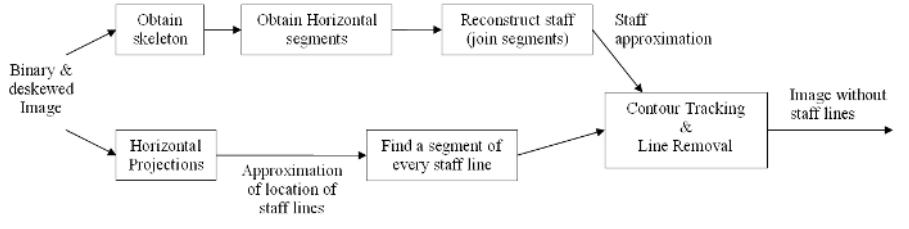
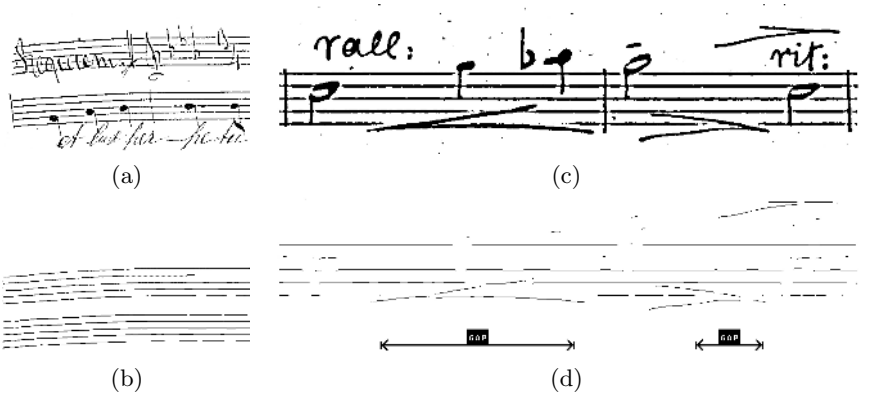


Fig. 4. Stages of the extraction of staff lines

Here, Hough Transform is only used to deskew the input image, so horizontal projections can obtain a rough approximation of the location of staff lines. Then, horizontal runs are used as seeds to detect a segment of every staff line, and a contour tracking process is performed in both directions following the best fit path according to a given direction. In order to avoid deviations (wrong paths) in the contour tracking process, a coarse staff approximation needs to be consulted.

The steps applied to obtain an image with horizontal segments (which will be candidates to form staff lines) are: First, the skeleton of the image is obtained and a median filter is applied with a horizontal mask. This process is repeated until stability (last two images are similar). In the output image, only staff lines and those horizontally-shaped symbols will remain. Notice that in a binary image, a median filter puts a pixel to 0 if most pixels of the neighborhood are 0, otherwise, its value will be 1. The size of this horizontal mask is constant (experimentally, dimensions are set to 1×9 pixels), because in the skeletonized image, each line is one pixel-width, so the width of lines in the original image is irrelevant.



(a) (c)

(b) (d)

Fig. 5. (a) Original Image. (b) Reconstruction of staff lines. (c) Original Image. (d) Line segments of staff lines with gaps and horizontal symbols.

In order to reconstruct the stave lines, each segment is discarded or joined with others according to its orientation, distance and area. Fig. 5(a) shows an original score suffering from a warping effect and its staff reconstruction (Fig. 5(b)). If there are big gaps in staff lines in presence of horizonal symbols this method could fail and follow a segment of this symbol instead of a segment of the staff line. Fig. 5(c) shows a big gap with a crescendo marking and Fig. 5(d) shows its reconstruction. An initial solution to it consists in increasing the size of the slice, but it could not work in scores with large deviations in staff lines.

Once we have obtained the reconstructed staff lines, the contour tracking process can be performed following the best fit path according to a given direction. If there is no presence of staff line (a gap), the process will be able to continue according to the location of the reconstructed staff line.

Concerning line removal, we must decide which line segments can be deleted from the image, because if we delete staff lines in a carelessly way, most symbols will become broken. For that reason, only those segments of lines whose width is under a certain threshold (depending on the width of staff lines, calculated using the statistical mode of line-segments) will be removed. Fig. 6 shows some examples of line removal: Fig. 6(b) is the original image, and in Fig. 6(a) we can see how in presence of a gap, the process can detect next segment of staff line to continue; in Fig. 6(c) a symbol crossing the line will keep unbroken, because the width of the segment is over the threshold. In this level of recognition, it is almost impossible to avoid the deletion of segments of symbols that overwrite part of a staff line (they are tangent to staff line, see Fig. 6(d) and whose width is under this threshold, because context information is not still available.

4.2 Recognition of Vertical Lines

After deleting staff and calculating the distance between stave lines, vertical lines and head notes are the first graphical primitives to recognize. First, some morphological operations and run length smearing techniques are used to reduce

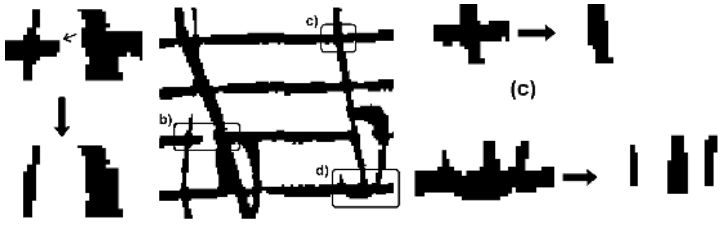


Fig. 6. Examples of Line Removal in Contour Tracking process. a) Gap in line, b) Original Image, c) Symbol crosses the staff line, d) Symbol is tangent to staff line: Symbol becomes broken.

noise. Afterwards, we use median filters with a vertical structuring element, so only symbols with vertical shape will remain (see Fig. 7(a)). Contrary to extraction of staff lines, here the size of the structuring element depends on the distance between staff lines. We have also tested Hough Transform to detect vertical lines (as we do in modern scores), but results using median filters are better and the algorithm is faster.

4.3 Recognition of Filled Head Notes

Working with printed scores makes this process easier, because all headnotes have similar shape. A morphological opening operation (with a circular structuring element), and choosing the ones with adequate circularity and area, does not work with handwritten scores, because there is too much variability in ways of writing to perform a process that detects exactly all head notes.

The method proposed performs a morphological opening with elliptical structuring element (whose size depends on the distance between staff lines), oriented 30 degrees, discarding elements with large area. This approach gets all filled headnotes and false positives (Fig. 7(b)), but it is better to discard false positives in next stages than loosing real head notes. Because some modern rules of musical notation are not applied in old scores, we will classify notes (filled headnotes with beams) in higher-level stages, using grammar rules and the knowledge of time signature.

4.4 Recognition of Bar Lines

Once we have detected vertical lines and filled head notes, lines must be classified (see Fig. 8(a)) in beams (which have headnotes), bar lines (longer than beams, without headnotes) and others (e.g. lines that are part of another kind of symbols). Bar lines are the most important vertical lines, because they divide scores in bar units. Once we have isolated every bar unit, we can process them in an independent way, looking for musical symbols using grammar rules.

A first approximation of bar lines is performed assuming that bar lines cover all staff and there are no headnotes in their extremes. So, if a vertical line is large enough and it is situated covering all five staff lines, then it will labelled as a bar line if there is no presence of filled headnotes in its extremes, see Fig. 8(b).

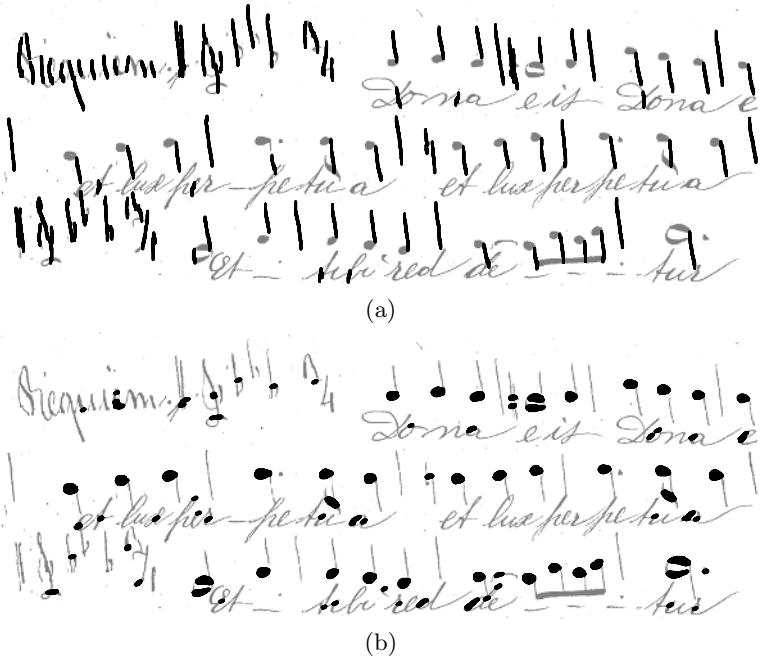


Fig. 7. A section of the Requiem Mass of the composer Aleix: (a) Vertical lines detected are in black color. (b) Filled head notes detected in black color.

4.5 Classification of Clefs

Once every measure of the score is obtained, it is processed independently in order to recognize and classify all musical symbols. The heading of every score is formed of the clef, time signature and key signature. Because the clef determines the pitch of every note, it should be one of the first elements to recognize.

Due to the enormous variations in handwritten clefs, the classification of clefs must cope with deformations and variations in writing style. Thus, the method proposed uses Zernike moments (which maintain properties of the shape, being invariant in front of deformations) and Zoning, which codifies shapes based in statistical distribution of points in a compact and easy way. A full description of these techniques can be found in [7].

Zoning consists in computing the percentage of foreground pixels in each zone: an *mxn* grid is superimposed on the character image, and for each of the *nxm* zones, the average gray level is computed, giving a feature vector of length *nxm*. Thanks to the fact that in bass clefs the top of the clef has the bigger area, the Zoning algorithm can be used for a initial classification of bass clefs: The image is divided in 3 rows and 1 column, and the zoning vector (3x1) is filled with its normalized area. If the first row has the biggest area of the vector (see squares in white color in Fig. 9(a)), then the clef is a bass clef. Afterwards, clefs not classified with Zoning will be classified using Zernike Moments.

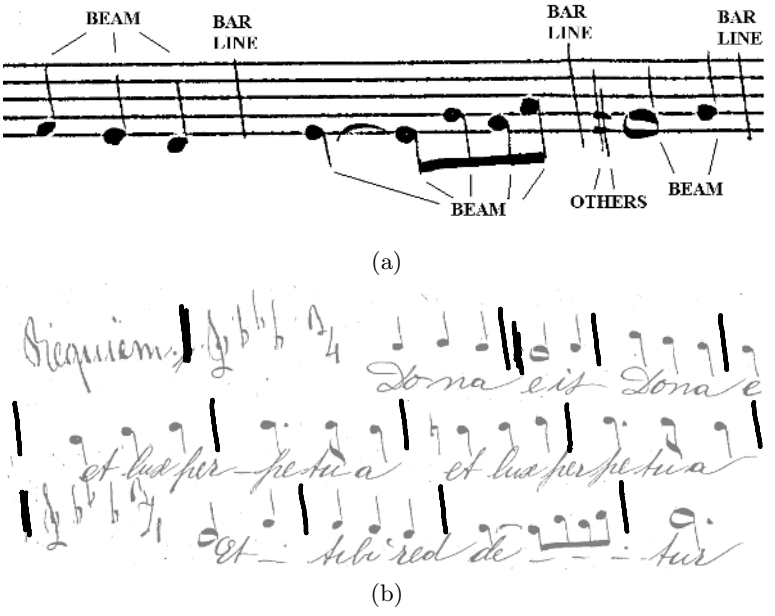


Fig. 8. (a) Verticals in scores. (b) Bar lines in black color.

Zernike moments are defined over a set of complex polynomials which forms a complete orthogonal set over the unit disk.

Polynomials of Zernike are denoted by:

$$ZP = \{V_{nm}(x, y) | x^2 + y^2 \leq 1\} \tag{1}$$

The form of the Zernike polynomial basis of order n an repetition m ($n \in N^+$, $m \in N, (n - |m|)$ even, and $|m| \leq n$) and the radial polynomial are defined as:

$$V_{nm}(x, y) = R_{nm}(x, y) \exp(jm \arctan(y/x)); \tag{2}$$

$$R_{nm}(x, y) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s!(\frac{n+|m|}{2} - s)! \cdot (\frac{n-|m|}{2} - s)!} \cdot (x^2 + y^2)^{(n-2s)/2} \tag{3}$$

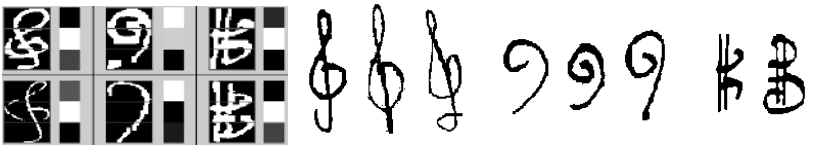


Fig. 9. (a) The application of Zoning technique to clefs using 3 files and 1 column to divide the images. (b) Clef Models for the classification using Zernike moments.

In our approach, 12 Zernike moments are used with 8 model classes for the three existing clefs (see Fig. 9(b)). The method normalizes the image of every model of the class and computes the Zernike moments and the feature vector. Afterwards, the Zernike moments and feature vector of the clef to be identified are computed. Then, the method will associate the new clef with the model class whose feature vector is closer to the feature vector of the clef to be classified.

5 Results

We have tested our method with a set of scores from the 19th century of several composers. These images of scores have been obtained through the archive

Table 1. Staff removal results: When lines are not perfectly reconstructed, it is impossible to reach rates of 100% in staff removal

Page	N. Staffs	Perfectly Reconstructed / Total, (%)	Perfectly Removed / Total, (%)
1	10	49 / 50 , 98%	48 / 50 , 96%
2	10	50 / 50 , 100%	50 / 50 , 100%
3	10	45 / 50 , 90%	45 / 50 , 90%
4	10	49 / 50 , 98%	48 / 50 , 90%
5	12	54 / 60 , 90%	53 / 60 , 88%
6	14	70 / 70 , 100%	70 / 70 , 100%
7	14	69 / 70 , 98%	69 / 70 , 98%

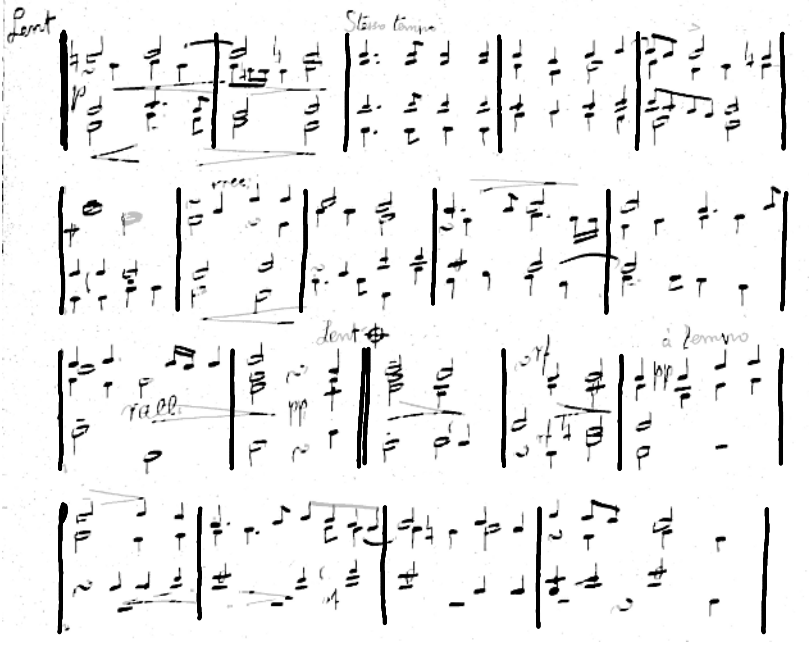


Fig. 10. Results from a section of "Salve Regina" of the composer Aichinger: Filled headnotes and beams in black color. Bar lines are the thickest lines.

Table 2. Graphical Primitive Recognition Results: 100% of Filled Headnotes, Vertical and Bar lines detected. Precision = Correct/Detected; FP= % of False Positives.

Page	N. Staffs	Verticals: C/D, (%FP)	Bar lines (%FP)	Head notes (%FP)
1	10	236 / 352 , 33%	71 / 80 , 11%	99 / 462 , 78%
2	10	177 / 237 , 25%	54 / 57 , 5%	96 / 465 , 79%
3	7	225 / 269 , 16%	40 / 43 , 7%	135 / 382 , 64%
4	7	218 / 284 , 23%	48 / 49 , 2%	128 / 365 , 65%
5	6	227 / 271 , 16%	38 / 41 , 7%	110 / 390 , 71%
6	6	180 / 254 , 29%	37 / 48 , 23%	122 /435 , 72%

of Seminar of Barcelona. Referring the staff removal stage and the graphical primitives detection, several pages of scores from different composers have been tested. In table 1 we can see that most staff lines are perfectly reconstructed, but sometimes a horizontal symbol is drawn over a staff line and causes the staff reconstruction to follow wrongly this symbol. More exhaustive results can be found in [8].

Concerning detection of graphical primitives (see an example in Fig. 10), in table 2 we can see that 100% of headnotes, vertical and bar lines are detected, although there is an important percentage of false positives (which will be detected in high-level layers). Performance in detection of filled headnotes decreases when strokes are very thick, so in such cases, other objects could also be detected as filled headnotes. Although the precision rates are low and there are many false positives, it is better to discard them in next stages than having false negatives (filled headnotes in thin strokes not detected). Finally, the classification of clefs reaches rates of 86% (44 clefs correctly classified of 51 existing clefs).

6 Conclusions

In this work an approach to segment primitive elements in handwritten old music scores has been presented. Our strategy consisted of the following steps: First, score line detection and removal, using Hough Transform and a line tracking algorithm. Then, the detection of vertical lines and circular primitives is performed. Finally, the classification of vertical lines and clefs is described.

We have obtained high performance rates in this primitive segmentation stage. False positives in the recognition process are due to the enormous variation in handwritten notation and the lack of a standard notation. Further work will be focused on extracting lyrics from the scores, improving the reconstruction of staff lines, obtaining other graphic primitives and formalizing a grammar to help in the classification of musical symbols.

Acknowledgements

We would like to thank Josep Maria Gregori Cifré from Art Department of UAB for his help in accessing to old resources of archive of Seminar of Barcelona.

References

1. D. Blostein, H. Baird, "A Critical Survey of Music Image Analysis," *Structured Document Image Analysis*, Eds. H. Baird, H. Bunke, and K. Yamamoto, Springer Verlag (1992), 405–434.
2. K.C. Ng, "Music Manuscript Tracing", *Proceedings of the Fourth IAPR International Workshop on Graphics Recognition (GREC)*, Kingston, Ontario, Canada (2001), 470–481.
3. J.C. Pinto, P. Vieira, J.M. Sosa, "A New Graph-like Classification Method Applied to Ancient Handwritten Musical Symbols", *International Journal of Document Analysis and Recognition (IJDAR)*, Vol. 6, Issue 1 (2003), 10–22.
4. H. Kato and S. Inokuchi, "The Recognition System for Printed Piano Music Using Musical Knowledge and Constraints". *Proceedings of the IAPR Workshop on Syntactic and Structural Pattern Recognition*, Murray Hill, New Jersey (1990).
5. W. Niblack, *An Introduction to Digital Image Processing*, Englewood Cliffs, Prentice Hall (1986), 115–116.
6. D. Bainbridge, N. Carter, "Automatic Reading of Music Notation", *Handbook of Character Recognition and Document Image Analysis*, eds. H.Bunke and P.S.P.Wang, World Scientific, Singapore (1997), 583–603.
7. Ø. D. Trier, "Goal-directed Evaluation of Binarization Methods", in Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(12), (1995).
8. A. Fornés, "Analysis of Old Handwritten Musical Scores", Master's Thesis, Universitat Autònoma de Barcelona, Spain (2005).

Generic Shape Classification for Retrieval*

Manuel J. Fonseca, Alfredo Ferreira**, and Joaquim A. Jorge

Department of Information Systems and Computer Science
INESC-ID/IST/Technical University of Lisbon
mjf@inesc-id.pt, alfredo.ferreira@inesc-id.pt, jorgej@acm.org

Abstract. We present a shape classification technique for structural content–based retrieval of two-dimensional vector drawings. Our method has two distinguishing features. For one, it relies on explicit hierarchical descriptions of drawing structure by means of spatial relationships and shape characterization. However, unlike other approaches which attempt rigid shape classification, our method relies on estimating the likeness of a given shape to a restricted set of simple forms. It yields for a given shape, a feature vector describing its geometric properties, which is invariant to scale, rotation and translation. This provides the advantage of being able to characterize arbitrary two–dimensional shapes with few restrictions. Moreover, our technique seemingly works well when compared to established methods for two dimensional shapes.

1 Introduction

Since shape is one of the primary low level features used in content-based image retrieval, shape representation has become a fundamental issue in these applications. The main objective of shape description is to measure geometric attributes of an object, that can be used for classifying, matching and recognizing objects. Moreover, a shape representation scheme should be affine invariant, robust, compact, easy to derive, easy to match and perceptually meaningful. Also, it is important that shape description schemes work well for practical applications. We have thus validated our work with two "real–life" applications, one to retrieve technical drawings and another to search for clip-art drawings. In this paper, after a short discussion of related work, we will briefly present our approach to drawing classification, consisting of topological and geometrical components. Then we focus on geometry extraction and describe our technique for shape classification. Next, we discuss experimental results obtained by comparing our method to five known techniques and briefly describe the two prototypes that use our method. Finally we present conclusions and future work.

* This work was funded in part by the Portuguese Foundation for Science and Technology, project POSC/EIA/59938/2004.

** Alfredo Ferreira was supported by the Portuguese Foundation for Science and Technology, grant reference SFRH/BD/17705/2004.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 291–299, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Related Work

There is an extensive body of related work on shape representation. Mehtre et al group existing techniques into two categories: boundary-based and region-based [17]. The former use only the contour or border of an object, which is crucial to human perception in judging shape similarity, completely ignoring its interior. The latter methods exploit shape interior information, besides its boundary. More recently Safar et al presented a taxonomy [22] that complements Mehtre's classification.

As examples of boundary-based methods we have Fourier descriptors [20], chain codes [11], autoregressive models [15], polygonal approximations [13], curvature scale space [18] and shape signature [3]. In region-based methods, we encountered geometric moments [14], Zernike moments [17], grid representation [16] and area.

Although contour-based methods such as Fourier descriptors, present good results in these studies, they have limited application. For one, these methods cannot capture shape interior content or deal with disjoint shapes, where single boundaries may not be available. Also, region-based methods can be applied to more general shapes, but usually require more computational resources.

3 Drawing Classification

Content-based retrieval of pictorial data, such as digital images, drawings or graphics, uses features extracted from the corresponding picture. Typically, two kinds of features are used; visual features (such as color, texture and shape) and relationship features (topological and spatial relationships among objects in a picture). However, in the context of our work, we consider that color and texture are irrelevant features and we focus only on topology (a global feature of drawings) and geometry (a local feature).

Our feature extraction technique processes drawings via two separate stages (topology and shape) until they are mapped into geometric and topological descriptors, as depicted in Figure 1. For retrieval purposes, these descriptors may be inserted in an indexing structure, during classification, or used to query a database, when searching for similar drawings.

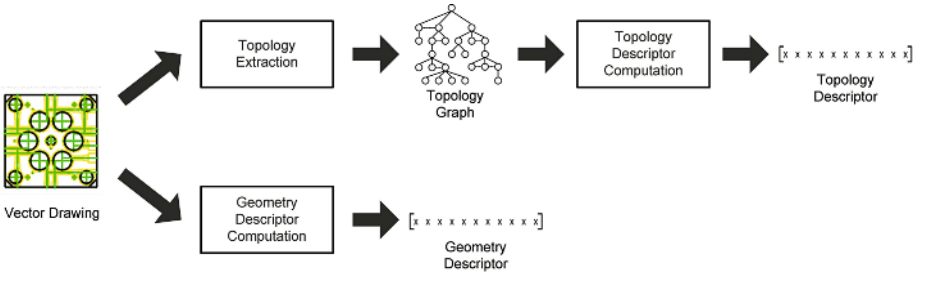


Fig. 1. Block decomposition of our approach to drawing classification

To describe the spatial organization in drawings, we use two relationships, inclusion and adjacency. While these two topological relationships are weakly discriminating, they do not change with rotation and translation, allowing unconstrained drawing classification. We then construct a topology graph representing the relationships among shapes. From this graph, we derive descriptors based on its spectrum [2]. We compute the graph spectrum by determining the eigenvalues of its adjacency matrix. Eigenvalues are then stored in a multidimensional vector, defining the topological descriptor. A detailed description of the topology extraction and the correspondent descriptor computation using eigenvalues can be found in [4].

4 Geometry Extraction

To describe the geometry of entities from drawings, we developed a general, simple, fast, and robust recognition approach called CALI [8, 7]. This was initially devised for recognition in calligraphic interfaces. However, since CALI performed well in recognizing hand-drawn input, we decided to generalize that approach by using it to classify more general shapes for retrieval. Thus, instead of using CALI to identify specific shapes or gestures from sketches, we compute a set of geometric attributes from which we derive features such as area and perimeter ratios from special polygons and store them in a multidimensional vector (see Figure 2). Indeed, our approach can be thought as a two–stage process. First, we evaluate a shape's geometric characteristics. Then we convert these

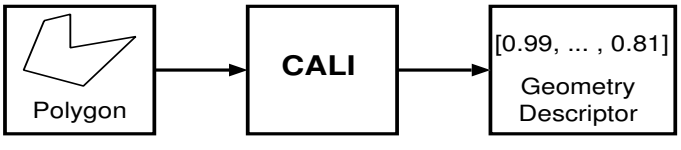


Fig. 2. Block diagram for computing the geometric descriptor

Table 1. List of relevant geometrical features

Feature	*Description*
A_{ch}	Area of the convex hull
A_{er}	Area of the (non-aligned) enclosing rectangle
A_{lq}	Area of the largest quadrilateral
A_{lt}	Area of the largest triangle
H_{er}	Height of the (non-aligned) enclosing rectangle
P_{ch}	Perimeter of the convex hull
P_{er}	Perimeter of the enclosing rectangle
P_{lq}	Perimeter of the largest quadrilateral
P_{lt}	Perimeter of the largest triangle
T_l	Total length, *i.e.* perimeter of original polygon
W_{er}	Width of the (non-aligned) enclosing rectangle

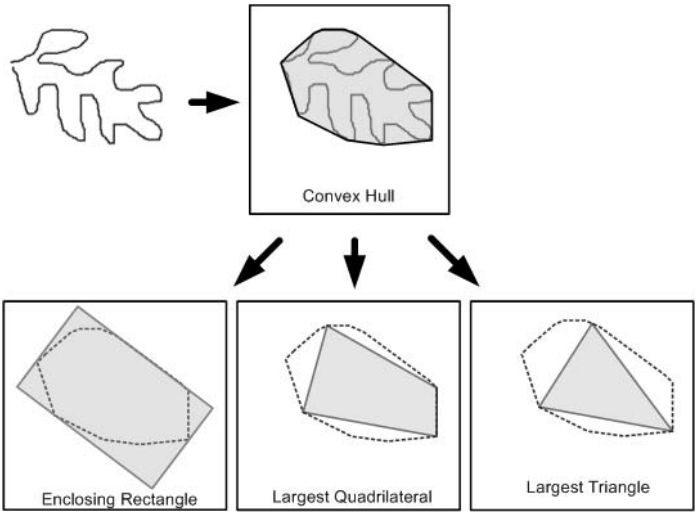


Fig. 3. Special polygons computed from shape

into affine–invariant geometric features by simple arithmetic operations which combine these attributes with known commensurable values for simple convex primitives, such as quadrilaterals and triangles. What is more important, using geometric features instead of polygon classification, allows us to index and store potentially unlimited families of shapes in a scalable manner.

Our geometric description method uses a set of global geometric properties extracted from drawing entities. We start the calculation of geometric features by computing the *Convex Hull* of the provided element, using Graham's scan [19]. Then, we compute three special polygons from the convex hull: the *Largest Area Triangle* and the *Largest Area Quadrilateral* inscribed in the convex hull [1], and finally, the *Smallest Area Enclosing Rectangle* [12]. Figure 3 depicts an example of polygons extracted from a irregular shape.

Finally, we compute the ratios between area and perimeter from each special polygon. We experimentally evaluated several ratios, as described in detail in [9], before we reach the set of features listed in Table 1. This set of features allow the description of shapes independently of their size, rotation, translation or line type. This way, such features can be used to classify either drawings or hand-sketched queries. Then, we combine these geometric features to produce a feature vector that describes the shape (descriptor).

Figure 4 shows the geometric features that compose the feature vector. To decide whether two shapes are similar we just compare (e.g. using dot–product)

$$\left[\frac{P_{ch}}{T_l} \; \frac{A_{ch}}{P_{ch}^2} \; \frac{H_{er}}{W_{er}} \; \frac{A_{lq}}{A_{er}} \; \frac{A_{ch}}{A_{er}} \; \frac{A_{lq}}{A_{ch}} \; \frac{A_{lt}}{A_{lq}} \; \frac{A_{lt}}{A_{ch}} \; \frac{P_{lq}}{P_{ch}} \; \frac{P_{lt}}{P_{ch}} \; \frac{P_{ch}}{P_{er}} \right]$$

Fig. 4. Geometric feature vector

their feature vectors. This contrasts to using the feature vectors to compute a classification (e.g. rectangle or circle) and then comparing the classes ascribed to each shape. Our approach tends to work well if individual features are stable and robust, which we have found out experimentally in [8].

5 Experimental Results

In order to evaluate the retrieval capability (i.e. accuracy) of our method, we measured recall and precision performance figures using calibrated test data. Recall is the percentage of similar drawings retrieved with respect to the total number of similar drawings in the database. Conversely, precision is the percentage of similar drawings retrieved with respect to the total number of retrieved drawings.

We compared our method to describe shapes (CALI) with five other approaches, namely Zernike Moments (ZMD), Fourier descriptors (FD), grid-based (GB), Delaunay triangulation (DT) and Touch-point-vertex-angle-sequence (TPVAS). To that end we used results of an experiment previously performed by Safar [21], where he contrasted his approach (TPVAS) to the FD, GB and DT methods.

In that experiment, authors used a database containing 100 contours of fish shapes, as the ones presented in Figure 5. From the set of one hundred shapes in the database, five were selected randomly as queries. Before measuring the effectiveness of all methods, Safar performed a perception experiment where users had to select (from the database) the ten most similar to each query. This yielded the ten most perceptually similar results that each query should produce.

We repeated this experiment, on the same database and performing the same queries, using our method and an implementation of Zernike moments.

First we computed descriptors for each of the hundred shapes in the data set. Then for each query, we computed the corresponding descriptor and used it to search for the ten nearest-neighbors. For each of the five queries, we determined the positions for the 10 similar shapes in the ordered response set. Using results from our method and the values presented in Table 2 from [21] we produced the precision-recall plot shown in Figure 6.

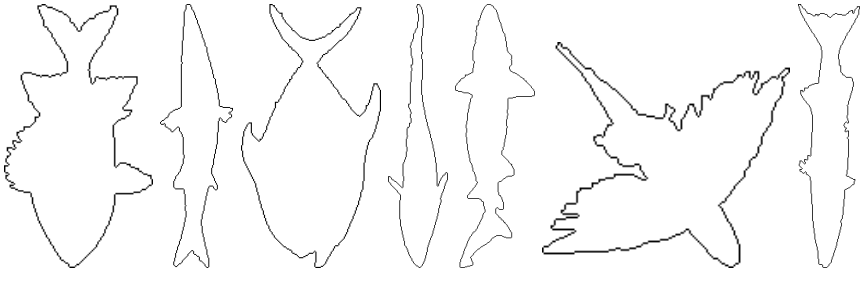


Fig. 5. Example of objects stored into the test database

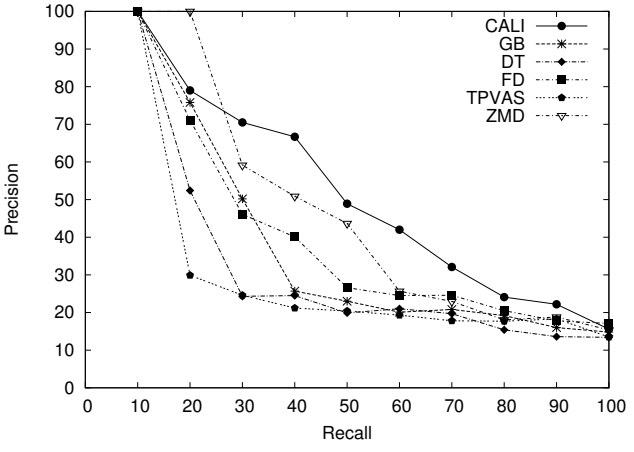
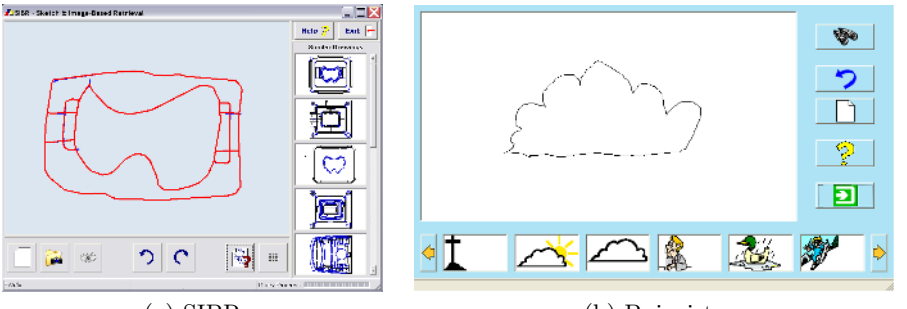


Fig. 6. Precision-recall comparison



(a) SIBR (b) Bajavista

Fig. 7. Screen-shots of prototypes

Looking at the precision-recall chart we can see that our approach outperforms all other algorithms studied, including for the most part, the Zernike moments, which according to a previous experimental evaluation [23], were considered the best method to describe geometric shapes. Furthermore, our technique yields superior precision to all methods for all measured recall values except for recall values equal or below 20% where Zernike moments show a slight advantage.

Thus we can say that our method presents better results when drawings in the database are slightly different from the query, while Zernike moments tend work better for elements in the database which are very similar to the query. While Zernike moments tend to present better results in the topmost three to five queries, our method will likely yield more correct matches, although some of these might be ranked in lower positions. Thus, we believe that our technique of describing shape geometry is more suited to approximate queries in content-based retrieval than Zernike moments.

Although the features used by CALI were mainly selected to classify and describe geometric shapes, we can conclude from this experimental evaluation,

that it can also be used to describe more general shapes, as the contours from this database. Furthermore, our geometric features were chosen to classify convex objects out of a limited vocabulary. One interesting finding is that the set is surprisingly expressive and general enough to measure shape similarity instead of classification.

To assess the applicability of our approach for content-based retrieval in real-life settings, we developed two prototypes, one to retrieve technical drawings (SIBR) [10] and other for clip-art drawings (BajaVista) [5]. The SIBR prototype allows retrieving sets of drawings similar to a hand-sketched query or a digitized drawing.

Figure 7(a) depicts a screen-shot of the calligraphic interface of the SIBR application. On the left we can see the sketch of a part and on the right the results returned by the implied query. These results are ordered from top to bottom and from left to right, with the most similar on top. On the other hand, the BajaVista prototype can index and retrieve clip-art drawings by content, either using sketches or querying by example. Figure 7(b) depicts a screen-shot of this application. On the top-left we can see the sketch of a cloud and on the bottom results returned by the implied query. These results are ordered from left to right, with the most similar on the left. It is also possible to perform query-by-example, thus allowing the user to select one of the results and using it to specify the next query, since our classification scheme handles graphics and sketches in the same manner.

These two prototypes were evaluated using medium–size databases. The SIBR prototype was tested on a database containing one hundred elements, while the database used to test BajaVista indexed 968 drawings. Tests with both prototypes showed effective results when searching for both technical or clip-art drawings. Moreover, we measured query execution for both prototypes and found times between 2 to 5 seconds on a Tablet PC (Pentium III@800 MHz, 256MB of RAM) and on a PC (AMD Duron@1.3GHz, with 256MB of RAM), while the total time for users to draw the sketch and obtain results was less than one minute, in most cases. Furthermore, users were satisfied that returned results matched their expectations. Indeed, while in first instances we presented the topmost five drawings in each case, feedback from tests convinced us to increase the displayed set to ten or twenty drawings. Surprisingly, users assigned greater importance to being able to retrieve the desired result among the top 10 or 20 elements, rather than finding the two "best" candidates. Indeed, we were told by users that they preferred recall over precision (at least in this limited sense) which empirically supports our claims about the precision–versus–recall performance of our technique against Zernike moments.

6 Conclusions

We presented a shape classification method which can be applied to content–based retrieval of two–dimensional vector graphics. Unlike other approaches, our method works not by a-priori classifications but by estimating the resemblance

of a given shape to each of the forms in a restricted set. In this manner we are able to characterize many different two–dimensional shapes with few restrictions. Experimental evaluation of our method seems to indicate superior performance against other known and sound approaches. However, we have not tested it with strongly concave shapes, for which it is not clear whether convex geometrical features will work well. This is the subject of ongoing work.

From an analysis of experimental results, our approach on shape classification for retrieval proved well on both theoretical and practical grounds. We have developed successful applications for retrieving two–dimensional CAD drawings and clip–art images.

Although we had only compared our geometry description method with others outside the retrieval systems, we plan to compare the precision-recall performance from both retrieval systems (SIBR and BajaVista) using the Zernike Moments.

One area for future work lies in extending our approach to three–dimensional vector drawings, where preliminary findings seem to yield promising results [6]. We strongly believe that an approach based on explicit structural descriptions has the potential to find a wide range of applications for human–made vector drawings.

References

1. J. E. Boyce and D. P. Dobkin. Finding Extremal Polygons. *SIAM Journal on Computing*, 14(1):134–147, February 1985.
2. Dragos Cvetkovic, Peter Rowlinson, and Slobodan Simic. *Eigenspaces of Graphs*. Cambridge University Press, United Kingdom, 1997.
3. E. R. Davies. *Machine Vision: Theory, Algorithms, Practicalities*. Academic Press, 1997.
4. Manuel J. Fonseca. *Sketch-Based Retrieval in Large Sets of Drawings*. PhD thesis, Instituto Superior Técnico / Universidade Técnica de Lisboa, July 2004.
5. Manuel J. Fonseca, Bruno Barroso, Pedro Ribeiro, and Joaquim A. Jorge. Retrieving ClipArt Images by Content. In *Proceedings of the 3rd International Conference on Image and Video Retrieval (CIVR'04)*, LNCS. Springer-Verlag, 2004.
6. Manuel J. Fonseca, Alfredo Ferreira, and Joaquim A. Jorge. Towards 3D Modeling using Sketches and Retrieval. In *Proceedings of the first Eurographics Workshop on Sketch–Based Interfaces and Modeling*, Grenoble, France, 2004. EG Publishing.
7. Manuel J. Fonseca and Joaquim A. Jorge. CALI : A Software Library for Calligraphic Interfaces. INESC-ID, available at `http://immi.inesc-id.pt/cali/`, 2000.
8. Manuel J. Fonseca and Joaquim A. Jorge. Experimental Evaluation of an on-line Scribble Recognizer. *Pattern Recognition Letters*, 22(12):1311–1319, 2001.
9. Manuel J. Fonseca and Joaquim A. Jorge. Experimental evaluation of an on-line scribble recognizer. *Pattern Recognition Letters*, 22(12):1311–1319, Jan 2001.
10. Manuel J. Fonseca, Alfredo Ferreira Jr., and Joaquim A. Jorge. Content-Based Retrieval of Technical Drawings. *International Journal of Computer Applications in Technology (IJCAT)*, 23(2/3/4):86–100, 2005.

11. H. Freeman and A. Saghri. Generalized Chain Codes for Planar Curves. In *Proceedings of the International Joint Conference on Pattern Recognition*, pages 701–703, Kyoto, Japan, November 1978.
12. Herbert Freeman and Ruth Shapira. Determining the Minimum-area Encasing Rectangle for an Arbitrary Closed Curve. *Communications of the ACM*, 18(7):409–413, July 1975.
13. Chuang Gu. *Multivalued Morphology and Segmentation-based Coding*. Phd thesis, Signal Processing Lab. of Swiss Federal Institute of Technology at Lausanne, 1995.
14. Ming-Kuei Hu. Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, 8:179–187, 1962.
15. Hannu Kauppinen, Tapio Seppanen, and Matti Pietikainen. An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 17(2):201–207, 1995.
16. G. J. Lu and A. Sajjanhar. Region-Based Shape Representation and Similarity Measure Suitable for Content-Based Image Retrieval. *Multimedia Systems*, 7:165–174, 1999.
17. B. M. Mehtre, M. S. Kankanhali, and W. F. Lee. Shape Measures for Content Based Image Retrieval: A Comparison. *Information Processing and Management*, 33(3):319–337, 1997.
18. F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and Robust Retrieval by Shape Content through Curvature Scale Space. In *International Workshop on Image Databases and Nultimedia Search*, pages 35–42, Amsterdam, The Netherlands, 1996.
19. Joseph O'Rourke. *Computational Geometry in C*. Cambridge University Press, 2nd edition, 1998.
20. Eric Persoon and King-Sun Fu. Shape Discrimination Using Fourier Descriptors. *IEEE Trans. on Systems, Man and Cybernetics*, 7(3):170–179, 1977.
21. Maytham Safar, Cyrus Shahabi, and Chung hao Tan. Resiliency and Robustness of Alternative Shape-Based Image Retrieval Techniques. In *Proceedings of IEEE International Database Engineering and Applications Symposium (IDEAS'00)*, pages 337–348, 2000.
22. Maytham Safar, Cyrus Shahabi, and Xiaoming Sun. Image Retrieval By Shape: A Comparative Study. In *Proceedings of the IEEE International Conference on Multimedia and Exposition. (ICME'00)*, pages 141–144, 2000.
23. D. S. Zhang and G. Lu. A comparative study of curvature scale space and fourier descriptors. *Journal of Visual Communication and Image Representation*, 14(1):41–60, 2003.

Polygonal Approximation of Digital Curves Using a Multi-objective Genetic Algorithm

Herve Locteau, Romain Raveaux, Sebastien Adam, Yves Lecourtier, Pierre Heroux, and Eric Trupin

LITIS, Université de Rouen,
F-76800 Saint-Etienne du Rouvray, France
Herve.Locteau@univ-rouen.fr

Abstract. In this paper, a polygonal approximation approach based on a multi-objective genetic algorithm is proposed. In this method, the optimization/exploration algorithm locates breakpoints on the digital curve by minimizing simultaneously the number of breakpoints and the approximation error. Using such an approach, the algorithm proposes a set of solutions at its end. This set which is called the Pareto Front in the multi objective optimization field contains solutions that represent trade-offs between the two classical quality criteria of polygonal approximation : the Integral Square Error (ISE) and the number of vertices. The user may choose his own solution according to its objective. The proposed approach is evaluated on curves issued from the literature and compared with many classical approaches.

1 Introduction

Polygonal approximation of digital planar curves is an important issue in pattern recognition and image processing. It is a classical way to represent, store and process digital curves. For example, its results are frequently used for shape recognition. The problem can be stated as follows: Given a curve C consisting of N ordered points $C = \{C_i \equiv (x_i, y_i)\}_{i=1}^{N}$, the goal is to find a subset S of M ordered points $S = \{S_i \equiv (x_i, y_i)\}_{i=1}^{M}$ with $M \leq N$ and $S \subseteq C$. These points constitute the extremities of line segments so that the polygon constructed by directly connecting these line segments best fits the given digital curve. Figure 1, issued from [1], illustrates such a process for two different curves. Many paradigms have been proposed in the literature to solve the problem of polygonal approximation, what leads to a great number of published papers. Among these approaches, one can cite split or split and merge techniques [2][3][4], Hough transform [5], perceptual organization [6], dominant points detection [7][8][9][10][11][12], competitive Hopfield neural networks [13] or dynamic programming [14][1][15]. Another paradigm has been recently proposed in [16][17][18]. It consists in using Genetic Algorithms in order to find a near-optimal polygonal approximation. In such an approach, as in dynamic programming methods, the polygonal approximation technique is considered as an optimization process and the algorithm automatically selects the best points of the polygon by

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 300 – 311, 2006.
© Springer-Verlag Berlin Heidelberg 2006

minimizing a given criterion. Two kinds of configuration may be distinguished in the published papers. In the first case, the number of vertices to be obtained is fixed and the method uses the concept of genetic evolution to obtain a near-optimal polygon [16][17]. In the second case, a maximal approximation error is fixed and the algorithm minimizes the number of vertices of the polygon [16]. One can note that this kind of approach has also been used in order to approximate curves with circular arcs or ellipses [18][19].

In this paper, we adopt the same paradigm and we propose a new algorithm for polygonal approximation using genetic algorithms. The originality of the described approach is the factorization of the two kinds of configuration mentioned above through the use of a multi-objective optimization process whereas existing approaches minimize an unique criterion as explained before. Such a new viewpoint enables the user of the system to choose a trade-off between different quality criteria, according to its objectives of use of the results (recognition, interpretation, storage...).

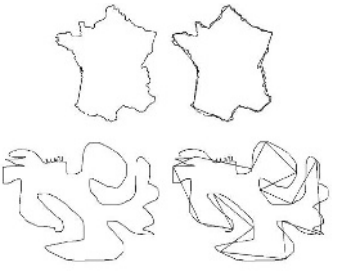


Fig. 1. Results provided by a polygonal approximation [1]. On the top left, a curve composed of 992 points. On the top right, its polygonal approximation (15 line segments). At the bottom left, a free curve of 3222 points, on the right, its polygonal approximation (16 line segments).

The remainder of the paper is organized as follows. In section 2, an introduction to the multi-objective optimization problem is proposed and our algorithm is detailed. In section 3, the application of this algorithm to the polygonal approximation problem is shown. Section 4 presents the experimentally obtained results, a comparison with classical approaches and a discussion concerning the interests of such an approach. Section 5 summarizes the concluding remarks and proposes some perspectives for this work.

2 A Genetic Based Multi-objective Optimization Algorithm

When an optimisation problem involves more than one objective function (a very frequent context in the document analysis field – one can cite recognition rate/reject rate, precision rate/recall rate, compression / quality), the task of finding one or more optimum solutions is known as multi-objective optimization. Some classical textbooks on this subject have been published, e.g. [20]. We just recall here some essential notions in order to introduce the proposed algorithm. The main difference between single and multi-optimization task lies in the requirement of compromises

between the various objectives in the multi-optimization case. Even with only two objectives, if they are conflicting, the improvement of one of them leads to a deterioration of the other one. For example, in the context of polygonal approximation, the decrease of the approximation error always leads to an increase of the vertices number. Two main approaches are used to overcome this problem in the literature. The first one is to combine the different objectives in a single one (the simpler way being to use a linear combination of the various objectives), and then to use one of the well-known techniques of single objective optimization (like gradient based methods, simulated annealing or classical genetic algorithm). In such a case, the compromise between the objectives is a priori determined through the choice of the combination rule. The main critics addressed to this approach are the difficulty to choose a priori the compromise and the fact that some objective points can not be reached. It seems a better idea to postpone this choice after having several candidate solutions at hand. This is the goal of Pareto based method using the notion of dominance between candidate solutions. A solution dominates another one if it is better for all the objectives. This dominance concept is illustrated on figure 2. On this example, two criteria J_1 and J_2 have to be minimized. The set of non-dominated points that constitutes the Pareto-Front appears as 'O' on the figure, while dominated solutions are drawn as 'X'.

Using such a dominance concept, the objective of the optimization algorithm becomes to determine the Pareto front, that is to say the set of non-dominated points. Among the optimization methods that can be used for such a task, genetic algorithms are well-suited because they work on a population of candidate solutions. They have been extensively used in such a context, with many variants. The most common algorithms are VEGA – Vector Evaluated Genetic Algorithm – [21], MOGA – Multi Objective Genetic Algorithm –approach [22], NSGA – Non-Dominated Sorting Genetic Algorithm – [23], NSGA II [24], PAES – Pareto Archived Evolution Strategy – [25] and SPEA – Strength Pareto Evolutionary Algorithm – [26]. The strategies used in these contributions are different, but the obtained results mainly vary from the convergence speed point of view. A good review of the existing approaches can be found in [27].

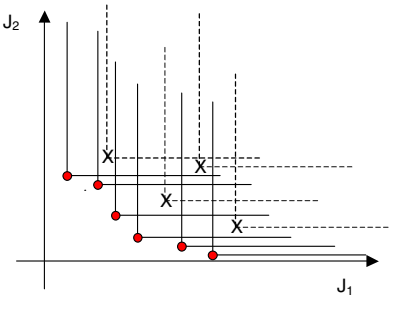


Fig. 2. Illustration of the Pareto Front concept

The proposed genetic algorithm is elitist and steady-state. This means that (i) it manages two populations and (ii) the replacement strategy of individuals in the populations is not made as a whole, but individual per individual. The two populations are a classical population, composed of evolving individuals and an "archive" population

composed of the current Pareto Front approximation elements. These two populations are mixed during the genetic algorithm. The first population guarantees space exploration while the archive guarantees the exploitation of acquired knowledge and the convergence of the algorithm.

Based on such concepts, our optimization method uses the algorithm #1. In this section, we only describe the optimization algorithm, without applying it to the polygonal approximation problem. The particular application to the polygonal approximation problem is described in the following section. This algorithm has been designed in order to be applied to various problems (see section 5). The design of a new application only consists in the choice of a coding scheme for individuals, in the design of the evaluation method and in the choice of the genetic parameters values.

Algorithm #1. The multi objective algorithm in use

Population Initialization

Archive Initialization (selection of the non dominated element in the population)

do

- *Random selection of two individuals I_1 and I_2 in the current population*
- *Crossover between the selected individuals to generate I_3 and I_4*
- *Mutation applied to the generated children I_3 and I_4*
- *Evaluation of children I_3 and I_4*
- *Selection either of the dominant individual I_5 between mutated children (if it exists) or random selection of I_5 between I_3 and I_4*
- *Random selection of an individual in the archive population (I_6)*
- *Crossover between I_5 and I_6 to generate children I_7 and I_8*
- *Evaluation of children I_7 and I_8*
- *Test for the integration of I_7 and I_8 in the archive*
- *Test for the integration of I_7 and I_8 in the current population*
- *Incrementation of the generation number*

While *the maximal generation number has not been reached*

In the current implementation of this algorithm, the coding of an individual is a classical bit set, the crossover is a well-known 2-point crossover, and the mutation is a simple transformation of a gene value by its complementary value. Concerning the replacement strategy, several choices can be made for the integration of a candidate individual in the archive. The simplest is a dominance test between the candidate and the archive elements. The candidate is inserted within the archive if no archive element dominates it. In the same time, archive elements dominated by the candidate are eliminated from the archive. A problem reported in the literature on evolutionary multi-objective optimization is the possible bad exploration of Pareto front: the archive population elements concentrate on only some parts of the front. This difficulty is overcome in our approach by defining a minimal distance between two points

in the objective space. This algorithm has been tested on classical multi-objective problems such as BNH, TNK or OSY [28] problems and the results have shown the quality of the proposed approach.

3 A Genetic Based Multi-objective Optimization Algorithm

In order to apply the algorithm presented above to the polygonal approximation problem, an individual has to represent a possible solution to the polygonal approximation problem. That is why an individual is composed of N genes, where N is the number of points in the initial curve. A gene is set to '1' if the point is kept as a breakpoint, '0' if it is not. An example of an individual coding is given in figure 3. Each point C_i of the curve S corresponds to a bit in the chromosome. In the example of figure 3, the individual is a binary string of 14 genes corresponding to the initial C_0.- C_{13}.. The example polygon is composed of 4 vertices whose extremities are C_0, C_4, C_7 and C_{11}. Such a polygonal approximation corresponds to the individual "10001001000100".

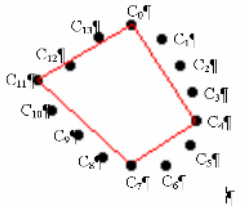


Fig. 3. An example of the coding scheme applied on a simple curve

Concerning the computation of the criterion to be optimized, two objectives have been included in the current version. The first one is the Integral Square Error (ISE) and the second one is the vertices number. Such a choice enables to have a trade-off between the precision of the result and the number of line segments, thanks to elements of the Pareto front. One can note that the use of a discrete objective (vertices number) guarantees itself the diversity on the Pareto front, we do not need to specify any minimal distance between any couples of solutions of the Pareto Front.

4 Experimental Results and Performance Analysis

In order to assess the performances of the proposed algorithm, it has been applied to the four digital curves presented in [7]. Fig. 4a is a chromosome shaped curve with 60 points, Fig. 4b is a leaf-shaped curve with 120 points, Fig. 4d is a curve containing four semi-circles with 102 points and Fig. 4c is a figure-eight curve with 45 points. These curves have been broadly used in the literature. Such tests allow to test the performances of the proposed algorithm versus those of published approaches. For each of these curves, the program has been run for 2000 generations, using a

population size of 100 individuals. Such a parameter set involves about 8000 calls to the evaluation method (see the algorithm below). The mutation rate has been fixed to 0.05 and the crossover rate to 0.6. As said before, the output of the presented algorithm is not a single ISE for a number of vertices given a priori. It consists in the whole Pareto front of the optimization problem. That is why the result is a set of couple (ISE – number of vertices). As an example, figure 5 shows the set of couple obtained at the end of one run on a "map of France" curve and the corresponding polygons. Another remark has to be done. Since GA are stochastic, results may be different at independent runs. That is why, in these experiments, we present the best and the worst ISE for each number of vertices obtained after 5 independent runs. Using such a strategy, obtained results are compared for each curve with the results proposed in the literature using the following scheme:

- A table containing the results obtained by several approaches from the literature on the given figure;
- The results (comparable points on the Pareto Front) obtained by the proposed algorithm on the same curve compared with the best ISE found in the literature;
- A figure representing the whole Pareto Front and showing a visual performance comparison between our results and the results issued from the state of the art.

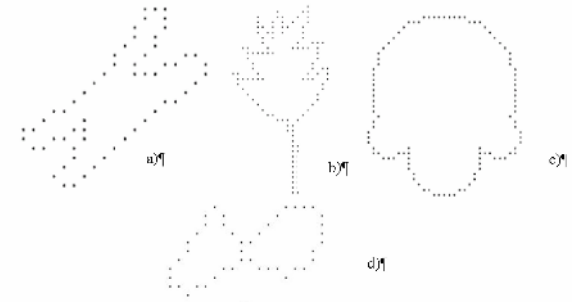


Fig. 4. Test curves: (a) chromosome-shaped curve: 60 points; (b) leaf-shaped curve: 120 points; (c) curve with four semi-circles: 102 points and (d) figure-of-eight curve: 45 points

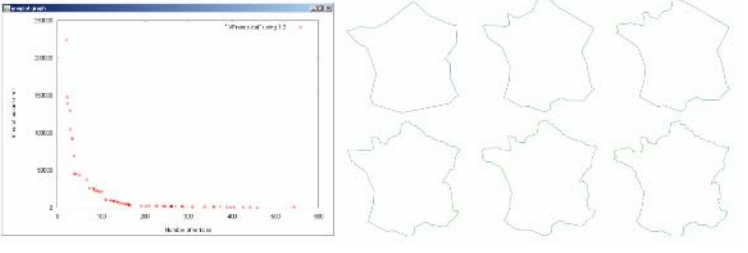


Fig. 5. Results obtained on a curve of a map of France: on the left, the obtained Pareto front, on the right, the polygons corresponding to the points labeled (1-6) on the Pareto front

4.1 Results Obtained on the Chromosome-Shaped Curve

Table 1. Results obtained with several existing methods on the chromosome-shaped curve

Algorithm	Number of vertices	Integral square error
(Wall and Danielsson, 1984) [4]	17	12.19
(Teh and Chin, 1987) [7]	15	7.20
(Ray and Ray, 1992a) [9]	18	4.81
(Ray and Ray, 1992b) [10]	18	5.56
(Cornic, 1997) [11]	12	9.57
(Cornic, 1997) [11]	17	5.54
(Perez and Vidal, 1994) [14]	12	5.82
(Perez and Vidal, 1994) [14]	17	3.13
(Huang and Sun, 1999) [17]	12	7.63
(Yin, 1998) [16]	12	7.99
(Marji and Siy, 2003) [12]	12	8.03
(Horng and Li, 2002) [15]	12	5.82

Table 2. Results obtained using 5 runs of the described approach on the chromosome-shaped curve compared with best results found in the literature

Number of vertices	10	11	12	13	14	15	16	17	18	19	20
Min. ISE	11.5	9.8	6.7	6.3	6.3	5.3	4.5	4.3	3.5	3.4	3.2
Max. ISE	15.1	12.2	8.4	8.1	7.3	6.8	5.5	5.0	4.7	4.2	3.9
Best ISE	-	-	5.8	-	-	7.2	-	3.1	4.8	-	-

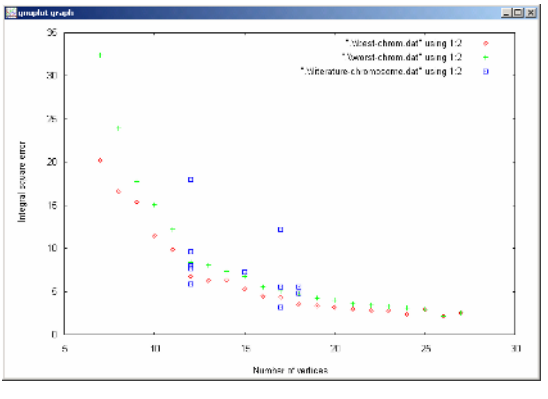


Fig. 6. Comparison between performances obtained using several approaches issued from the literature (□); our best result (o) and our worst result (+) for 5 runs on the chromosome-shaped curve

4.2 Results Obtained on the Leaf-Shaped Curve

Table 3. Results obtained with several existing methods on the leaf-shaped curve

Algorithm	Number of vertices	Integral square error
(Teh and Chin, 1987) [7]	29	14.96
(Ray and Ray, 1992a) [9]	32	14.18
(Ray and Ray, 1992b) [10]	32	14.718
(Cornic, 1997) [11]	23	25.8
(Cornic, 1997) [11]	28	19.88
(Perez and Vidal, 1994) [14]	17	22.42
(Perez and Vidal, 1994) [14]	28	6.80
(Huang and Sun, 1999) [17]	21	17.96
(Marji and Siy, 2003) [12]	22	13.21
(Horng and Li, 2002) [15]	17	22.42

Table 4. Results obtained using 5 runs of the described approach on the leaf-shaped curve compared with best results found in the literature

Number of vertices	17	21	22	23	24	25	26	27	28	29
Min. ISE	26.3	18.8	17.8	17.3	16.0	14.5	14.4	12.4	10.9	12.1
Max. ISE	45.3	28,2	24.2	20.0	18.3	17.2	15.9	15.0	14.5	12.7
Best ISE	22,4	18.0	13.2	25.8	-	-	-	-	6.8	15.0

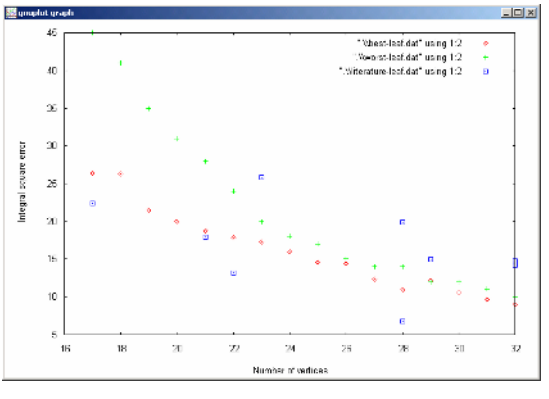


Fig. 7. Comparison between performances obtained using several approaches issued from the literature (□); our best result (o) and our worst result (+) for 5 runs on the leaf-shaped curve

4.3 Results Obtained on the Curve with 4 Semi-circles

Table 5. Results obtained with several existing methods on the curve with 4 semi-circles

Algorithm	Number of vertices	Integral square error
(Wall and Danielsson, 1984) [4]	30	7.03
(The and Chin, 1987) [7]	22	20.61
(Ray and Ray, 1992a) [9]	27	11.50
(Ray and Ray, 1992b) [10]	29	11.82
(Cornic, 1997) [11]	22	13.00
(Cornic, 1997) [11]	30	8.38
(Perez and Vidal, 1994) [14]	15	14.40
(Perez and Vidal, 1994) [14]	30	2.64
(Huang and Sun, 1999) [17]	14	17.74
(Huang and Sun, 1999) [17]	22	9.02
(Yin, 1998) [16]	14	29.93
(Yin, 1998) [16]	22	12.91
(Marji and Siy, 2003) [12]	26	9.01
(Horng and Li, 2002) [15]	15	14.40

Table 6. Results obtained using 5 runs of the described approach on the curve with 4 semi-circles compared with best results found in the literature

Number of vertices	14	15	18	20	22	24	26	27	29	30	31
Min. ISE	27.8	19.2	12.7	10.3	8.7	7.0	5.7	5.1	4.8	4.2	3.7
Max. ISE	41.6	23.7	16.4	13.7	11.5	9.5	7.0	6.2	5.6	5.0	4.7
Best ISE	17.7	14.4	-	-	9.0	-	9.0	11.5	11.8	2.6	-

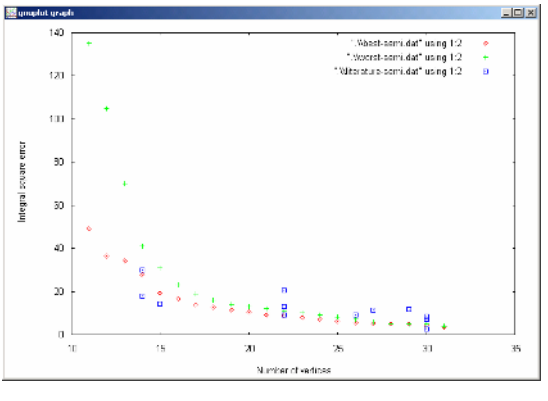


Fig. 8. Comparison between performances obtained using several approaches issued from the literature (□); our best result (o) and our worst result (+) for 5 runs on the curve with 4 semi-circles

4.4 Results Obtained on the Figure-Eight Curve

Table 7. Results obtained with several existing methods on the curve with 4 semi-circles

Algorithm	Number of vertices	Integral square error
(Wall and Danielsson, 1984) [4]	13	13.92
(Teh and Chin, 1987) [7]	13	5.93
(Ray and Ray, 1992a) [9]	15	4.39
(Cornic, 1997) [11]	12	3.89
(Perez and Vidal, 1994) [14]	11	2.90
(Yin, 1998) [16]	11	3.83
(Yin, 1998) [16]	13	2.24
(Yin, 1998) [16]	15	2.01
(Horng and Li, 2002) [15]	11	2.90

Table 8. Results obtained using 5 runs of the described approach on the curve with 4 semi-circles compared with best results found in the literature

Number of vertices	8	9	10	11	12	13	14	15	16	17	18
Min. ISE	6.2	4.7	3.8	3.3	2.4	2.0	1.8	1.6	1.4	1.3	1.2
Max. ISE	8.8	5.0	4.1	3.5	2.9	2.4	2.2	2.0	1.8	1.7	1.4
Best ISE	-	-	-	2.9	3.9	2.2	-	2.0	-	-	-

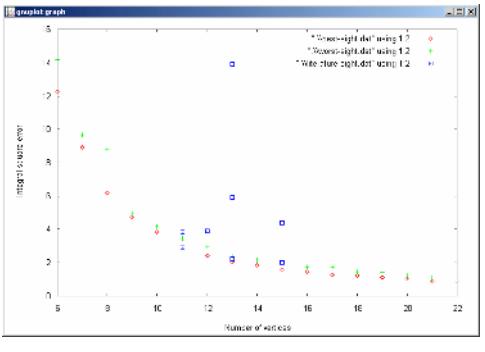


Fig. 9. Comparison between performances obtained using several approaches issued from the literature (□); our best result (o) and our worst result (+) for 5 runs

4.5 Discussion

Many observation can be drawn from the results presented above. Concerning the obtained performances for a given number of vertices, one can see that the proposed approach provides results that are only outperformed by optimal methods [14][15]. These results are logical since GA are known to be near-optimal. On the contrary, the non-optimal or sub-optimal approaches are outperformed by the proposed algorithm.

Another advantage of our approach is the fact that no input parameters concerning the initial curve or the desired results have to be defined a priori. All the tests mentioned above have been led with exactly the same configuration of the program. The most important in this context is that our approach does not need the number of vertices to be obtained as an a priori parameter. Furthermore, the algorithm gives the user a better information on the solved problem as it proposes as output a set of solutions containing a wide range of values for the number of vertices and the corresponding ISE. The user of the system can choose the better solution regarding its application constraints, as shown in Fig. 4. Concerning the complexity, our algorithm has an important computational cost (comparable to Yin's one) but for a pool of solutions. Moreover, this computational cost may be reduced using a parallelization since genetic algorithms are parallel by nature. Another solution to reduce this complexity is to consider an evolution of the genetic parameters (mutation and crossover rates) during the algorithm. This point is actually under consideration.

5 Conclusion and Future Works

In this paper, we have proposed a new approach for the polygonal approximation of curves. This approach considers the polygonal approximation as an optimization process. The fundamental difference with existing approaches lies in the fact that we use a multi-objective optimization process while other contributions only optimize a unique objective. One can see several interests in such an approach. As many solutions are proposed, the user may choose the optimal solution regarding its constraints. Another interest is that it is easy to add a new objective. For example, our current work concerns the maximization of the parallel line segments, in order to apply contour matching in a vectorization process. Another future work concerns the integration of the detection of circular arcs in our approximation system, using an approach inspired from [18]. A more global objective is to generalize the principles of multi-objective optimization to the different steps constituting the chain of a document image analysis system. The aim is to build a multi-objective document analysis system which adapts its objective thanks to a dialog with the user.

References

1. Salotti, M.: An efficient algorithm for the optimal polygonal approximation of digitized curves", PRL 22 (2001) 215-221
2. Ramer, U.: An iterative procedure for the polygonal approximation of plane curves, CGIP 1 (1972) 291-297
3. Pavlidis, T., Horowitz, S.L.: Segmentation of plane curves, IEEE Transaction on Computers 23 (1974) 860-870
4. Wall, K., Danielsson, P.E.: A fast sequential method for polygonal approximation of digitized curves, CVGIP 28 (1984) 220-227
5. Gupta, A. Chaudhury, S., Parthasarathy, G.: A new approach for aggregating edge points into line segments, PR 26 (1993) 1069-1086
6. Hu, J., Yan, H.: Polygonal approximation of digital curves based on the principles of perceptual organization, PR 30 (2002) 701-718

7. Teh, C., Chin, R.T.: On the detection of dominant points on digital curves, IEEE transaction on PAMI 23 (1989) 859-872
8. Ansari, N., Delp, E.J.: On detecting dominant points, PR 24 (1991) 441-451
9. Ray, B.K., Ray, K.S.: An algorithm for detecting dominant points and polygonal approximation of digitized curves, PRL 13 1992) 849-856
10. Ray, B.K., Ray, K.S.: Detection of significant points and polygonal approximation of digitized curves, PRL 12 (1992) 443-452
11. Cornic, P.: Another look at the dominant point detection of digitized curves, PRL 18 (1997) 13-25
12. Marji, M., Siy, P.: A new algorithm for dominant points detection and polygonization of digital curves, PR 36 (2003) 2239-2251
13. Chung, P.C., Tsai, C.T., Chen, E.L., Sun, Y.N.: Polygonal approximation using a competitive Hopfield neural network, PR 27 (1994) 1505-1512
14. Perez, J.C., Vidal, E.: Optimum polygonal approximation of digitized curves, PRL 15 (1994) 743-750
15. Horng, J.H., Li, J.T., An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves, PRL 23 (2002) 171-182
16. Yin, P.Y.: A new method for polygonal approximation of digital curves, PRL 19 (1998) 1017-1026
17. Huang, S.C., Sun Y.N.: Polygonal approximation using genetic algorithm, PR 32 (1999) 1409-1420
18. Sarkar, B., Singh, L.K., Sarkar, D.: Approximation of digital curves with line segments and circular arcs using genetic algorithms, PRL 24 (2003) 2585-2595
19. Yin, P.Y.: A new circle/ellipse detector using genetic algorithm, PRL 20 (1999) 731-740
20. Deb, K.: Multi-Objective optimization using Evolutionary algorithms, Wiley, London, 2001
21. Schaffer, J.D., Grefenstette, J.J.: Multiobjective learning via genetic algorithms, In Proceedings of the 9th IJCAI (1985) 593-595
22. Fonseca, C.M., Fleming, P.J.: Genetic algorithm for multi-objective optimization: formulation, discussion and generalization, In the proceedings of the fifth ICGA (1993) 416-423
23. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithm, EC 2 (1994) 221-248
24. Deb, K., Agrawal, S., Pratab, A., Meyarivan, T., A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions on EC 6 (2000) 182-197
25. Knowles, J.D., Corne, D.W.: Approximating the nondominated front using the Pareto archived evolution strategy, EC 8 (2000) 149-172
26. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms : a comparative study and the strength pareto approach, IEEE Transactions on EC 3 (1999) 257-271
27. Coello Coello C. A.: A short tutorial on Evolutionary Multiobjective Optimisation, In First International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, . Springer-Verlag n° 1993 (2001) 21-40
28. Chafekar, D., Xuan, J., Rasheed, K.: Constrained Multi-objective Optimization Using Steady State Genetic Algorithms, In Proceedings of GECC (2003), 813-824

A Contour Shape Description Method Via Transformation to Rotation and Scale Invariant Coordinates System

Min-Ki Kim

Research Institute of Computer and Information Communication
Gyeongsang National University, Dept. of Computer Science Education
900, Gajwa-dong, Jinju, 660-701, Korea
mkkim@gsnu.ac.kr

Abstract. Rotation and scale variations complicate the matters of shape description and recognition because these variations change the location of points composing the shape. However, some geometric invariant points and the relations among them are not changed by these variations. Therefore, if points in image space depicted with the *x*-*y* coordinates system can be transformed into a new coordinates system that are invariant to rotation and scale, the problem of shape description and recognition becomes easier. This paper presents a shape description method via transformation from the image space into the invariant feature space having *d*- and *c*-axes: representing relative distance from a centroid and contour segment curvature (CSC) respectively. The relative distance describes how far a point departs from the centroid, and the CSC represents the degree of fluctuation in a contour segment. After transformation, mesh features were used to describe the shape mapped onto the *d*-*c* plane. Traditional mesh features extracted from the *x*-*y* plane are sensitive to rotation, whereas the mesh features from the *d*-*c* plane are robust to it. Experimental results show that the proposed method is robust to rotation and scale variations.

1 Introduction

A variety of graphic symbols have been widely used in maps, architectural drawings, and mechanical drawings etc. A standardized graphic symbol can substitute a lengthy description. It also has the added advantage that illiterates and foreigners can easily understand the meaning. Pictogram used on public sign shows these advantages of graphic symbols. In general, a graphic symbol can represent a meaning through the use of shape only. In some cases, additional information such as color is used to represent more accurate meanings.

Shape is a topic of great interest, in the field of graphic symbol recognition but also in other fields of pattern recognition. Numerous researches for shape description and recognition have been steadily performed since the early days of pattern recognition [1, 2, 3]. One of the greatest concerns is regarding the feature of invariant to two-dimensional transformations [4]. Since Hu [5] used moment invariants for two-dimensional pattern recognition application, many researchers have attempted shape recognition using moment invariants. The Hu moment invariants were used in [6],

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 312 – 322, 2006.
© Springer-Verlag Berlin Heidelberg 2006

and new moments were proposed in [7, 8]. The main problem with moment invariants is that the few invariants derived from lower order moments are insufficient to accurately describe shape. Higher order moments are difficult to derive and sensitive to noise. Furthermore, the meaning behind higher order moments is difficult to understand. Geometric invariants such as angle, area, and length ratio have also been widely used for shape recognition. The center of mass is a representative geometric invariant feature. Chang[9] and Zhang[10] used the relative distances of feature points from the centroid for shape description. The performance of these methods depends on the accurate extraction of the feature points. A shape description method based on invariant signature, calculated point-wise, was used in [11, 12]. Generally, it is more robust to local noise than the method based on feature points.

This paper presents a shape description method via transformation to new coordinates system. The coordinates of points can be fixed regardless of rotation or scale variation by using geometric invariants. It makes the problem of shape description and recognition easier. The new coordinates system is described in section 2, and the feature extraction and matching method is explained in section 3. In section 4, experimental results showing the performance of the proposed method are presented. Finally, the conclusion is presented in section 5.

2 Rotation and Scale Invariant Coordinates System

Images with same shape have different coordinate values in the x-y plane after being rotated or scaled. If the coordinate values can be fixed regardless of the rotation or scale variation, the shape deformation through the process of similarity transformation can be overcome. This section presents a new coordinates system based on two geometric invariants: relative distance from a centroid and contour segment curvature.

2.1 Relative Distance from a Centroid

The distance from the centroid to a point on contour is invariant to rotation, but it is proportional to the scale. Therefore, the relative distance that is defined by equation (1) was used, where C and P_i respectively represent the centroid and a point on contour, and n is the number of points composing contour. Accordingly, the $D(P_i)$ has a value between 0 to 1.

$$D(P_i) = \frac{Length(\overline{CP}_i)}{Max(Length(\overline{CP}_i))}, \text{ where } i = 1..n \tag{1}$$

2.2 Contour Segment Curvature

A contour is a closed path. Thus if a point on contour is selected as starting point and the next point is visited in one direction continuously, the starting point is revisited. That is, a contour is a circularly ordered points list. A contour segment curvature (CSC) is defined as the ratio of the line length connecting two endpoints of a contour segment to the curve length of the segment. As shown in equation (2), there are n distinct contour segments length L having midpoint P_i and two endpoints A_i and

B_i (n is the number of points composing contour). The $C(P_i)$ represents the CSC at point P_i and it also has a value between 0 to 1.

$$C(P_i) = \frac{Length(\overline{A_i B_i})}{L}, \text{ where } i = 1..n \tag{2}$$

As described in Fig. 1, the CSC of P_1 has value near to 0 because segment S_1 has high fluctuation. Whereas the CSC of P_3 has value near to 1 because segment S_3 is nearly a straight line. The CSC at point P_i is invariant to rotation and scale and it can represent local or global features of contour with respect to the segment length. Although two contour segments S_2 and S_3 are defined at the same point P_2, the CSC of segment S_2 describe the local features and the CSC of segment S_3 describe the global features. That is, the two CSCs at P_2 represent local straightness and global curve. In comparison with that, the two CSCs at P_3 represent local and global straightness.

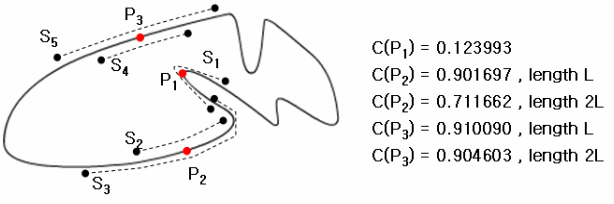


Fig. 1. Contour segments and CSCs

2.3 Coordinates Transformation

This paper proposes a new coordinates system with two orthogonal axes: the *d*-axis and the *c*-axis. The *D*-axis represents the relative distance described in section 2.1, and the *C*-axis describes the CSC described in section 2.2. The plane defined on the new coordinate system is named as *d*-*c* plane. Both values of the relative distance and the CSC are in the range of 0 to 1. Therefore the *d*-*c* plane is a unit plane of which area is 1. A point $P(fd, fc)$ in the *d*-*c* plane is identified by two real numbers *fd* and *fc*. It is convenient to use integers instead of real numbers to indicate a point. Thus we have extended the plane by multiplying constant C_p as described in equation (3).

$$d = fd \times C_p + 0.5, \quad c = f_C \times C_p + 0.5 \tag{3}$$

Now a point $P(d, c)$ can be pointed with two integers d and c. If the C_p is not sufficiently large, the mapping from the *x*-*y* plane to the *d*-*c* plane may be many-to-one. In this study, the many-to-one mapping is not a problem because the purpose of the mapping is to extract features invariant to rotation and scale. In contrast, the C_p can be used to set the resolution of the feature space. Fig. 2 shows an example of mapping from the x-y plane to the d-c plane. In this example, the segment length L was determined as the twofold of line CP_i length to compute the CSC at point P_i. The nearest points from the centroid such as P_2, P_4, P_6, P_8 have been mapped to P_a, and the farthest points from the centroid such as P_1, P_3, P_5, P_7 have been mapped to P_b. The other points between the midpoint on the edge and the vertex have been mapped between P_a and P_b.

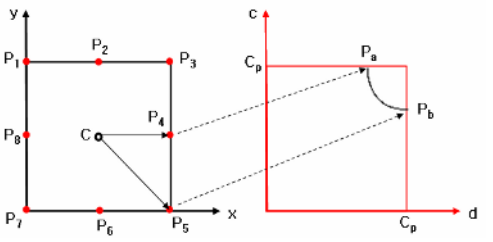


Fig. 2. Mapping of a perfect square from the *x*-*y* plane to the *d*-*c* plane

Fig. 3 shows transformation examples of a circle, a perfect square, and an irregular shape. All the points on a circle have an equal distance from the centroid and an equal CSC, thus the transformed result appears as a single point on the *d*-*c* plane. Theoretically, all circles of different size are transformed as a single point. However, they can be discriminated because each has a different accumulated value. Fig. 4 shows the transformed results of rotated or scaled ellipses. The scaled or rotated shapes on the *x*-*y* plane appear to be almost same shape on the *d*-*c* plane. It proves that the proposed coordinates system represents rotation and scale invariant feature space.

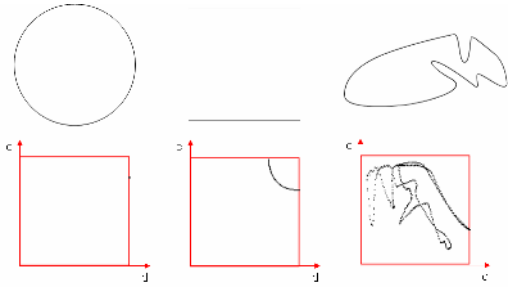


Fig. 3. Transformation of a circle, a perfect square, and an irregular shape

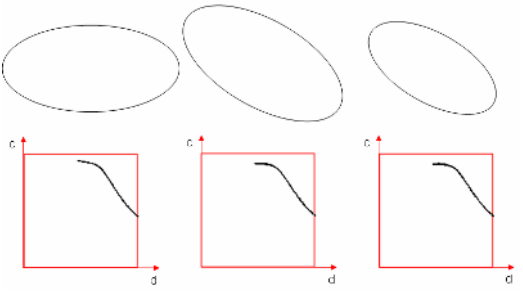


Fig. 4. Transformation of rotated or scaled ellipses

3 Contour Shape Description and Recognition

3.1 Mapping to D-C Plane

When computing the relative distance and the CSC at point P_i, the centroid becomes a base point. As described in section 2.3, the max distance from the centroid is the norm distance for calculating the relative distance. Additionally, the twofold of line CP_i length is the norm segment length for calculating the CSC at P_i. Needless to say, the base point has a dominant role in the process of mapping to the *d-c* plane. A centroid is unique in a simple shape composed of single contour. However, two-dimensional shape usually contains many contours. In this case, there are two kinds of centroids: local centroid and global centroid. The local one is the centroid of each contour, whereas the global one is the centroid of entire contours.

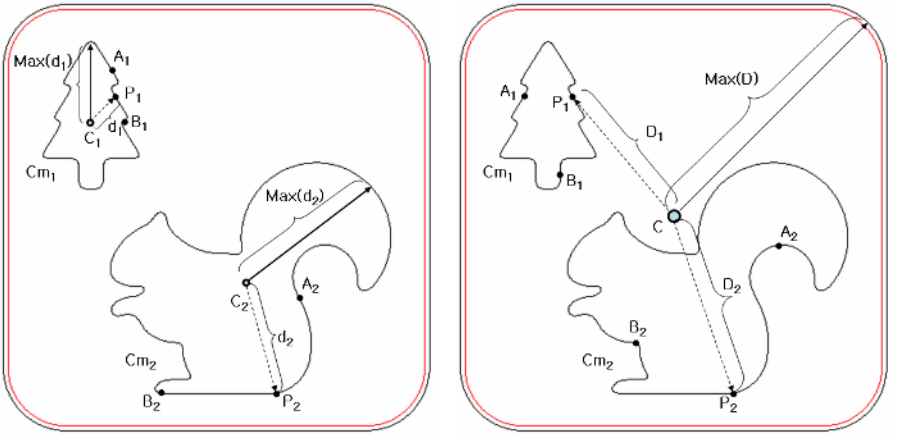
According to the centroid, the transformed *d-c* plane has different properties. The transformed *d-c* plane based on the local centroids represents the properties of only each contour shape, whereas the *d-c* plane based on the global one describes not only the shape information with a global view but also the positional information among the contours. Two *d-c* planes (local plane and global plane) are generated from a single *x-y* plane. In the local plane, the local centroid becomes the base point for calculation the relative distance and the CSC. In the global plane, only the global centroid becomes the base point. The local plane represents features extracted from the individual viewpoint of each component and the global plane represents features extracted from the overall viewpoint. Fig. 5 and Table 1 illustrates how to calculate the relative distance and the CSC for mapping the local or global plane. The pictogram shown in Fig. 5 means the nature conservation. It has total four contours: three outer contours and one inner contour. Only two components were marked as Cm_1 and Cm_2 for brief display. In Table 1, $Max(d_1)$ is the farthest distance from the local centroid C_1 to the points on the contour of Cm_1, whereas $Max(D)$ is the farthest distance from the global centroid C to the points on the all the contours. The contour segment length at point P_i is defined as the distance from the relevant centroid to the point P_i. So the length of segment A_1B_1 having midpoint P_1 in Fig. 5-(a) shorter than that of Fig. 5-(b). As a result, the segment A_1B_1 at Fig. 5-(b) is more curved than the segment A_1B_1 at Fig. 5-(a). That is, feature extracted from the same point represents different properties according to the base point.

3.2 Contour Shape Description and Matching

Mesh feature is extracted from these two *d-c* planes and they are used to describe contour shape. In the early days, the mesh feature was one of the traditional features widely used in the area of statistical pattern recognition. Its sensitivity to rotation variation is the disadvantage of the mesh feature. However, the mesh feature on the *d-c* plane is invariant to rotation as the rotated contour shape is mapped into the same coordinates on the *d-c* plane. The plane is divided with $m \times n$ meshes to extract statistical features, and subsequently the number of points in each mesh is counted. The mesh feature is normalized by dividing the number of points in each mesh by the total number of points and is converted into integer by multiplying constant C. The feature

vector is used to describe contour shape. It is calculated by the equation (4), where N_{ij} represents the number of points in each mesh. Now, the problem of contour shape matching can be solved through the comparison of feature vectors. The distance between two feature vectors can be computed easily through matrix subtraction.

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{bmatrix}, \text{ where } v_{ij} = \frac{N_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} N_{ij}} \times C \tag{4}$$



(a) Base points for local plane (b) Base point for global plane

Fig. 5. Base points for the local plane and the global plane

Table 1. The calcuation of the relative distance and the CSC in the two *d-c* planes

		Relative distance: $D(P)$	CSC: $C(P)$
Local plane	P_1	$D(P_1) = \frac{Length(\overline{C_1P_1})}{Max(d_1)}$	$C(P_1) = \frac{Length(\overline{A_1B_1})}{2 \times d_1}$
	P_2	$D(P_2) = \frac{Length(\overline{C_2P_2})}{Max(d_2)}$	$C(P_2) = \frac{Length(\overline{A_2B_2})}{2 \times d_2}$
Global plane	P_1	$D(P_1) = \frac{Length(\overline{CP_1})}{Max(D)}$	$C(P_1) = \frac{Length(\overline{A_1B_1})}{2 \times D_1}$
	P_2	$D(P_2) = \frac{Length(\overline{CP_2})}{Max(D)}$	$C(P_2) = \frac{Length(\overline{A_2B_2})}{2 \times D_2}$

4 Experimental Results and Analysis

4.1 Experimental Environment

The proposed contour shape description method has been applied to pictogram recognition. A full set of Korean national pictogram standard was used. The standard includes 302 pictograms that are categorized into two groups: facility-related pictograms and safety-related pictograms. Each group is further categorized down into 5 subgroups. The information about the pictogram images is summarized in Table 2. Fig. 6 shows some sample images selected from each subgroup.

Table 2. Summary of pictogram images

Group	Subgroup	Color	# of pictogram
Pictogram related with facility	Public facilities	BW (color 1)	86
	Transport facilities	BW (color 2)	29
	Commercial facilities	BW	21
	Tourism and cultural facilities	BW	33
	Sports facilities	BW	42
Pictogram related with safety	Safe condition	Green	5
	Fire safety and emergency	Red	5
	Prohibition	Red, Black	39
	Warning	Yellow, Black	27
	Mandatory action	Blue	15

Fig. 6. Examples of pictogram images

Most of the facility-related pictograms are black and white images. In contrast, safety-related pictograms have additional color that is closely related to each subgroup. In this experiment, only shape information was used for pictogram recognition. To test the robustness to the variation of rotation and scale, five sets of rotated images and two sets of scaled images are generated. The proposed method was implemented at Pentium4 PC using Visual C++ 6.0.

4.2 Experimental Results and Analysis

Contours were extracted based on 8-neighbor connectivity. The extracted contours can be partially disrupted by noise or variation. Especially, the CSC is sensitive to the tiny fluctuation of contour. The contour smoothing process based on Gaussian convolution mask was performed to remove the unfavorable distortion.

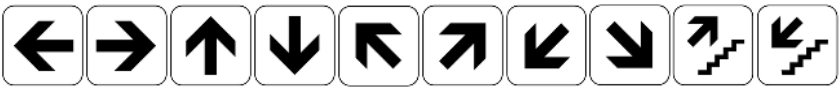
It is important to find a proper mesh size to describe the contour shape precisely. For this purpose, three different meshes (4 × 4 mesh, 8 × 8 mesh, and 16 × 16 mesh) were tested in this experiment. If m is too small, the $m \times m$ mesh can't discriminate fine differences. In contrast, if m is too large it becomes too sensitive to noise or distortion. Table 3 shows the recognition result based on the three different meshes.

Table 3. Pictogram recognition results using m × m mesh

4×4 Mesh		# of miss	Correct recognition	# of miss among same shape group			Practical correct recognition
				a~d	e~h	i~ j	
Scale	80%	3	99.01%	0	1	0	99.34%
	130%	3	99.01%	2	1	0	100.0%
Rotation	3º	3	99.01%	2	1	0	100.0%
	7º	3	99.01%	2	0	0	99.67%
	13º	6	98.01%	2	2	0	99.34%
	23º	14	95.36%	2	0	1	96.36%
	37º	8	94.04%	0	0	1	94.37%

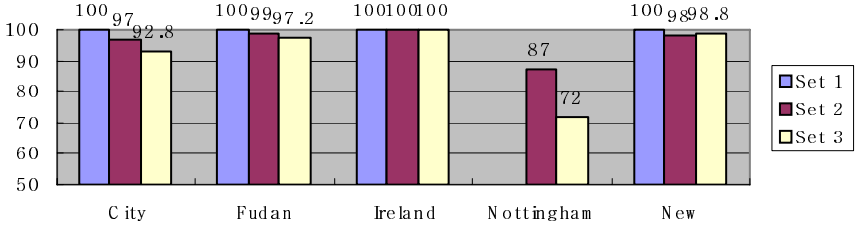
8×8 Mesh		# of miss	Correct recognition	# of miss among same shape group			Practical correct recognition
				a~d	e~h	i~ j	
Scale	80%	1	99.67%	0	1	0	100.0%
	130%	4	98.68%	2	2	0	100.0%
Rotation	3º	2	99.34%	1	1	0	100.0%
	7º	3	99.01%	2	1	0	100.0%
	13º	4	98.68%	2	2	0	100.0%
	23º	5	98.34%	2	3	1	100.0%
	37º	9	97.02%	0	0	1	97.35%

16×16 Mesh		# of miss	Correct recognition	# of miss among same shape group			Practical correct recognition
				a~d	e~h	i~ j	
Scale	80%	0	100.0%	0	0	0	100.0%
	130%	6	98.01%	1	2	0	99.01%
Rotation	3º	1	99.67%	1	0	0	100.0%
	7º	6	98.01%	2	3	0	99.67%
	13º	9	97.02%	2	3	1	99.01%
	23º	16	94.70%	0	3	1	96.03%
	37º	13	95.70%	0	0	0	95.70%



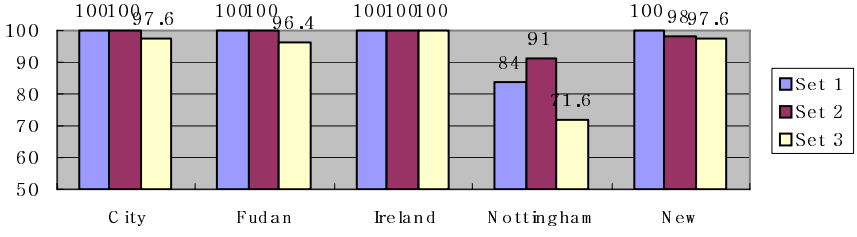
(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 7. Pictograms regarded as the same: (a)~(d), (e)~(h), and (i)~(j)



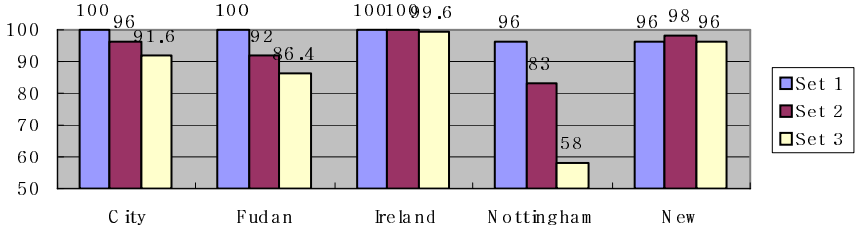
T

(a) Tests with the rotated symbols



T

(b) Tests with the scaled symbols



(c) Tests with the rotated and scaled symbols

Fig. 8. Recognition results for the GREC 2003 symbol contest data

The 8 × 8 mesh was proved to be the best among them. There are eight pathway direction signs and two stair direction signs as described in Fig. 7. It is impossible to discriminate the first four signs (a)~(d) when rotation occurs. The next four signs (e)~(h) and last two (i)~(j) are the same. The last column in Table 3 represents the

recognition rate with the exception of the inevitable errors that occurred within the same shape group.

Although the simple matrix subtraction and the nearest neighbour classifier have been used for feature matching, excellent performance was acquired. This result confirms that the proposed method is robust to rotation and scale variation. To compare the performance of the proposed method with others', one additional experiment using the test data once employed in the GREC 2003 symbol contest [13] was performed. In this experiment, the 8 × 8 mesh was applied and three sets of rotated or scaled test data were used: set 1 (5 kinds of 25 symbols), set 2 (20 kinds of 100 symbols), and set 3 (50 kinds of 250 symbols). As shown in Fig. 8, although the result was not exceptional, the proposed method denoted 'New' showed the high recognition results. It also keeps the good recognition result against the rotated and scaled variation as shown in Fig. 8-(c). What is more important is that this additional experiment was performed without any tuning for a new domain.

5 Conclusion

This paper has proposed a shape description method via transformation from the image space into the feature space having *d*- and *c*-axes: representing the relative distance from a centroid and the contour segment curvature (CSC) respectively. A point on the *d*-*c* plane based on the two geometric invariants can be fixed regardless of rotation or scale variation. The relative distance describes how far a point departs from the centroid, and the CSC represents the degree of fluctuation in a contour segment. After transformation, the mesh feature was used to describe the shape mapped onto the *d*-*c* plane. Traditional mesh features extracted from the *x*-*y* plane are sensitive to rotation, whereas the mesh features from the *d*-*c* plane are robust to it.

To guarantee high performance, it is important to extract clean contour shape because the method is a contour-based descriptor. The experimental data are relatively clean. Therefore a good recognition result could be acquired although the simple similarity measure and the nearest neighbour classifier were used. In the following study, to recognize real pictograms extracted from outdoor scenes, further exploration on pre-processing such as image enhancement, noise removal, and searching the location of pictogram will be continued.

References

1. D. Zhang and G. Lu: Review of shape representation and description techniques. *Pattern Recognition*, 37 (2004) 1-19
2. M. Safar, C. Shahabi, and X. Sun: Image Retrieval by Shape: A Comparative Study. *Proc. of the IEEE International Conference on Multimedia and Expo*(I) (2000) 141-154
3. S. Loncaric: A Survey of Shape Analysis Techniques. *Pattern Recognition*, 31(8) (1998) 983-1001
4. Z. Li Stan: Shape Matching Based on Invariants. In O.M. Omidvar (ed.), *Progress in Neural networks: Shape Analysis*, 6 (1998) 203-228

5. M.K. Hu: Visual Pattern Recognition by Moment Invariants. *IRE Trans. on Information Theory 8* (1962) 179-187
6. L. Keyes and A. Winstanley: Using moment invariants for classifying shapes on large-scale maps. *Computers, Environment and Urban Systems*, 25 (2001) 119-130
7. K. Tsirikolias and B.G. Mertzios: Statistical Pattern Recognition using Efficient Two-Dimensional Moments with Applications to Character Recognition. *Pattern Recognition*, 26(6) (1993) 877-882
8. D. Shen and H.S. IP Horace: Discriminative wavelet shape descriptors for recognition of 2-D patterns. *Pattern Recognition*, 32 (1999) 151-165
9. C.C. Chang, S.M. Hwang, and D.J. Buehrer: Shape Recognition Scheme Based on Relative Distances of Feature Points from the Centroid. *Pattern Recognition*, 24(11) (1991) 1053-1063
10. J. Zhang, X. Zhang, H. Krim, and G.G. Walter: Object representation and recognition in shape spaces. *Pattern Recognition*, 36 (2003) 1143-1154
11. S. Manay, B. Hong, A.J. Yezzi, and S. Soatto, Integral Invariant Signatures. T. Pajdla and J. Matas (Eds.): *ECCV 2004,* LNCS 3024 (2004) 87-99
12. M. Kliot and E. Rivlin: Invariant-Based Shape Retrieval in Pictorial Databases. *Computer Vision and Image Understanding*, 71(2) (1998) 182-197
13. E. Valveny and P. Dosch: Symbol Recognition Contest: A Synthesis. J. Llados and Y.B. Kwon (Eds.):In Graphics Recognition: Recent Advances and Perspectives (2004) 368-385

Feature Detection from Illustration of Time-Series Data

Tetsuya Takezawa and Toyohide Watanabe

Department of Systems and Social Informatics
Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan
{takez, watanabe}@watanabe.ss.is.nagoya-u.ac.jp

Abstract. We propose a method for extracting the geometric feature and the comprehensive fluctuation from time-series data and also a method for detecting a reference sequence effectively on the basis of the distance graph. The prevalent methods such as one based on the frequency characteristics do not deal with time-series data in the time dimention. Therefore, our method for extracting the features is temporally sensitive to fluctuations of time-series data. We experimented using the time-series data whose frequency bands were changed variously in order to make clear the availability of the proposal procedures such as smoothing and encoding.

Keywords: Geometric feature, detection of reference sequence, time-series data.

1 Introduction

Time-series data are time-dependently generated from the observation of various activities. They include temporal features which correspond to the states of observed activities. Analyzing time-series data makes it possible to predict the future states of observed activities. Detecting a particular sequence on time-series data is a primary process of data analysis or data mining on time-series data (In this paper, the particular sequence to be detected is called "reference sequence.").

For example, an electrocardiogram or a stock chart make clear a geometric feature of the time-series data by representing sequential values as a figure. In these cases, it is required to detect not only the partial sequences which correspond to the reference sequences but also the partial sequences which are similar to the reference sequence geometrically. When time-series data are compared, the geometric features and comprehensive fluctuations of time-series data are especially focused on. The geometric feature means the wave shape which represents the relation between each point of time-series data. And the comprehensive fluctuation means the main fluctuation which almost consists of the low-frequency components. Many researchers have proposed methods for detecting partial sequences from time-series data. These methods include digital filters and spectrum analysis, and are based on the frequency characteristics[1, 2, 3]. The smoothing process is thought to be achieved by extracting the low-frequency components or eliminating the high-frequency components. However, extracting or eliminating particular frequency components may drastically change the wave shape, since the wave shape consists of various frequency components. Furthermore, because the frequency characteristics are defined in terms of a certain time interval, the time resolution

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 323–333, 2006.
© Springer-Verlag Berlin Heidelberg 2006

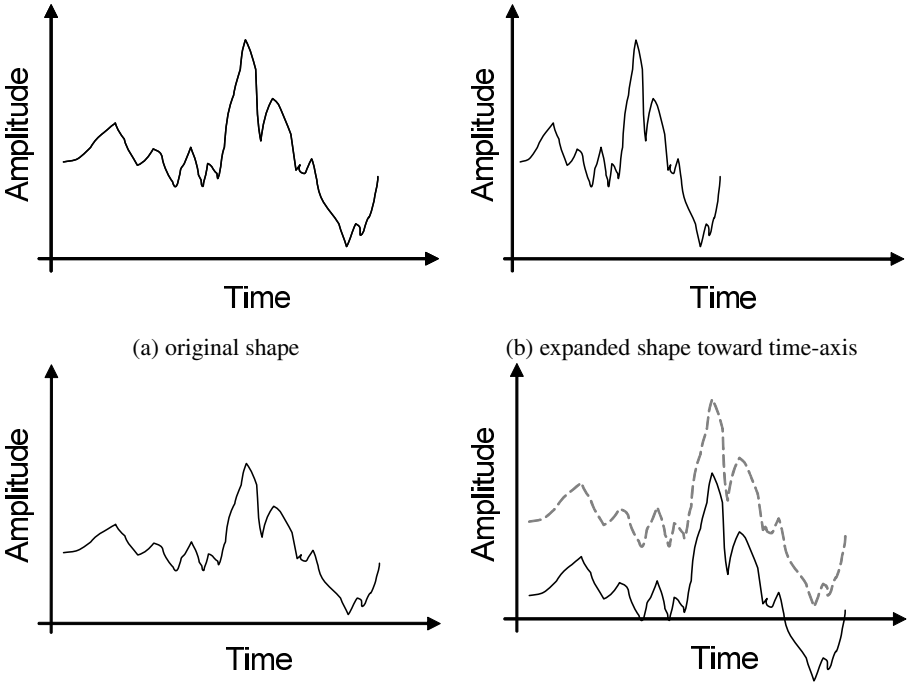
of the frequency characteristics is essentially low. In this paper, we address a method for detecting a reference sequence on the basis of the geometric features and comprehensive fluctuations of time-series data. Especially, we discuss the relation between the smoothness and the feature retainment in terms of smoothing method which is sensitive to fluctuations of time-series data.

The layout of this paper is as follows. Our approach is shown in Section 2. And the extraction method is described in Section 3. Section 4 presents some experimental results to make clear the availability of our method. Finally, we give some conclusions in Section 5.

2 Framework

2.1 Geometric Feature

When time-series data are compared in point of the shape like time-depentdent flow, the geometric features and comprehensive fluctuations of time-series data are especially focused on. The geometric features should be dealt without amplitude-axis and time-axis in order to compare the relation between the values of time-series data. Namely, we define that the reference sequence is geometrically similar to the partial sequences which are transformed by the following operations.



(a) original shape (b) expanded shape toward time-axis

(c) contracted shape toward amplitude-axis (d) shifted shape toward amplitude-axis

Fig. 1. Transformation of time-series data

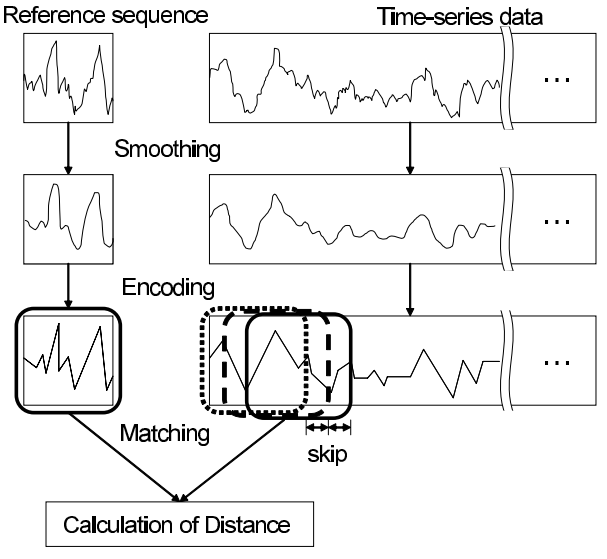


Fig. 2. Processing flow

1. expansion/contraction toward time-axis
2. expansion/contraction toward amplitude-axis
3. shift toward amplitude-axis

DTW (Dynamic Time Warping) [4] is a method for estimating the dissimilarity between sequences which are nonlinearly expanded/contracted toward time-axis. In this paper, we discuss a procedure for extracting the geometric feature from time-series data which do not depend on the transformations of time-axis and amplitude-axis. It is important to deal with time-series data in the time dimension in order to prevent the geometric feature. Thus, we adopt a line approximation which is one of the most primitive factors for the temporal feature of fluctuation. The time-interval length of each line segment must be decided respectively and dynamically because the most important factor for extracting the geometric feature is sensitivity to temporal fluctuations. In order to deal with the time-series data shifted toward amplitude-axis, the wave shape is composed of various types of line segments. And the values of each line segment are represented relatively as the shapes are expanded/contracted toward amplitude-axis.

2.2 Processing Flow

The processing flow in our method is shown in Fig.2. The process in our method is composed of the following steps:

1. smoothing time-series data,
2. encoding the smoothed sequence, and
3. matching a reference sequence using the encoded sequence.

The smoothing step is most important for our detection process because a procedure for smoothing time-series data influences synergistically the following steps. Therefore, the smoothing procedure which not only eliminates the noises but also keeps the feature of time-series data is required. In the encoding step, in order to extract the geometric relation between individual points from time-series data, a smoothed sequence is converted into the approximated lines. Furthermore, the line segments are categorized into nine directions according to the gradient of the lines to represent the approximated lines relatively. The time-interval length of line segments should be decided dynamically because the fluctuation of time-series data is not fixed. Therefore, the temporal lengths of line segments are decided dynamically on the basis of the approximation error. In the matching step, in order to improve the effectiveness of detection, the distance calculations are partly skipped. However, the time-interval length to be skipped is different according to the reference sequences. Therefore, we focus on the distance fluctuation (We call it "distance graph") which is made by estimating the distance of the partial sequences near to the reference sequence.

3 Detection of Reference Sequence

3.1 Smoothing

In this section, we discuss a procedure for extracting the comprehensive fluctuations from time-series data. It is important to retain the amplitude in the smoothing step because the attenuation of amplitude means the loss of the geometric feature.

Moving average method is a traditional one to extract the comprehensive fluctuation continuously. In the moving average method, a value to be smoothed is estimated by the average of some values near to the current point. The smoothed value of each point x'_i is given by the following expression,

$$x'_i = \sum_{k=-W}^{W} \frac{x_{i+k}}{2W+1}, \tag{1}$$

where x_i is the value of each point and W is the time-interval length of the data. The smoothness of the extracted comprehensive fluctuation depends on W. The larger the value of W is, the more sequences become smooth. However, there is a trade-off relation between the smoothness of the comprehensive fluctuation and the retainment of amplitude. Namely, if the value of W is large, sequences are smoothed strongly but the amplitude of sequence is attenuated severely. Therefore, we apply the moving average method several times using a small value for W. To apply the moving average method repeatedly corresponds to WMA (Weighted Moving Average). And the weights are determined by R (the repeating count of applying the moving average method) and W. As shown in Fig.3, the sequence smoothed with a large W is attenuated severely, but the sequence which smoothed repeatedly with a small W is not so.

3.2 Encoding

In the encoding step, the geometric features are extracted from time-series data. It is important to extract the features from a segment of a sequence, not from one point,

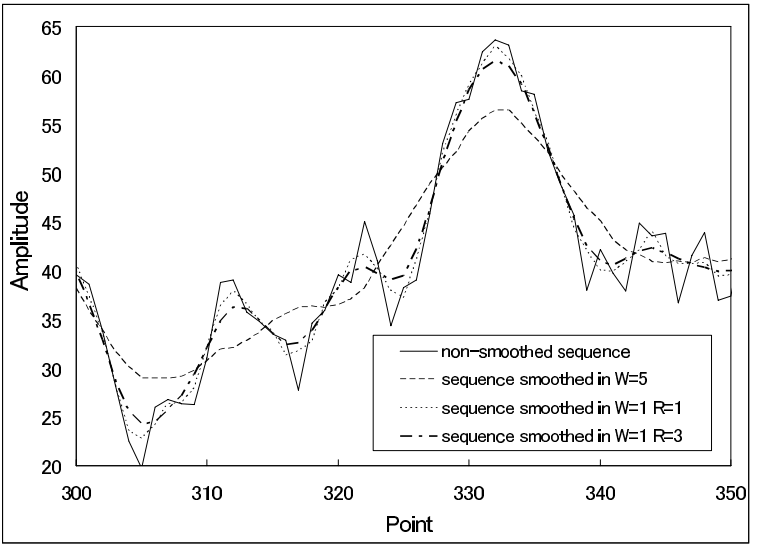


Fig. 3. Repeating count and attenuation

because the geometric feature is composed of some sequential points. Furthermore, the geometric features of sequences should be compared independently from the values of each point. Therefore, we utilize the line approximation for encoding time-series data.

We define a line segment which approximates a segment of time-series as a component unit of geometric feature (Fig.4). However, there is a problem for generating approximated lines. The interval of time-series data whose values change drastically need to be approximated by many line segments in comparison with interval whose values change slowly. Therefore, time-interval lengths of segment lines are determined on the basis of the approximation error. An approximated line is given by the following expression,

$$y = a \times i + b, \tag{2}$$

where a is a gradient and b is an intercept of an approximated line. And approximation error ϵ is defined as the following expression,

$$\epsilon = \sum_{i=i_s}^{i=i_e} (a \times i + b - x_i), \tag{3}$$

where x_i is each value of sequence in the interval ($i_s \leqq i < i_e$).

A procedure for generating approximated lines consecutively is shown in Fig.5. In order to construct the geometric relation by approximated line segments, it is important to deal with each line segment relatively. Namely, magnitude correlation of each gradient is a meaningful feature of figure. Therefore, line segments are classified into nine directions according to the relative value of the gradient (Fig.6). The geometric feature without amplitude axis can be represented by classification in the directions. In this step, Smoothed time-series data are converted into a vector sequence shown in Fig.4.

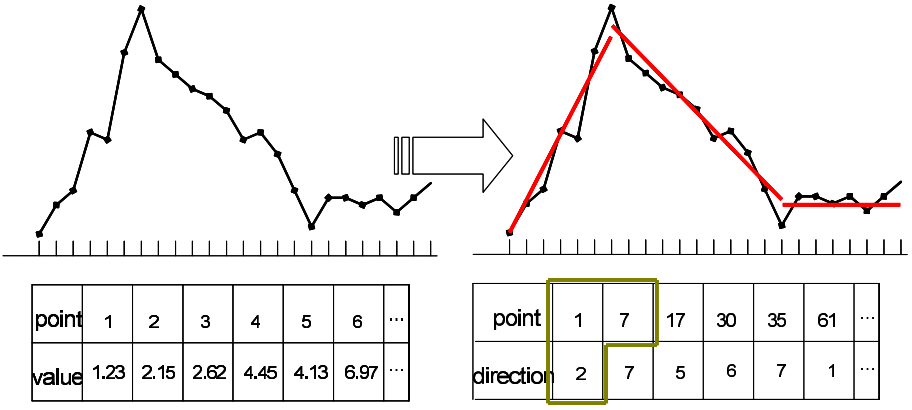


Fig. 4. Encoding time-series data

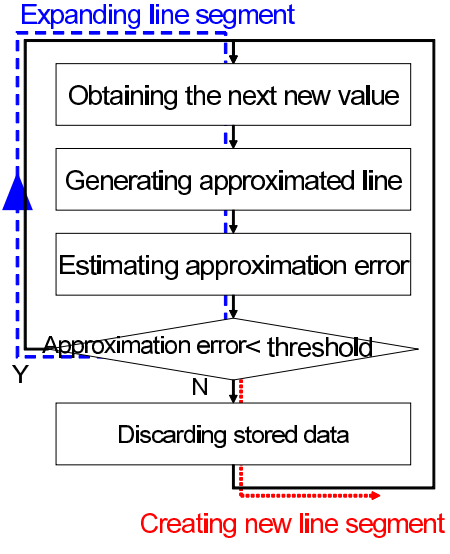


Fig. 5. Line approximation

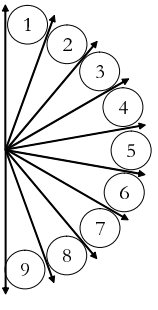


Fig. 6. Direction of vector

3.3 Matching

We propose a method for detecting a partial sequence, which is similar to a reference sequence geometrically, from time-series data. Here, we assume that objects to be detected are the same time-interval length as the reference sequence. We describe the calculation of distance which represents the dissimilarity between sequences. And we explain a skip procedure of the distance calculation to improve the effectiveness of detection.

3.4 Distance Between Geometric Features

In order to compare the geometric feature of time-series data, the compositive line segments are compared. However, the time-interval length of each line segment is different respectively. As shown in Fig.7, line segments are separated according to the ends of another sequence. The distance is calculated by the sum of differences in each time-interval which are decided by the ends of line segments and definitional equation of distance is shown below,

$$Dis = \frac{\sum_{i=0}^{N}(|D_{1i} - D_{2i}|W_i)}{\sum_{i=0}^{N} W_i}, \tag{4}$$

where D_{is} is the distance between the shapes of two sequences, D_{1i} and D_{2i} are two directions of individual line segments, and W_i is a time-interval length. To detect the partial sequence which corresponds to a certain reference sequence means to find out the partial sequence whose distance is less than a certain threshold:

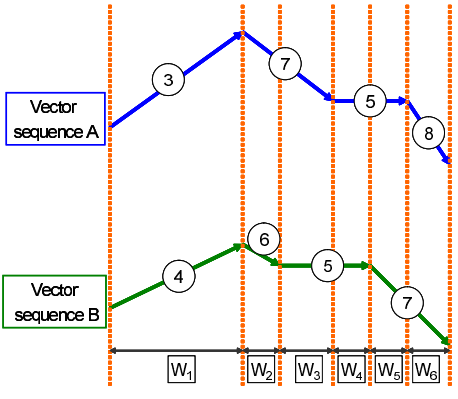


Fig. 7. Separation of line segments

3.5 Skip of Distance Calculation

Stored time-series data, generally speaking, are much larger than reference sequences. If partial sequences, whose time-interval length is the same as the reference sequence, are segmented from time-series data shifting its time-interval point-by-point, the number of distance calculation is almost the same as the number of time-series data points. However, the distance calculations for partial sequences in every interval are not necessary. Namely, the partial sequences which are similar to the reference sequence can be detected even if the distance calculations of some partial sequences are skipped. There are two important factors to improve the effectiveness of detection. One is whether the partial sequences which correspond to the reference sequence can be found out certainly. Another is the rate that the number of distance calculations decreases.

If the reference sequence is on time-series data, the distance of the interval where reference sequence exists must be approximately zero. And the distance owing to partial sequences near to the reference sequence is also very small. However, there is just one reference sequence even if the distance between some partial sequences were very small. The reason that the distance is small is that the partial sequences consist almost of the reference sequence. The fact that there may be the reference sequence near to the current partial sequence becomes clear by just calculating the distance which is smaller than the threshold. Therefore, the number of the distance calculation can be reduced by skipping the partial sequences.

Next, we discuss the time-interval length where the distance calculation can be skipped. If the time-interval length of skip is long, the number of the distance calculation decreases but the reference sequence may not be detected. Thus, it is required to estimate the maximum time-interval length of skip with which the reference sequence can be detected certainly.

In order to estimate the time-interval length of skip, we define "distance graph" and "time-interval length of detection". The distance graph is the fluctuation of the distance which is calculated by shifting the interval of partial sequences point-by-point (Fig.8). The time-interval length of detection is the length of time-interval where the distance is smaller than the threshold. The time-interval length of detection for distance graph which is made by the distance fluctuation near to the reference sequence is adopted as the time-interval length of skip.

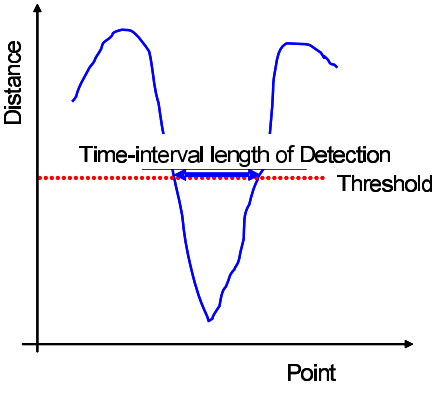


Fig. 8. Distance graph

4 Experiment

Our detection method is too much dependent on a frequency band of time-series data. Therefore, we experimented by the time-series data whose frequency band is varied to make clear the availability of the proposal method. We made clear the relation between the availability of the smoothing and encoding procedures and the frequency band of sequences. However, we assumed that the sampling frequency is $1[kHz]$ for simplification.

4.1 Generation of Sequences

A sequence is composed of several waves whose frequencies are different respectively and a wave (f[Hz]) is given by the following expression.

$$s_{fi} = a \times \sin(\omega i + \theta) \tag{5}$$

$$a = a' \times RANDOM \tag{6}$$

$$\omega = \frac{2\pi}{NUM_{seq}} \times f \tag{7}$$

$$\theta = 2\pi \times RANDOM \tag{8}$$

In the expression (5), a is the amplitude, s_{fi} is a value of sequence (i is the time, and f is the frequency), ω is the angular velocity, and θ is the phase. In the expression (7), Num_{seq} is the point number of the sequence. Furthermore, the sequence which has the main frequency band ($f_s < f < f_e$) is generated by the following expression.

$$s_i = \sum_{f=f_s}^{f_e} s_{fi} \tag{9}$$

Table 1. Set value for generating sequences

	reference sequence	noise sequence
NUM_{seq}	1000	1000
frequency band[Hz]	$1 \sim 10$	$1 \sim 10, \cdots, 91 \sim 100$
a'	15.0	$1.5, 3.0, \cdots, 15.0$

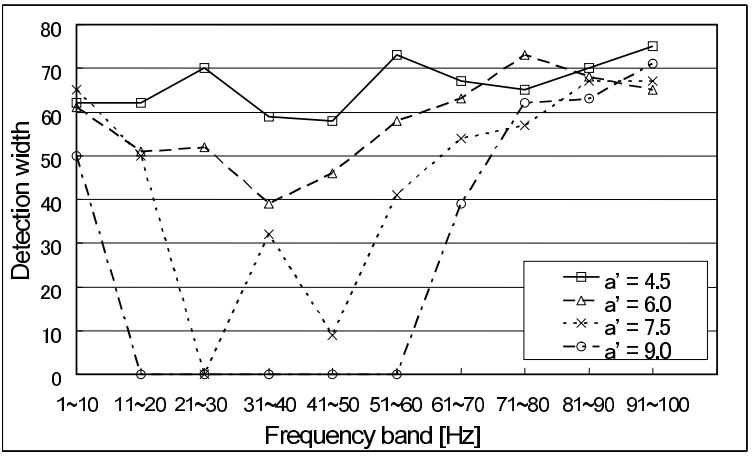


Fig. 9. Time-interval length of detection for similar sequences

The reference sequence whose frequency band is $1 \sim 10$ is generated. The noise sequences whose frequency band and amplitude are ten kinds individually are also generated. The set values which are used in generating sequences are shown in Table1.

4.2 Method of Experiment

One reference sequence was generated. And hundred noise sequences whose kinds are hundred totally were also generated. The similar sequence which has the difference with the reference sequence was made by adding a noise sequence to the reference sequence. The reference sequence and similar sequences which had been generated were smoothed and encoded. Next, the distance graph of the reference sequence and each similar sequence was made and the time-interval length of detection was estimated. The average values of time-interval length of detection according to hundred kinds of similar sequences were calculated respectively.

4.3 Result of Experiment

The time-interval length of detection is summarized (amplitude of the noise sequences is respectively 4.5, 6.0, 7.5, 9.0) in Fig.9. The time-interval length of detection, decreased as the amplitude of the noise sequence, were expanded. However, the time-interval length of detection on the frequency band $1 \sim 10$ and $70 \sim 100$ decreased slowly.

4.4 Consideration of Experiment

The time-interval length of detection of the distance graph which is drawn by the reference sequence and the noise sequence means the similarity of two sequences. Therefore, as the result of experiment, it was made clear that the difference between encoded sequences at the low frequency or high frequency does not appear clearly. The reasons are interpreted that the high frequency component on a sequence is attenuated by smoothing and the fluctuations of a low frequency sequence is essentially slow.

5 Conclusion

In this paper, we proposed a method for extracting geometric feature on time-series data and a method for detecting a reference sequence on the basis of distance graph. We considered that the status of observed objects appeared as the wave shape. The procedure of extracting the geometric feature was composed of two steps: smoothing and encoding. The time-series data was smoothed by applying the moving average method repeatedly and our proposal method achieved a balance between smoothness of time-series data and feature retainment. The geometric features estimated by smoothing were approximated by line segments in order to extract the fluctuation such as increasing or decreasing. Furthermore, line segments which compose the shape of time-series data were classified into nine directions in order to represent the wave shape relatively. It was important to detect a reference sequence effectively because time-series data for various activities is being generated and the amount is essentially large. Thus, the procedure for

skipping the distance calculation was proposed. In order to make clear the relation between availability in terms of smoothing and encoding method and the frequency band of time-series data, the result of experiment was shown. As a result, it became clear that the feature on low-frequency and high-frequency components of time-series data does not appear as the geometric feature.

In the future work, we must evaluate our method using the time-series data which are observed from actual activities. Furthermore, methods for calculating the distance based on DTW should also be investigated.

Acknowledgements

The authors would like to thank the 21^{st} Century COE(Center of Excellence) Program for 2002, a project titled Intelligent Media (Speech and Images) Integration for Social Information Infrastructure, proposed by Nagoya University. And we also would like to thank Mr.Shimanuki who is currently on doctorial candidate in our university for his kindly support and advice and also our research members.

References

1. Davood Rafiei and Alberto Mendelzon, "Similarity-based Queries for Time Series Data", *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, pp.13-25, (1997).
2. Juan J. Rodriguez and Carlos J. Alonso, "Interval and Dynamic Time Warping-based Decision Trees", *Proceedings of the 2004 ACM Symposium on Applied Computing*, pp.548–552, (2004).
3. Juan P. Caraca-Valente and Ignacio Lopez-Chavarrias, "Discovering Similar Patterns in Time Series", *Proceedings of the 2000 ACM Symposium on Applied Computing*, pp.497–505, (2000).
4. Donald J. Berndt and James Clifford, "Finding Patterns in Time Series: A Dynamic Programming Approach", *Advances in Knowledge Discovery and Data Mining*, pp.229-248, (2000).

Sketch Parameterization Using Curve Approximation

Zhengxing Sun, Wei Wang, Lisha Zhang, and Jing Liu

State Key Lab for Novel Software Technology, Nanjing University, P. R. China, 210093
szx@nju.edu.cn

Abstract. This paper presents a method of parameterization for online freehand drawing objects based on a piecewise cubic Bezier curve approximation. The target is to represent sketches in a compact format within a certain error tolerance with lower computation to be practically adaptable for the online graphics input. A set of user's intended breakpoints in digital ink is firstly produced in terms of pen speed and local curvatures. Each of strokes of a skechy shape is then parameterized by the optimization of piecewise Bezier curve approximation to minimize the fitting error between stroke path and the curve. The experimental results show both effective and efficient for a wide range of drawing graphic objects.

1 Introduction

As computers become integrated into everyday life, pen-based user interface is considered as a primary input method. Moreover, the feature to rapidly visualize and deliver human's ideas using graphic objects, which cannot be efficiently represented by speech or text, is highly desirable in graphic computing [1]. The rapid growth of graphic data has sustained the need for more efficient ways to represent and compress the sketchy graphic data. The data representing freehand sketching needs not only to be compressed in order to reduce the internal handling size and to transfer in low bandwidth, but also to preserve the original intention of user and the convenience of easy access of the information and for further processing such as shape recognition, cooperative design, idea permutation and so on.

For existing techniques, the pen movements are typically captured by a digitizing tablet and stored as sampled pen points of their paths, so called as digital ink, while an image for receptor is captured. The drawback of this technique is that sketches transferred in image usually require considerable storage capacity and cannot be modified by receptor. Although there have been a large amount of experiments on sketchy graphics recognition, such as feature-based [2][3], graph-based [4][5], machine learning [6][7][8] and Parametric methods such as polygon [9], B-spline [10] and Bezier curve [11], most of them guess and convert the drawing sketches into regular shapes. However, they are charged with desertion of users' intension, and are the burden of computation especially for mobile devices. Parametric methods fitting techniques have been considered in shape representation and classification. A benefit of these approaches is that they can approximate the path of pen movements during user drawing with a few parameters and they are computationally efficient. Only a few researches have bent themselves to this issue [11][12].

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 334 – 345, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this paper, a sketch approximating method is introduced, which performs the efficient parameterization of on-line freehand drawing objects captured in digital ink using a piecewise recursive cubic Bezier curve approximation. The target is to represent freehand sketches in a compact format, achieving high compression rate within a certain error tolerance and with lower computation to be practically adaptable solution for the real applications.

The remainder of this paper is organized as follows: The main idea of our proposed strategy is outlined in Section 2. In Section 3, the method of stroke fragmentation for generation of user intended breakpoints is given. In Section 4, we will discuss sketch parameterization by Recursive Bezier Curve Approximation in detail. Section 5 will present our experiments. Conclusions are given in the final Section.

2 The Proposed Strategy of Sketch Parameterization

The Bezier curve representation has been widely used since its coefficients can be easily obtained and its shape can be easily manipulated. A Bezier curve is defined using two anchor points, on-curve control points and at least one shape point, off-curve control point. The on-curve control points are the two end points of the curve actually located on the path of the curve, while the other off-curve control points define the gradient from the two end points, which are usually not located on the curve path. The off-curve points control the shape of the curve. The curve is actually a blend of the off-curve control points. The more off-curve control points a Bezier curve has, the more complicated shape can be represented; however the order of the mathematical curve equation becomes higher. The approximation of hand-drawing strokes using high order Bezier curve reduces the number of on-curve points and produces more compact data representation. However, the approximation of high order curve equation usually requires large amount of computation.

In order to easily manipulate a sketch, Raymaekers et al [11] have proposed a method to group the individual pixels of sketch into segments represented by a cubic Bezier curve. Whenever the least square error of the curve fitting passes a preset threshold, a new segment is created and two off-curve points of this segment can be constructed by means of least square minimization. For the computational efficiency and data compression of sketch, Park and Kwon [12] have recently presented a method of sketch approximation by piecewise fitting of a series of quadratic Bezier curves using least square error approximation. The common ground between the above two methods is that the curve control points are produced dependent only upon the local geometric curvature so that the difference between the curve and the corresponding data points is as small as possible. However, it is not a reliable way to deliver the users' intentions that the curve control points are determined alone by curvature information. It is also time consuming to fit the undetermined segmented points using least square error approximation.

We propose a method to parameterize sketches using a cubic Bezier curve, where the complicated shaped strokes are represented by piecewise approximated cubic Bezier curves, because the Bezier curves can only represent simple arc shape curves. **Fig. 1** shows the processing diagram of our proposed approximation method. This process has two independent sub-processing modules, stroke fragmentation and curve

approximation loop. The stroke fragmentation is performed only once per given input set of strokes to generate a series of the user intended breakpoints at each stroke, where the speed of pen movement reaches the minimal value and where the curvatures change sharply. These candidate breakpoints are selected as the initial on-curve control points in curve approximation loop. The curve approximation routine, including curve control points (including on-curve and off-curve control points) selection, Bezier curves approximation and fitting error evaluation, operates recursively till the piecewise fitting error is satisfactory. The recursive curve approximation loop has two nested sub-processes. The inner is the optimization of off-curve control points, which optimizes the piecewise curve approximation to minimize the fitting error. The outer is the adjustment of on-curve control points, which generates some new on-curve control points by bisecting each of current curves to increase the pieces of segments and reduce the fitting error. The inputting graphic object, drawn by a set of strokes, can finally be represented as a set of parameters of some pieces of cubic Bezier curve. The risk generating excessive breakpoints (on-curve points) on the stroke because of using series of cubic Bezier curves (compared to approximated by high order Bezier curves) can somewhat avoided by optimization of off-curve control points.

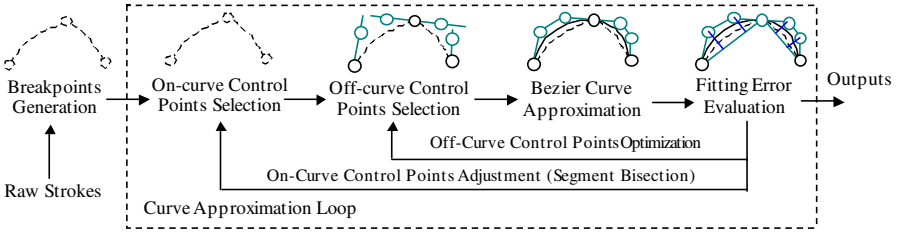


Fig. 1. The Diagram of sketch parameterization using recursive curve approximation

3 User Intended Breakpoints Generation

Stroke fragmentation is a very basic problem, making it widely applicable to intelligent ink manipulation as well as other higher-level digital ink analyses. The goal of stroke fragmentation is to fragment a wide variety of sketched symbols into simpler structures so that they could faithfully represent the original form with a less complex form. The key challenge of stroke fragmentation is to find out which bumps and bends (breakpoints) are intended and which are accident. Most of existing methods do stroke fragmentation based on curvature only. Usually, curvature information alone is not reliable enough to determine such points. Instead, the speed based methods have proven to be a much more reliable measure to determine the intended breakpoints for the observation that the speed of the pen tip significantly reduces at the intended corner points [5][13]. For sketch parameterization, the purpose of stroke fragmentation is to extract sharp turning pen movement points and bending points at each stroke quickly by the information of pen speed and curvature, which are located as initial on-curve control points of Bezier curves.

3.1 Candidate Breakpoints Selection Based on Pen Speed

It has been discovered that it is natural to slow the pen when making many kinds of intentional discontinuities in the shape [5][13], for instance the corners formed by two lines. Similarly, when drawing a rectangle with a single pen stroke, users would likely slow down at the corners, which are supposed to be the segment points. **Fig. 2.** shows the speed profile for a typical square in our experiments. The corners can be easily identified by the low pen speed.

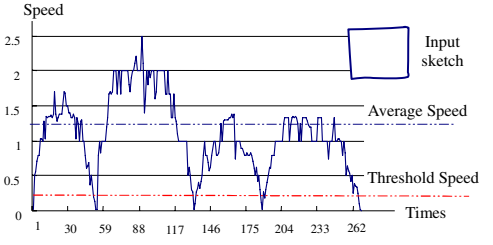


Fig. 2. 1 Illustration of speed profile for a square

Given a digital stroke $\boldsymbol{S}$, which contains N numbers of time-ordered points $\{\boldsymbol{P_1}, \boldsymbol{P_2},\ldots, \boldsymbol{P_N}\}$. There is also a corresponding speed sequence: $\{\boldsymbol{SP_1}, \boldsymbol{SP_2},\ldots, \boldsymbol{SP_N}\}$, where $\boldsymbol{SP_i}$ represents the pen speed at the point $\boldsymbol{P_i}$. The point $\boldsymbol{P_i}$ would then be selected as a candidate breakpoint when the following conditions are satisfied:

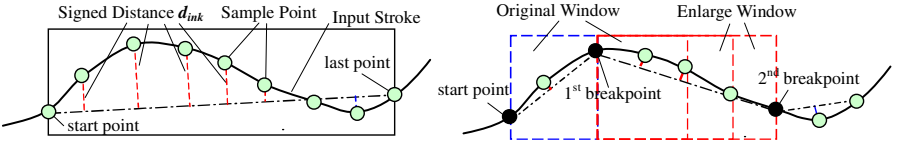
$$\begin{cases} SP_i < \overline{SP} * a \\ \forall j \;\; SP_i < SP_j, |\, j - i \,| < N * b \end{cases} \tag{1}$$

where, $\overline{SP}$ is an average pen speed of the whole stroke, a and b are two thresholds which will affect the spacing of the candidate breakpoints dependent on the length of a stroke and must be set by means of some statistical experiments.

3.2 Candidate Breakpoints Generation Based on Curvature

In some case, some users' intended breakpoints cannot be located with pen speed because users probably draw it smoothly without the variation of pen speed. The information of curvature must be used to capture such points. We develop an approximate method to measure the variation of curvature of a segment or a stroke and use greedy method based on sliding window algorithm (SWA) to pick up the candidate breakpoints. Given a window with fixed sizes that envelops a lot of successive ink points of a stroke, the distances $\boldsymbol{d_{ink}}$ from each sample point to the line connecting the first point and last point in the window can be calculated and signed according to their relative positions along with the drawing direction, that is, the distance located in left side is positive and in right side is negative, as shown in **Fig. 3**(a). The sum of distances of all ink points between these two points can be seen as a measurement of the curvature of a segment affected by two end points.

Accordingly, starting with the first sample point of a stroke, we set a sliding window initially with one unit width that envelops at least two successive ink points of the stroke, the measurement of the curvature of segment in the window are calculated, as shown in **Fig. 3**(b). The width of window broadens step by step alone with drawing direction of stroke and the measurement of the curvature of segment in the window are incrementally calculated until the measurement exceeds the experimental threshold or all ink points of the stroke are tested. Whenever a measurement exceeds the experimental threshold, the last point in that window is chosen to be a candidate breakpoint and as a new start point of next repetition. Then, the window with original size moves to the position of that point, the same process described above is repeated until all successive points in a window are examined, as shown in **Fig. 3**(b). It can minimize the chance of over-splitting caused by jitter of pen movements. The window size are dependent on the density of ink points for a stroke and can be given as a function of the perimeter and bounding box of each stroke. The step of window enlargement and the threshold of measurement of curvature must be experimental defined by making a tradeoff between precision and efficiency. In our experiments, two parameters set as 15 (pixels) and 150 (pixels) respectively.



(a) Measurement of the curvature of a segment (b) Greedy method using sliding window

Fig. 3. Candidate Breakpoints Generation Using Curvature

After the processing described above, two portions of candidate breakpoints can be got besides two end points of a stroke, which are either the ink points where the pen speed reaches the local minimum or the ones where the measurement of curvature is local maximum. However, some of them are redundancy, and the candidates mergence is required to remove the superfluous or adjacent points. Ultimately, any two adjacent breakpoints separate the segment between them as a sub-stroke and will be used as the initial on-curve control points of a cubic Bezier curve.

4 Sketch Parameterization Using Piecewise Cubic Bezier Curve

4.1 Principle of Sketch Approximating by Cubic Bezier Curve

For a fragmentized stroke (or a segment of the stroke) with some breakpoints (can be two end points only) to be eligible for the further processing, a cubic Bezier curve approximation method is applied to find out the curve control points. A cubic Bezier curve with two on-curve control points and two off-curve points are estimated which satisfy following conditions as shown in **Fig. 4**.

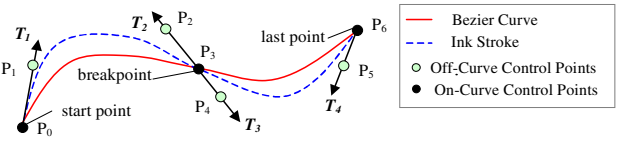


Fig. 4. Sketch approximating by cubic Bezier curve

(1). The two on-curve control points of the estimated curve should be breakpoints (including first and last ink points) of the stroke; such as, $\boldsymbol{P_0}$ and $\boldsymbol{P_3}$, $\boldsymbol{P_3}$ and $\boldsymbol{P_6}$, shown in **Fig. 4**.

(2). The two off-curve control points of the estimated curve should be located on the corresponding tangents at two end (on-curve control) points of the segment respectively; for example, $\boldsymbol{P_1}$.is located on the tangent $\boldsymbol{T_1}$ at $\boldsymbol{P_0}$, $\boldsymbol{P_2}$.is on the tangent $\boldsymbol{T_2}$ at $\boldsymbol{P_3}$, and so on, as shown in **Fig. 4**.

(3). For any two adjacent segments that share a common breakpoint such as $\boldsymbol{P_3}$ in **Fig. 4**, their tangents at that point should be collinear in order to make the curves connect smoothly, that is, $\boldsymbol{T_2}$ and $\boldsymbol{T_3}$ are collinear vectors in **Fig. 4**. In other words, the first off-curve control point in sequential segment should be collinear with two last curve control points in previous segment; for example, $\boldsymbol{P_2}$, $\boldsymbol{P_3}$ and $\boldsymbol{P_4}$ should be collinear.

(4). The fitting error between actual stroke and the approximated curve must be within an acceptable range by optimization of off-curve control points. If the fitting error exceeds a certain threshold, the on-curve control points should be adjusted. The recursive curve fitting operation is repeated for each of the newly generated segment till the error falls within tolerance.

For a given segment (or a whole stroke) with two end points $\boldsymbol{P_0}$ and $\boldsymbol{P_3}$, its approximated cubic Bezier curve with two on-curve control points $\boldsymbol{P_0}$ and $\boldsymbol{P_3}$ and two off-curve control points $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ can be represented as follow, as shown in **Fig. 4**.

$$P(u)=\sum_{k=0}^{3} P_k BEZ_{k,3}(u),\ 0\le u\le 1 \tag{2}$$

where, $BEZ_{k,3}(u)=C(3,k)u^k(1-u)^{3-k}$ is a cubic Bernstein polynomial function, and two off-curve control points $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ are defined based on the location and tangent of two on-curve control points (two end points of a segment or a stroke) $\boldsymbol{P_0}$ and $\boldsymbol{P_3}$, as follow respectively:

$$\begin{cases} P_1 = c_1 * T_1 + P_0 \\ P_2 = c_2 * T_2 + P_3 \end{cases} \tag{3}$$

where, $\boldsymbol{T_1}$ and $\boldsymbol{T_2}$ are two unit tangent vectors of two end points of a segment respectively, which can be approximated by calculating the delta difference of the end points to its adjacent ink points denoted as ΔP_0 and ΔP_3 respectively; c_1 and c_2 are two variable coefficients to locate the off-curve control points on the tangent of end points of the segment, usually we can set c_1=c_2=c, which is proportionate to the length of segment for uniform fitting of the segment.

Therefore, an approximated cubic Bezier curve can be represented as:

$$P(u) = [2(P_0 + P_3) + c(\Delta P_0 + \Delta P_3)]BEZ_{k,3}(u),\ 0 \le u \le 1 \tag{4}$$

4.2 Fitting Error Evaluation

To inspect the suitability of the sketch parameterization, the fitting error between actual ink points and corresponding points of the approximated curve must be evaluated. We evaluate the fitting error by two steps. Firstly, several pairs of points on the original ink segment and corresponding approximated Bezier curve are selected using fixed size sliding window, and all Euripidean distances from them to their opposite chord connecting two end points of a segment are calculated one by one as shown in **Fig. 5**, including a set of d_{ink} for each selected ink point and a set of d_{curve} for each selected curve point (for illustrating clearly, some scenes in Fig.5 such as space between sliding windows and lines in **Fig. 5** are artificial). Then, the distance difference (d_{curve}-d_{ink}) for each pair is calculated, and the fitting error is defined as the ratio of the maximal distance difference $(d_{curve}\text{-}d_{ink})_{max}$ to the length of chord L, which reflects approximately the maximal difference of the local curvature distribution between the original segment and the approximated Bezier curve at a certain extent as described in section 3.2.

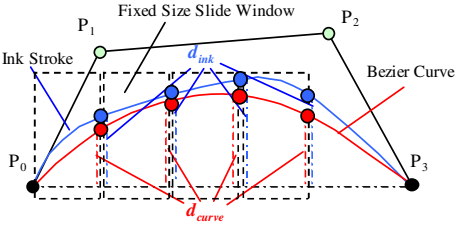


Fig. 5. The distance calculation using fixed size slide window for fitting error evaluation

Accordingly, the fitting error can then be represented as:

$$e_f = \max\left[e_f^1, e_f^2, \cdots, e_f^i, \cdots, , e_f^m\right] \text{ Where } e_f^i = \left|\frac{\left(d_{curve}^i - d_{ink}^i\right)}{L}\right| \tag{5}$$

where, m is the numbers of selected on-curve points. The value of m must not be less than three, and the points with the maximal local curvature on the original ink path should be selected as possible. Meanwhile, The selected points from the original ink path and the approximated Bezier curve should be corresponding. By defining a local coordinate system, the distance difference (d_{curve}-d_{ink}) for each pair of selected points, as shown in **Fig. 5**, can be easily calculated as follows:

$$\left|d_{curve}^i - d_{ink}^i\right| = \left|[2(y_0 + y_3) + c(\nabla y_0 + \nabla y_3)]BEZ_{k,3}(u_i) - y_{ink}^i\right| \tag{6}$$

4.3 Sketch Parameterization

As described in section 4.1, every stroke of a sketchy object can be parameterized by piecewise cubic Bezier curve within an acceptable fitting error tolerance with a set of parameters as follows:

$$A\ stroke \propto \left\{\left(Pm_i \mid i = 0,1,2,\cdots,k\right),\left(c_i \mid i = 1,2,\cdots,k\right),e_f\right\}, e_f \le e_{\exp} \tag{7}$$

where, $\{Pm_i\}$ is a series of on-curve control points, Pm_0 and Pm_k are two end points of a stroke respectively, $\{Pm_i|i=1,2,\ldots,k-1\}$ is a sequence of middle on-curve control points that fragmentize a stroke into $\boldsymbol{k}$ numbers of segments (initially, there are breakpoints generated by stroke fragmentation), $\{\boldsymbol{c}_i\}$ is a set of proportional coefficients for locating the off-curve control points of the approximated cubic Bezier curve for each of segments of a stoke, e_f and e_{exp} are the actual fitting error and the experimental threshold of the fitting error respectively.

Our key ideas of sketch parameterization are (i) to optimise off-curve control points of the piecewise curve approximation for each segments of a stroke to minimize the fitting error and (ii) minimize the numbers of new added breakpoints for each segment as possible to preserve the users' intention. That is, for each segment, the optimization of two off-line curve control points is the priority. The new ink breakpoint would be generated and inserted to bisect the segment into two sub-segments only if the fitting error of optimization of off-line curve control points great than the experimental threshold. Accordingly, for an inputting stroke, the task of sketch parameterization using piecewise cubic Bezier curve can be seen as a process to search out some parameters of curve approximation, including a series of stroke breakpoints and a set of the proportional coefficients, to fit as closely as possible to each of strokes. In fact, this process is the recursive piecewise cubic Bezier curve approximation within an acceptable approximated error, as shown in **Fig. 1**.

As described in equation (5) and (6), for every segment of a stroke between each pair of on-curve control points $\{Pm_{i-1},Pm_i|i=1,2,\ldots,k\}$, the fitting error is variable with this pair of points and a proportional coefficient $\{c_i|i=1,2,\ldots,k\}$ corresponding to the selection of off-curve control points, that is:

$$e_f^i \propto f\left(\left\{Pm_{i-1},Pm_i\right\},c_i\right), i = 1,2,\cdots,k \tag{8}$$

Based on the generation of the users' intended candidate breakpoints, the proportional coefficients $\{c_i\}$ for every segment of a stroke between each pair of on-curve control points can then be gained from solving the following optimization problem:

$$\begin{cases} Minimize: f(c_i) = e_f^i = \max\left[\left|\frac{\left(d_{curve}^1 - d_{ink}^1\right)}{L}\right|,\left|\frac{\left(d_{curve}^2 - d_{ink}^2\right)}{L}\right|,\cdots,\left|\frac{\left(d_{curve}^m - d_{ink}^m\right)}{L}\right|\right] \\ Subject\ to\ constrains: 2\left(Pm_{i-1} + Pm_i\right) + c\left(\nabla Pm_{i-1} + \nabla Pm_i\right) = 0, c \ge 0 \end{cases} \tag{9}$$

Equation (9) can be solved easily by traditional optimal approach such as Newton's method. If the value of the fitting error for an optimal solution of equation (9) is greater than the experimental threshold, a new on-curve control points must be generated between the current pair of on-curve control points by bisecting the current segment into two newly sub-segments. The process of solving the optimization

problem for each of sub-segments described in equation (9) would then be repeated till the desired fitting error is achieved. A stroke is parameterized after all its segments are approximated by piecewise cubic Bezier curve suitably.

As a result, a sketchy object with *n* number of inputting strokes can be represented as a set of parameters of some pieces of cubic Bezier curve as follow:

$$Sketch \propto \left\{ \left(Pm_i^j \mid i = 0,1,2,\cdots,k^j \right), \left(c_i^j \mid i = 1,2,\cdots,k^j \right), e_f^j \right\}, \; j = 1,2,\cdots,n \tag{10}$$

4.4 Curve to Line Approximation

For some applications such as sketch recognition and display of sketch, the curve components should be approximated as a line component instead of curve representation within the allowable error range. Since the line approximation from raw digital ink data is a computationally intensive process, the Bezier curve fitting parameters can be used directly to examine the possibility of a line approximation.

One of the simplest methods of line approximation is to replace the approximated cubic Bezier curve with its three boundaries of the control polygon, which connects orderly all control points. The process is the recursive dichotomy (u=1/2 in equation of Bezier curve) of a cubic Bezier curve, and the control points of each segment can be obtained by de Casteljau algorithm till all control points are located within the error tolerance boundary, as shown in the **Fig. 6**. If two conditions are satisfied, i) the Euclidean distances between the on and off control points of each segment are smaller than allowable delta tolerance, and ii) the minimum Euclidean distance between the off-line point and the straight line between the two end (on-curve) control points is smaller than the error tolerance. Then, the off-line control point is assumed to be within the error tolerance. If the test is successful, a piece of Bezier Curve can be represented as three straight lines by connecting orderly each of control points.

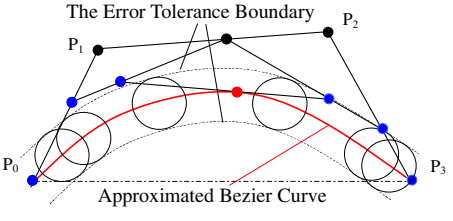


Fig. 6. Illustration of curve to line approximation

5 Experiments and Discussion

In the experiment, two different types of freehand drawings are designed: graphic drawings and Chinese characters. Some of our experimental results are listed in **Table 1**. The first column is the original inputting sketch, and the second is the fitted sketch where all the segments have been modeled curves and some of the segments between every two breakpoints are subdivided because the error is higher than the

threshold. The number of strokes, candidate breakpoints and fragmented segments for each sketch are listed in third, fourth and fifth column respectively. The data listed in sixth column is the maximal among all of the fitting errors to fit for every segment using piecewise Bezier curve in a sketch. The compression ratio is the ratio of file storage of the parameterized sketch to the original ink points. All experiments are done on an Intel PC with a 2.8 GHz CPU and 512MB memory running on Microsoft Windows XP Professional.

Table 1. Some instances of our experiments for sketch parameterization

Input Sketch	Fitted Sketch	Number of Strokes	Pieces of Segments	Max. Fitting Error	Compression Ratio	Computing Cost (ms)
		2	4	0.0802	0.160	0.045
		1	3	0.0510	0.108	0.078
		2	4	0.0756	0.165	0.153
		8	10	0.0714	0.173	0.176
		14	23	0.0975	0.206	0.369
		5	5	0.0478	0.129	0.107
		8	9	0.0783	0.179	0.201
		15	22	0.0856	0.197	0.485

From **Table 1**, we can see that our method of sketch parameterization is both effective and efficient. Firstly, the approximating time for all inputting sketches is less than 0.5 milliseconds. This indicates that our algorithm can provide the efficient-computation for a wide range of freehand drawing graphic objects, and is suitable for online freehand sketches inputting. Secondly, all drawing objects or handwritings in Chinese characters can be parameterized by some piecewise Bezier curves within allowed fitting error (in our experiments, the allowed fitting error is set to 0.1). These connote that sketch parameterization using a piecewise cubic Bezier curve approximation is adaptable for a wide range of freehand drawing graphic objects. It is very useful for many applications where the original of ink path must be hold and transmitted or re-used, such as conceptual expression for product or software design, computer supported cooperative work, message exchange in mobile computing, pervasive computing and ambient intelligence, electric notes/white-board for digital classroom/office, and so on. Thirdly, all freehand drawings can be approximated with more than fifteen percent of data compression. This means that it can reduce the redundancy of on-line freehand drawing data within expected fitting tolerance for graphic data exchange and transmission. This would be very useful especially in low bandwidth wireless networks.

Fig. 7 illustrates the visual approximated effects of our two typical cases for sketch parameterization.

(a) Approximation of drawn symbols (b) Approximation of Chinese signature

Fig. 7. Examples of our experiments for sketch parameterization

6 Conclusions

In this paper, we present a practical solution for efficient representation of hand drawing graphics. The goals of the proposed method are to parameterize and reduce the redundancy of online hand drawing graphic data. A piecewise Bezier curve approximation is implemented in recursive architecture. For the computational efficiency, the cubic Bezier curve representation is only used in our method since it is relatively simple and the curve coefficients can be easily obtained. But the cubic Bezier curves can only represent simple arc shape curves. Thus, complicate shaped strokes are represented by piecewise approximated cubic Bezier curves in our approach. The advantages of our method can be concluded as follows. Firstly, it is suit for the retainable freehand drawing of a wide range of graphic objects without loss of user originals. Secondly, it can reduce the redundancy of on-line freehand drawing data and provide the efficient-computation for graphic data exchange and transmission. Thirdly, it is user-independent because it is dependent only on the allowed fitting error without loss of user intention. The experimental results show both effective and efficient.

Theoretically, sketch parameterization must be done with the minimization of fitting error, the maximization of computing efficiency and the optimization of data compression. However, there is a conflict in sketch parameterization between the fitting tolerance and the computing complexity as well as the data compression ratio. That is to say, the approximated curve can fit better to original ink path by generating more segments for each of strokes in a sketch, however, the run time would be higher and the ratio of data compression would be lower. Therefore, a compromise must be made for sketch parameterization between the minimization of fitting error, the maximization of computing efficiency and the optimization of data compression dependent on the requirements of particular applications. For example, the accurate fitting may be preferential for cooperative carton design over Internet, the data compression must be priority for message exchange and transmission using carton over wireless network, the computing complexity would be the most important in drawing graphic objects with portable devices. Efficient high order curve approximation and optimization idea may be highly desired as a solution in the near future.

Acknowledgement

This paper is supported by the grants from the National Natural Science Foundation of China [Project No. **69903006** and **60373065**] and the Program for New Century Excellent Talents in University of China [Project No. **NCET-04-04605**].

References

1. Zhengxing Sun and Jing Liu, Informal user interfaces for graphical computing, Lecture Notes in Computer Science, Vol. 3784, 2005, pp. 675-682.
2. Rubine Dean, Specifying gestures by example, Computer Graphics, 1991, Vol. 25, pp. 329-337.
3. Fonseca M. J., Pimentel C. and Jorge J. A., CALI-an online scribble recognizer for calligraphic interfaces, In: AAAI Spring Symposium on Sketch Understanding, AAAI Press, 2002, pp. 51-58.
4. Xiaogang Xu, Zhengxing Sun, Binbin Peng, et al, An online composite graphics recognition approach based on matching of spatial relation graphs, International Journal of Document Analysis and Recognition, Vol. 7, No.1, 2004, pp. 44-55.
5. Calhoun C., Stahovich T.F., Kurtoglu, T. et al: Recognizing Multi-Stroke Symbols. In: AAAI Spring Symposium on Sketch Understanding, AAAI Press, 2002, pp. 15-23.
6. Zhengxing Sun, Wenyin Liu, Binbin Peng, et al, User Adaptation for Online Sketchy Shape Recognition, Lecture Notes in Computer Science, Vol. 3088, 2004, pp. 303-314.
7. Sezgin T. M. and Davis R., HMM-Based Efficient Sketch Recognition, Proceedings of the 10th international conference on IUI, Jan., 2005, San Diego, California, USA.
8. Zhengxing Sun, Wei Jiang and Jianyong Sun, Adaptive online multi-stroke sketch recognition based on Hidden Markov model, Lecture Notes in Artificial Intelligences, Springer-Veralg, 2006, to be published.
9. Wenyin Liu, Xiangyu Jin and Zhengxing Sun, Sketch-Based User Interface for Inputting Graphic Objects on Small Screen Device, Lecture Notes in Computer Science, Vol. 2390, 2002, pp. 67-85.
10. Huang Z and Cohen F, Affine-invariant b-spline moments for curve matching. IEEE Transactions on Image Processing. Vol. 5, No.10, 1996, pp. 1473-1480.
11. Raymaekers C., Vansichem, G., and Reeth F. V., Improving sketching by utilizing haptic feedback, In AAAI Spring Symposium on Sketch Understanding, AAAI Press, 2002, pp. 113-117.
12. Park Jaehwa and Kwon Young-Bin, An Efficient Representation of Hand Sketch Graphic Messages Using Recursive Bezier Curve Approximation, Lecture Notes in Computer Science, Vol. 3211, 2004, pp. 392-399.
13. Sezgin T. M., Stahovich T. and Davis R., Sketch based interfaces: early processing for sketch understanding. Proceedings of the 2001 Workshop on PUI, Orlando, Florida, 2001, pp. 1-8.

Biometric Recognition Based on Line Shape Descriptors*

Anton Cervantes[1], Gemma Sánchez[1], Josep Lladós[1], Agnès Borràs[1], and Ana Rodríguez[2]

[1] Centre de Visió per Computador i Departament de Ciències de la Computació
Universitat Autònoma de Barcelona, Edifici O, Campus UAB
08193, Bellaterra, Catalonia, Spain
{anton, gemma, josep, agnesba}@cvc.uab.es
[2] Hospital Universitari Arnau de Vilanova
Lleida, Catalonia, Spain
arodriguez@arnau.scs.es

Abstract. In this paper we propose biometric descriptors inspired by shape signatures traditionally used in graphics recognition approaches. In particular several methods based on line shape descriptors used to identify newborns from the biometric information of the ears are developed. The process steps are the following: image acquisition, ear segmentation, ear normalization, feature extraction and identification. Several shape signatures are defined from contour images. These are formulated in terms of zoning and contour crossings descriptors. Experimental results are presented to demonstrate the effectiveness of the used techniques.

1 Introduction

Biometric technology is based on identifying one individual from another by measuring some unique features like face, iris, voice, DNA, fingerprint or ear shape. A number of contributions exist in the literature presenting mature solutions using the above biometric descriptors. The reader is referred to [1] for a good introduction to biometric recognition. Although being apparently different research fields, biometrics and graphics recognition are in some cases close areas, at least from the methodological point of view. Some biometric descriptors, in particular fingerprints consist of line structures spatially arranged. A fingerprint structure encoded, and matched, in terms of the geometry and topology of ridges and minutiae is somehow equivalent to a line drawing consisting of lines and junctions. Probably due to these "close" representation, some authors have experimented with similar techniques in document analysis and biometrics. For example, Bunke and his team have applied graph matching techniques, often used in symbol recognition, in fingerprint classification [2]. Govindaraju et al. [3] used chaincodes, a typical representation in line drawings, for fingerprint matching. In addition to fingerprints, a biometric descriptor traditionally

* This work has been partially supported by the Spanish project CICYT TIC 2003-09291.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 346–357, 2006.
© Springer-Verlag Berlin Heidelberg 2006

familiar among the Document Analysis community, other biometric descriptors can also be formulated from a "graphics recognition" perspective. Actually it is necessary to represent the biometric descriptor using geometric and structural properties of basic features as lines (contour approximation) or characteristic points. The recognition is therefore formulated in terms of shape similarity. Other examples are the shape of vessels [4] or the ear shape [5], [6]. Ear shape is another non-intrusive biometric descriptor that can also be formulated in terms of line-to-line matching. In particular, in this paper we propose the ear shape for the identification of newborns.

The use of biometric approaches for newborn presents several drawbacks. DNA is invasive so it can not be used each time the baby is changed of room. Iris pattern and retina are also invasive because the first days newborns have their eyes closed so taking images is very difficult. Foot geometry is not characteristic enough in the first days of life, and hand geometry is difficult to acquire because newborns usually have their hands closed and keeping all the fingers in the correct position is not easy. In our research we have tested two different approaches: fingerprints and ear shapes. Using fingerprints we could check that although fingerprints are fully formed at about seven months [7] of fetus development, the first days of life fingerprints seem not to be mature enough to be acquired properly. In the case of ear shapes, the two main reasons to choose this biometric descriptor are because it has enough recognition ratio for our purpose and because while other techniques historically associated to newborn identification (for example footprints) are not 100% passive, ear acquisition does not require any kind of cooperation. The results obtained working with fingerprints and ear shapes are in [8].

Ear shape is not as discriminant as other features but in some frameworks could be more suitable. In the literature there exist some approaches about the ear features extraction. In [5] a set of circles are created and centered in the centroid of the contours of the ear, and a count of the number of the intersection points for each radius and all the distances between neighboring points are used in the recognition process. In [6] a geometrical vector containing normalized distances between characteristic points of the inner ear is used, a vector describing the outer ear contour is also used to compare ears. In [9] each ear is modelled as an adjacency graph built from the Voronoi diagram of its curve segments and a graph matching process is performed. In [10] a linear transform that transforms an ear image into a smooth dome shaped surface whose special shape facilitates a new form of feature extraction that extracts the essential ear signature without the need for explicit ear extraction is developed. Our work proposes a set of shape signatures designed to describe ear shapes and to be used in a biometric newborn identification framework. Some interesting reviews on general shape recognition have been looked up and some of the descriptors in this paper proposed are inspired on techniques explained in [11], [12] and [13].

The general organization of our approach consists in the following steps: image acquisition, ear segmentation, ear normalization, feature extraction and identification process. In the ear segmentation step a preprocessing to enhance ear

Fig. 1. Grey-level images

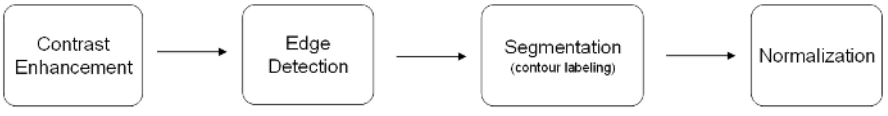


Fig. 2. A flowchart of the proposed algorithm

edges and a processing of the *Canny* edges using some typical characteristics of the ear to get the chain around the ear is performed. In the ear normalization step an affine transformation to standardize ears before the feature extraction process is performed. In the feature extraction process the distribution of some characteristic points is studied using zoning and contour-crossings based descriptors. Finally, in the identification step an algorithm to reach the most similar class in the database using different signatures is explained.

The organization of this paper is as follows: in section 2 the image acquisition, ear segmentation and the normalization process algorithms are developed, in section 3 the descriptors used to describe the ear structure are explained. In section 4 the recognition algorithm to reach the final decision is described, in section 5 the experimental results obtained with our own database are presented, finally in section 6 conclusions and future work are explained.

2 Ear Segmentation and Normalization

Ear images have been acquired with a high resolution digital camera. The model of the camera is a Nikon Coolpix 4300 and has a resolution of 4 megapixels. The images acquired have been cropped (512x512) to obtain images where only the ear appears, this way the following step, the ear segmentation process, becomes easier. Illumination conditions were controlled trying to avoid brightness points. Some examples of grey level images are shown in Fig 1.

After acquiring the image four different steps are performed before the feature extraction could be done, a flowchart of the proposed algorithm is shown in Fig 2. In the first one a preprocessing to improve the outer ear contour location is performed. In the second one the contours of the image are obtained using an edge detector. In the third step the edge segmentation is developed after

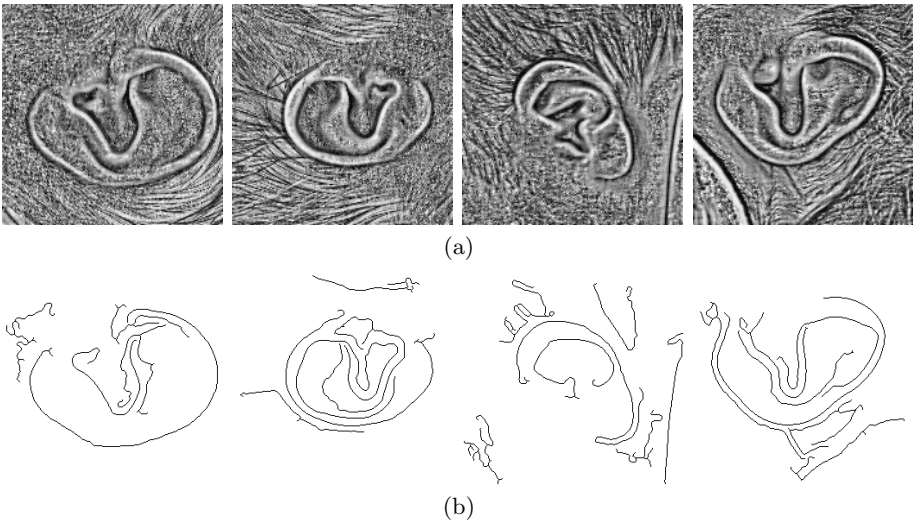


Fig. 3. (a) Local maximization of the contrast of the grey level images (b) Edges found by the *Canny* detector applied over the the images of a)

labelling and processing the pixels obtained in the previous step. Finally the ear normalization is computed. Let us further describe each step of the process:

The aim of the first part of the process consist of a little preprocessing of the grey level image to enhance the outer ear contour. For this purpose a local maximization of the contrast is applied before the edge extraction to obtain a image where the outer ear contour becomes more contrasted, see Fig 3 (a). Once this enhancement is performed getting the edge around the ear becomes easier. The window used in the enhancement process is square because at this moment of the process the position and orientation of the ear in the image is still unknown. The size of the window is empirically set.

In the second step of the process, after the local maximization of the contrast is computed the edge map is obtained using the *Canny* operator in the preprocessed image. The sigma value of the *Canny* detector should be great enough to remove spurious artifacts but small enough to avoid unnecessary smoothing to preserve the original shape and getting the correct location of the ear. In spite of the smoothing carried out by the edge detector many spurious lines as freckles, possible changes in color of the skin, hair and so on also appear in the line structure. Visualizing the edges obtained is easy to see that the line around the ear is one of the longest ones. Therefore a cleaning process removing lines whose length is lower than a fixed threshold is performed. After this process an image with the contours of the ear and spurious lines that have survived to the removing process is obtained, see Fig 3 (b).

In the third step of the process, the ear segmentation is performed after a pixel labelling carried out depending on some characteristics as number of neighbors and level and sign of curvature computed along the edges. In the labelling process

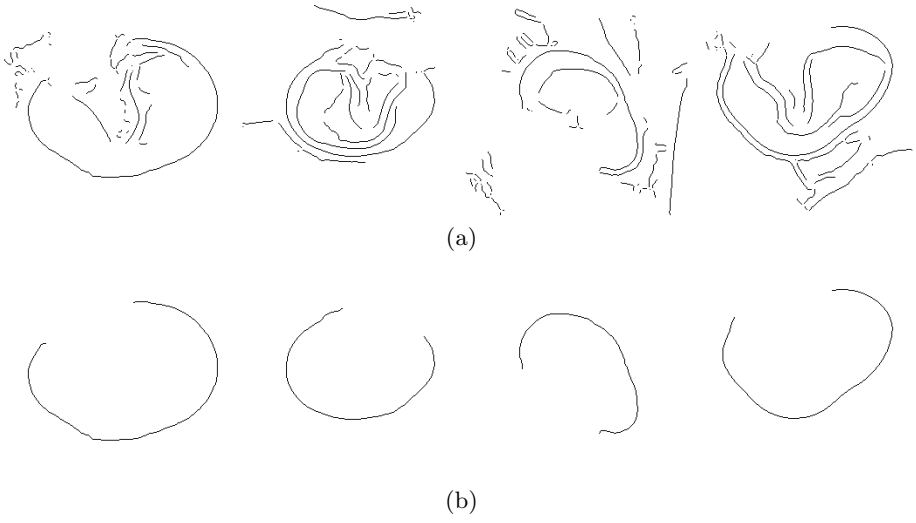


Fig. 4. (a) *Canny* image without the pixels whose curvature is higher in magnitude than a threshold. (b) Longest line (segmentation).

pixels are grouped into three different classes. The first class of pixels is named P_b and contain pixels that have more than two neighbors (bifurcation pixels), the second class is P_{hc} and the curvature level (magnitude) of these pixels is higher than a predefined threshold, *thr*. The rest of the pixels are the pixels whose curvature is lower than thr, and are assigned to class P_{lc}. This threshold is image dependent, so that the contours obtained with the *Canny* operator depends on how far the image has been taken. The pixel classes have been selected this way because in general, the outer ear contour does not contain pixels from P_b class nor from P_{hc}. For these reasons the algorithm steps to find the correct edge is as follows. First of all a removing step of the pixels of P_b class is performed. Next, the curvature along the edges remaining in the image is computed, and the pixels belonging to P_{hc} are also removed, see Fig 4 (a). After performing these two previous stages, only smooth shapes remain in the image, in Fig 5 (d) P_b, P_{hc} appear in black and P_{lc} are painted in grey. Finally, two images are extracted from the surviving pixels. The first one contains pixels whose curvature runs on $I_1 = [-thr, 1]$ and the second one contains the pixels whose curvature runs on $I_2 = [-1, thr]$. The first image contains pixels whose curvature is negative or with a positive value but with a magnitude under 1. In the second image it happens the contrary. This separation has been inspired by the morphology of the ear. The outer ear contour is similar to an ellipsoidal shape, therefore the sign of the curvature should remain unchanged almost everywhere. The outer ear contour is then assigned to the longest chain among all the edges of the two images, see Fig 4 (b). Two examples illustrating the segmentation reached whether the high curvature pixels are not removed or whether the pixels are not classified by the sign of their curvature are shown in Fig 5 and 6.

Fig. 5. (a) *Canny* edges (b) Longest edge before removing pixels of high curvature and pixels with more than two neighbors (c) Longest edge obtained using the information displayed in d), i.e., after removing pixels of high curvature and pixels with more than two neighbors (d) Black: bifurcation and high curvature pixels. Gray: Pixels with the magnitude of the curvature under a fixed threshold.

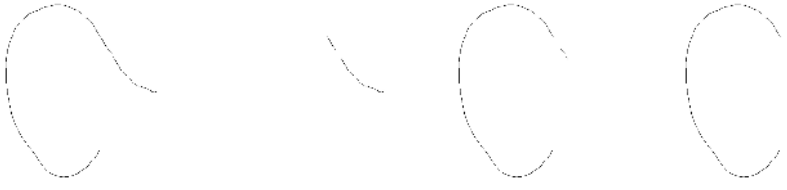


Fig. 6. (a) Segmentation without using the sign of the curvature (b) Pixels where the curvature is in I_1 (c) Pixels where the curvature is in I_2 (d) Longest edge among the edges from b) and c) (better approximation of the ear)

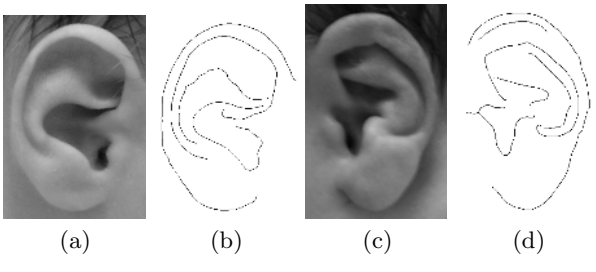


Fig. 7. (a) (c) Ear normalizations (b) (d) *Canny* map, the edges 'outside' the outer ear contour have been removed automatically

Finally, the normalization process is performed. This step consist of three different parts: a cropping, a rotation and a scaling. Cropping, to eliminate everything out of the ROI, rotation, to get the ear vertical, scaling to obtain similar sizes for all the images of the database. Some normalized images are shown in Fig 7 (a) and (c) and the *Canny* operator computed over the normalized images without short lines and lines out of the outer ear contour are in (b) and (d).

The curvature along all the lines of the image has been computed in each pixel as the difference between the mean orientation of the pixels before and after the current pixel divided by the size of the neighborhood used. The size of the neighborhood depends on what kind of points should be detected. Small neighborhoods must be used to find sudden changes of direction while greater neighborhoods must be used when smoother changes must be found. In our case a neighborhood of 10 pixels has been used.

3 Feature Extraction

The aim of this process consists in extracting some features to compare different ears in the recognition process. The features are extracted from the *Canny* edges computed over the normalized images. To obtain normalized features the major and minor axes are computed following the approach of [6]. These are used during the classification step as reference data to align the input image to the model. The major axe, Ay, is set as the segment joining the two furthest points of the outer ear, the center, O, is defined as the midpoint of Ay. The second axe, Ax, is the orthogonal to Ay by O, see Fig 8(a).

In this paper four different shape descriptors are used to characterize the ear contours:

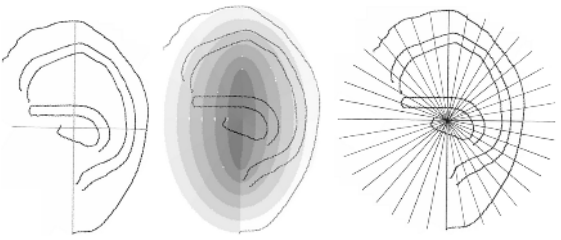
- Z_d density zoning features.
- Z_α angle zoning features.
- C contour-axes crossings
- E elongation (major-minor axes ratio).

Each descriptor is classified regarding to models and the corresponding outputs are combined to get the final decision. Let us further describe the feature descriptors defined to characterize a shape:

3.1 Zoning-Based Descriptors

Zoning is a well-known technique that describes a shape in terms of the spatial distribution of a set of feature points along a predefined lattice of regions or zones covering the shape. In our case, as ears usually have a shape similar to an ellipse, an ellipsoidal mask to extract the features of the same scale as the ear is constructed Fig 8(b). The ellipse is divided into 12 sectors, i.e, 30 degrees for each sector and each of these ones in 6 concentric rings. Two zoning descriptors are defined. Let us describe them.

1. The first feature extracted using the zoning mask is the density of edge points in each zone. This characteristic is computed dividing the number of pixels of the edge inside the zone by the number of pixels of the whole zone. The Euclidean distance is used to compare zonings of different ears in the recognition process:



(a) (b) (c)

Fig. 8. (a) Ear Axes (b) Zoning Mask (c) Contour-Axes Crossings

$$d_1(z_1, z_2) = \sum_{i=1}^{n} |z_1^i - z_2^i| \tag{1}$$

where z_j^i is the value of the i-zone of the j-zoning and n is the number of zones of the mask.

2. The second feature extracted using the zoning mask is the mean angle of the orientation of the edge pixels in each zone. The mean angle, α_m, is computed as follows:

$$\alpha_m(\alpha_1, ..., \alpha_n) = \frac{1}{2} \arctan \frac{\sum_{i=1}^{n} \sin(2\alpha_i)}{\sum_{i=1}^{n} \cos(2\alpha_i)} \tag{2}$$

Where $\alpha_i \ i = 1, ..., n$ are angles running on [0,180] and α_m is the mean orientation. The distance between two angles is given by:

$$d_\alpha(\alpha_1, \alpha_2) = min(|\alpha_1 - \alpha_2|, 180 - |\alpha_1 - \alpha_2|) \tag{3}$$

Where α_1 and α_2 are two angles running on [0,180]. The distance used to compare these zonings is:

$$d_2(z_1, z_2) = \sum_{i=1}^{n} d_\alpha(z_1^i, z_2^i) \tag{4}$$

3.2 Contour-Axes Crossings

It is another well-known shape signature. In general the idea is to measure the distribution of crossings between contour shapes and a set of reference axes strategically distributed. In our case 17 radial straight lines regularly distributed at a frequency of 10 degrees and centered at O. This descriptor computes the distribution of the intersections between ear diameters and the *Canny* edges in the inner ear, see Fig 8(c). To produce a normalized distribution the value assigned to each intersection is saved as its Euclidean distance to O divided by the half of the length of the major axe. Depending on the side where the intersection is found a different sign is assigned: if the intersection is over Ax the positive sign is preserved and if the intersection is under Ax the sign is changed

to negative. This way the values of the distribution are always on [-1,1]. For each ear a vector with the information about the intersections is saved in the database. The vector is the following :

$$V = \{[O_0, I_0^1, ..., I_0^{n0}], ..., [O_i, I_i^1, ..., I_i^{ni}], ..., [O_{17}, I_{17}^1, ..., I_{17}^{n17}]\} \quad (5)$$

Where O_i is the orientation of the i-diameter, I_i^j is the $j-$intersection in the i-diameter, n_i is the number of intersections in the i-diameter.

Given two contour-axes crossing signatures, the similarity between them is computed as follows. The Euclidean distance between all the intersections of one distribution is computed over all the intersections of the other distribution. Next, the smallest distance is taken and if this value is under a predefined threshold this distance is saved. The elements that produce this minimum value are removed (the elements have matched). This process is repeated until one of the distributions has no elements or until the minimum distance is higher than the predefined threshold. At this point, a vector with the distances of the matching intersections is obtained. The maximum value of this vector is selected as the distance between the distributions.

4 Identification Process

Using the descriptors explained in section 3 a several number of combinations of descriptors (signatures) have been tested to check which signature is the best to recognize newborns in our framework. The different signatures tested have been: $S_1 = [C]$, $S_2 = [Z_\alpha]$, $S_3 = [Z_d]$, $S_4 = [E]$, $S_{1,2} = [C, Z_\alpha]$, $S_{1,3} = [C, Z_d]$, $S_{2,3} = [Z_\alpha, Z_d]$, $S_{1,2,3} = [C, Z_\alpha, Z_d]$ and $S_{1,2,3,4} = [C, Z_\alpha, Z_d, E]$, see section 3 for an explanation of each descriptor.

To compare the values of the different components the distances and measures of similarity explained in the previous section are used: d_1 for density zonings, d_2 for angle zonings, the measure of similarity explained in section 3.2 for contour-axes crossings and finally the *Euclidean* distance is used to compare elongations.

Once all the descriptors of the ear models have been computed and stored in the database, the final result in the recognition process using a new ear image is obtained as follows (an example using the signature $S_{1,2,3,4} = [Z_d, Z_\alpha, C, E]$ with five models registered in the database is shown in Table 1). For each component of the current ear the corresponding similarity value with the same component of all the models in the database is obtained and with it a score is computed. The class whose similarity value is the lowest one is assigned the value 1, the second one is assigned the value 2 and so on (rows in Table 1). This process is performed for all the components of the signature. The scores each class of the database have obtained are summed (columns in Table 1). For each class of the database a similarity value is obtained. Now, a new similarity vector is obtained (last row in Table 1). The class with minimum sum of scores is returned. In the example of Table 1 the model returned by the algorithm would be the model *Ear3* which has the smallest sum of scores, 9.

Table 1. Score table

Current Ear	Classes inside the database Ear1	Ear2	Ear3	Ear4	Ear5
Z_α	2	3	1	4	5
Z_d	1	3	5	4	2
C	3	5	1	2	4
E	4	3	2	1	5
Sum of scores	10	14	9	11	16

5 Experimental Results

Although our project is oriented to newborn recognition the experiments for quantitative evaluation have been carried out using an adult ear database. This fact has been motivated because the project is in a preliminary stage and the newborn ear database is not great enough to obtain representative results. In spite of this fact the analysis performed over the newborn images that we have collected until now let us think that the recognition results obtained using adult images will be very similar or outperformed using the newborn ones due to the great variety of ear shapes observed in newborn ears.

As it has been said above the experiments have been carried out using our own database of adult ear images. In this moment the database consist of 140 images of 14 people. All the images were taken with the camera perpendicularly to the ear and avoiding brightness points. Trying to emulate a real process one of the images randomly selected of each ear is used as model in the registration process, and only one ear per person have been used in the recognition step. The rest of the images acquired have been used in the identification process.

The individual recognition ratios of each descriptor used and the ratio of the different signatures tested are shown in Table 2 and are graphically displayed in Fig 9. In both cases we can observe that results are grouped into three different groups: 1 NClass, 2 NClass, 3 NClass. A group i NClass shows the recognition values obtained using the i nearest classes, using the class ordination produced after computing the voting scheme among the input image and the registered ones.

Studying the obtained results the following conclusions can be extracted:

– Analyzing the recognition results obtained by S_1, S_2, S_3 and S_4 (signatures using only individual descriptors) it can be observed that S_1, S_2 and S_3

Table 2. Identification percentages of the whole set of signatures

	S_1	S_2	S_3	S_4	$S_{1,2}$	$S_{1,3}$	$S_{2,3}$	$S_{1,2,3}$	$S_{1,2,3,4}$
1 NClass	68	71	73	40	76	76	78	86	82
2 NClass	81	86	79	51	87	90	93	97	86
3 NClass	86	90	87	55	94	96	98	99	92

obtain similar rates around 70% using only the nearest class (1 NClass), around 83% using the two nearest classes (2 NClass), and around 88% using the three nearest classes (3 NClass). However, the fourth signature, S_4, does not obtain similar results, but much lower. Using the 1 NClass the recognition rate is only around 40% and using the 3 NClass the rate does not arrive to 60%. With these values we can conclude that the descriptor used in the fourth signature, S_4, is not characteristic enough, i.e., not useful to use in ear recognition.

- Analyzing the recognition results obtained by the signatures that combine 2 descriptors, $S_{1,2}$, $S_{1,3}$ and $S_{2,3}$, it can be observed that the recognition rates are also very similar among them and, in general, higher than the rates obtained by the signatures that only use one descriptor (S_1, S_2, S_3 and S_4). In this case the recognition values reached using the 3 NClass are between 94% and 98%.
- Finally, two signatures combining 3 and 4 descriptors have been tested. The signature that combines all the descriptors, $S_{1,2,3,4}$ uses the elongation which as it has been explained before it gets a very low recognition value. For this reason although being the signature that uses more descriptors is not the signature with the best results.
  The signature that combines the other three descriptors, $S_{1,2,3}$, obtains the best results and only using the 1 NClass the recognition ratio is higher than 85%. Using the 2 NClass and 3 NClass the recognition ratio is around 97% and 99% respectively.

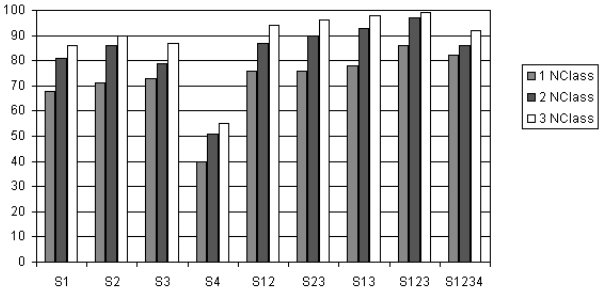


Fig. 9. Graphic of recognition percentages using the different signatures

6 Conclusions

In this paper an algorithm to recognize newborns using the biometric information extracted from the ear has been proposed. The algorithm presented begins with the image acquisition, goes on with the ear detection and normalization process of the ear an ends with the recognition process.

In the recognition process some combinations of different shape descriptors (signatures) have been tested to check which signature achieve the best results.

As can be observed in section 5 in general the signatures with only one descriptor achieve similar recognition ratios around 70% using only the nearest class. In the case of signatures that combine two descriptors it happens the same, this time around 77% using only the nearest class. The worst results are obtained using S_4 which is the signature that uses only the elongation of the ear in the recognition process, while the best results are obtained using $S_{1,2,3}$, where only using the 1 NClass the recognition ratio is higher than 85% and using the 2 NClass and 3 NClass the recognition ratio goes up to 97% and 99% respectively. Therefore, we conclude that a first approach to newborn recognition using the ear shape can be performed using these descriptors.

References

1. Anil K. Jain, 'An Introduction to Biometric Recognition'. IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics, Vol. 14, No. 1, January 2004.
2. A. Serrau, G.L. Marcialis, H. Bunke and F. Roli. 'An Experimental Comparison of Fingerprint Classification Methods Using Graphs'. GbRPR. pp. 281-290, 2005.
3. V. Govindaraju, Z. Shi and J. Schneider. 'Feature Extraction Using a Chaincoded Contour Representation of Fingerprint Images'. AVBPA. pp. 268-275, 2003.
4. Z. Xu, X. Guo, X. Hu, X. Chen and Z. Wang. 'The Recognition Based on Shape for Blood Vessel of Ocular Fundus'. Proceedings of Sixth IAPR International Conference on Graphics Recognition (GREC2005), pp. 129-135, 2005.
5. M. Choras 'Ear Biometrics Based on Geometrical Method of Feature Extraction'. AMDO 2004, LNCS 3179, pp. 51-61,2004.
6. Z. Mu, 'Shape and Structural Feature Based Ear Recognition'. Sinobiometrics 2004, LNCS 3338, pp. 663-670, 2004.
7. W.J. Babler. 'Embryologic Development of Epidermial Ridges and Their Configuration'. Birth Defects Original Article Series, vol. 27, no. 2, 1991.
8. A.F. Cervantes 'Biometric Newborn Identification'. Master Thesis, Universitat Autnoma de Barcelona - Computer Vision Center, september, 2005.
9. M. Burge 'Ear Biometrics' Johannes Kepler University, Linz, Austria 1999.
10. David J. Hurley, 'Force Field Feature Extraction for Ear Biometrics'. Computer Vision and Image Understanding 98 (2005) pp. 491-512.
11. S. Loncaric 'A Survey of Shape Analysis Techniques' Pattern Recognition. Vol 31, No 8, pp. 983-1001, 1998.
12. D. Zhang 'Review of Shape Representation and Description Techniques' Pattern Recognition. Vol 37, pp. 1-19, 2004.
13. Remco C. Veltkamp and Michiel Hagedoorn. 'State-of-the-art in shape matching'. Technical Report UU-CS-1999-27, Utrecht University, the Netherlands, 1999

The Third Report of the Arc Segmentation Contest

Liu Wenyin

Dept of Computer Science, City University of Hong Kong, Hong Kong SAR, PR China
csliuwy@cityu.edu.hk

Abstract. The Arc Segmentation Contest, as the sixth in the series of graphics recognition contests organized by IAPR TC10, was held in association with the GREC'2005 workshop. Three systems have participated in the contest. In this paper we present a brief summary: the contest rules, the updated performance metrics, test images and their ground truths, and the outcomes.

1 Introduction

This contest on arc segmentation held at the sixth International Workshop on Graphics Recognition (GREC'2005), Hong Kong, August 25-26, 2006 is the sixth in the series of graphics recognition contests and the 3rd on arc segmentation in particular, organized by the International Association for Pattern Recognition's Technical Committee on Graphics Recognition (IAPR TC10). A brief history and the first report of the contest series is presented in [1] and the second edition is reported in [2]. The purpose of this series of contests is to encourage third-party independent and objective evaluation of the industrial and academic solutions to the graphics recognition problem and therefore push the research in this area.

This contest is similar to the previous two on arc segmentation [1] [2], but with new test images and updated performance metrics. In this paper we briefly present the final report of the contest, including the contest rules, test images and their ground truths, the updated performance metrics, the winners and their performance, and discussions.

2 General Rules

The rules are exactly the same with the previous two [1] [2], but with new test images and updated performance metrics, except for the new test images. Main contest rules are summarized below.

- Recognition accuracy was measured on only solid arcs.
- In total 18 images (6 real life scanned drawing images and their 12 noisy versions) were tested. See Section 0 for detail descriptions of these test images.
- An overall average score based on each image's VRI [3] was used as the unique measure of performance of each participant's system. VRI is changed this time to $VRI=\text{sqrt}(D_v*(1-F_v))$, which is different from the one used for the previous two contests. The performance evaluation software is also available at the contest website [4].

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 358 – 361, 2006.
© Springer-Verlag Berlin Heidelberg 2006

3 Test Images and Their Ground Truths

In total we have used 18 test images. Six of them are generated by scanning six paper drawings in 256 grayscales and then binarizing with moderate thresholds. Random noise and salt and pepper noise are then added to each of these six images to generate the rest 12 images. **Fig. 1** shows the most difficult test image (7_sp.tif) and its ground truth arcs (in gray). All these test images and their and ground truth files can be downloaded at the contest website [4] and the effect of the ground truths arcs over the test images are shown by Keysers and Breuel [8].

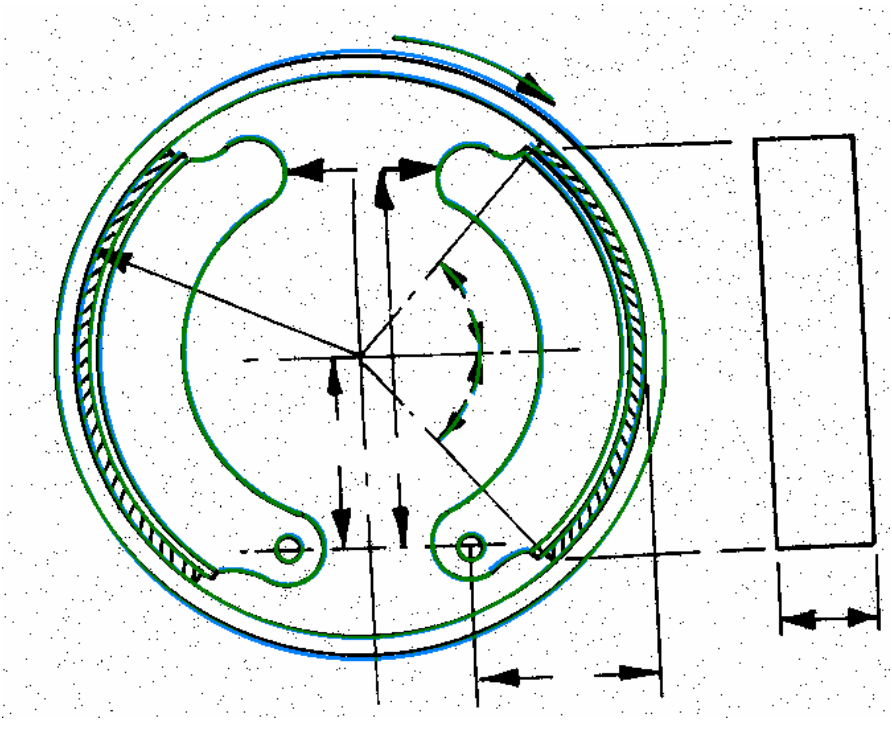


Fig. 1. Comparison of the ground truth arcs and the test image

4 Winners and Their Scores

This time, we have three participants:

- Dave Elliman [5], University of Nottingham, UK
- Daniel Keysers & Thomas Breuel [7], University of Kaiserslautern & DFKI, Germany
- Xavier Hilaire [6], LORIA - Universite Henri Poincare, Nancy, France
- The scores (*VRI*s) of their systems are listed in Table 1. A number in bold means the best score for that image and an italic number means a slightly lower score

than the best for that image. Finally, Hilaire's system obtained an overall score of 0.801, which is the best among the three participants.

Table 1. The scores of the participants

Image (*.tif)	Elliman	Keysers & Breuel	Hilaire
5	0.119	0.591	**0.904**
6	0.896	0.796	**0.939**
7	0.092	0.268	**0.404**
8	**0.760**	*0.729*	*0.736*
9	0.855	0.611	**0.970**
10	0.458	0.614	**0.862**
5_rn	0.111	0.615	**0.898**
6_rn	0.852	0.774	**0.943**
7_rn	0.126	0.347	**0.444**
8_rn	*0.658*	**0.717**	*0.693*
9_rn	0.722	0.704	**0.930**
10_rn	0.585	0.576	**0.866**
5_sp	0.119	0.591	**0.910**
6_sp	0.841	0.797	**0.959**
7_sp	0.099	0.265	**0.415**
8_sp	*0.727*	*0.729*	**0.732**
9_sp	0.764	0.732	**0.961**
10_sp	0.466	0.614	**0.856**
Average	0.514	0.615	**0.801**

5 Summary and Discussion

All of the test drawings are real scanned images with/without synthesized random noise or salt and pepper noise. Some of the images (e.g., 7*.tif) are really tough, which contain many thin arcs touching with short hatching lines and arcs tangent and connected to one another, and even slightly deformed circles. All the three participants cannot perform well on 7*.tif. However, Hilaire's system obtained very well results on other images. The scores for some images are more than 0.9 and the average is 0.801, which is higher than the concluded threshold for satisfactory/acceptable results we mentioned in our last report [2]. This is also compatible with the human vision evaluation results, as we can also see from the effect shown by Keysers and Breuel [8] that Hilaire's results on the images except for 7*.tif are really very good. However, his results are not consistently better than others. All the three participants' scores on 8*.tif are very close and Hilaire's even got lower scores on some of them. We expect the participants to help investigate the reason(s).

The impact of noise is still not big for the noise level we used. Some noisy versions obtained even high scores than their clean versions. Hence, we can conclude that moderate noise does not affect the performance too much.

In this edition, VRI is defined as the geometric average instead of arithmetic average of the two rates, to avoid the cases with no or few detections. It works well and its difference with the previous VRI is very small in normal cases.

We are happy that we had successfully attracted new participants and made the number of participants greater than two. We hope we can attract more participants in future contests and accumulate more and more data for a more comprehensive understanding of arc segmentation algorithms. In addition, we welcome suggestions to further improve the contest series in terms of all aspects, including organization, performance metrics, test images, and so on.

Acknowledgement

The participants are greatly acknowledged for their contributions to the contest. The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 1147/04E].

References

1. Liu W., Zhai J., and Dori D.: Extended Summary of the Arc Segmentation Contest. In: Graphics Recognition: Algorithms and Applications, Lecture Notes in Computer Science, Vol. 2390, Springer (2002), pp. 343-349.
2. Liu W.: Report of the Arc Segmentation Contest", In: Graphics Recognition: Recent Advances and Perspectives, Lecture Notes in Computer Science, Vol. 3088, Springer (2004), pp. 363-366
3. Liu W. and Dori D.: A Protocol for Performance Evaluation of Line Detection Algorithms. Machine Vision and Applications 9 (1997) 240-250.
4. http://www.cs.cityu.edu.hk/~liuwy/ArcContest/ArcSegContest.htm
5. Elliman D.: TIF2VEC, An Algorithm for Arc Segmentation in Engineering Drawings. In: Graphics Recognition: Algorithms and Applications, Lecture Notes in Computer Science, Vol. 2390, Springer (2002), pp. 350-358.
6. Hilaire X.: RANVEC and the Arc Segmentation Contest: Second Presentation. In: Graphics Recognition: Ten Year Review and Perspectives (Post-Proc. of GREC2005), Lecture Notes in Computer Science, Vol. 3926 (This volume), Springer (2006), pp.372-378.
7. Keysers D. and Breuel T: Optimal Line and Arc Detection on Run-Length Representations. In: Graphics Recognition: Ten Year Review and Perspectives (Post-Proc. of GREC2005), Lecture Notes in Computer Science, Vol. 3926 (This volume), Springer (2006), pp. 379-390.
8. Keysers D. and Breuel T: http://www.iupr.org/~keysers/grec2005/grec2005.html

RANVEC and the Arc Segmentation Contest: Second Evaluation

Xavier Hilaire

LORIA – Université Henri Poincaré
B.P. 239, 54506 Vandœuvre-lès-Nancy, France
hilaire@loria.fr

Abstract. This paper provides some information regarding the winning system at the GREC'2005 contest on arc segmentation. Important facts are first recalled, then the changes made on the system since its first presentation at GREC'2001 are detailed. The obtained results are briefly commented, and the paper finally concludes with some clues for possible, future improvements regarding the system.

1 Introduction

1.1 A Short Recall

The system presented in this paper was initially designed to vectorize architectural drawings (with no particular emphasis given on arc detection). It is fully described in [5, 4], and this section only aims to recall a very brief description of it.

An overview of our system is available at Fig. 1. The following steps are involved (in which τ, m, q, ρ_{min}, ρ_{max}, and θ_0 are parameters supplied by the user, see the discussion in section 3 regarding the values used for the contest):

- *Binarization and filtering*, which are optional, preprocessing steps. The filtering step removes all the black connected components whose diameter is lower than f (described below), fills the holes having the same property, and performs a mathematical closure.
- *Text elimination*, which aims at removing text (if any) in the image. This step implements the method of Tombre et al. [8], and is used with default parameters.
- *Thin/thick separation*, which finds the q modes in the thickness histogram that best explain the image in terms of thickness. Each of these layers, with upper estimated thickness f, is then processed independently by the system.
- *Skeletonization*, which computes the (3,4)-distance transform map of the source image, and deduces a skeleton from it following Sanniti di Baja's algorithm [2].
- *Segmentation*, which partitions the skeleton into a set of meaningful primitives (lines and arcs) by resorting to random sampling. The algorithm ensures that all the primitives are correctly found with a probability greater

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 362–368, 2006.
© Springer-Verlag Berlin Heidelberg 2006

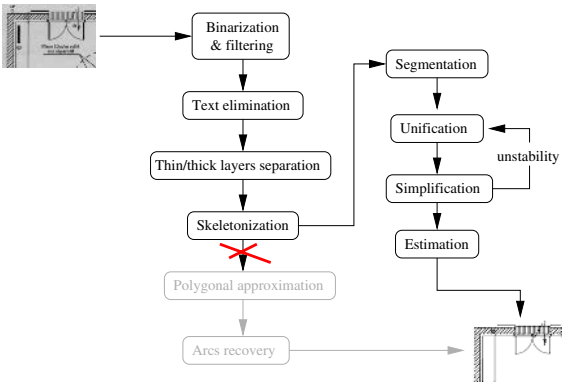


Fig. 1. An overview of the vectorization system presented at the contest

than τ, and that the pixels which define a primitive do not lie away from it further than m units. Circular arcs must also have a radius in the interval $[\rho_{min}, \rho_{max}]$, and an opening angle greater than θ_0.

- *Unification and simplification*, which are applied in turn in a loop, unify or delete primitives in order to simplify the solution. This loop removes spurious arcs or lines (which appeared either because of noise, or because of the skeletonization itself). These steps not only ensure a strict geometric equivalence at a representation point of view, but also permit to compute the preimages of all the detected primitives.
- *Estimation*, which uses the previous preimages to compute optimal estimates of the parameters of each primitive.

1.2 Some Historical Notes

The method described above was evaluated for the first time during the GREC'2001 contest on arc segmentation [3, 9]. At that time, the results were not really enthusiastic: the average VRI value did not exceed 0.63, many arcs were misdetected, and false alarms were also numerous. Indeed, most of these faults are simply explained by an early and poor implementation of the method – the program even crashed on a image, and obtained a score of zero.

To the exception of what is presented in the next section, the system used during the contest this year still follows the description of [5] and [4]. It has been reimplemented only recently[1] as a 64-bit PowerPC application, and runs on any Apple computer equipped with a G5 processor.

2 The Changes

Strictly speaking, there have been only two changes made on the vectorization method since its first presentation in [5]: the first is related to the thickness

[1] It was not possible to produce a new implementation in due time for the GREC'2003 contest.

evaluation, while the second is a revision of the reconstruction procedure. In this section, we only discuss the first change: the modification brought to the reconstruction procedure has no influence at all on the result as long as the question of the concurrency of more than two primitives does not arise, which was the case for the contest's images.

To explain the revision brought to the thickness estimation method, let us first recall that a discrete circular ring $\mathcal{R}(x_0, y_0, \rho, w)$ with center (x_0, y_0), radius ρ, and thickness w (all possibly real) is the set of *integer* points (x, y) satisfying

$$(\rho - \frac{w}{2})^2 \leq (x - x_0)^2 + (y - y_0)^2 < (\rho + \frac{w}{2})^2$$

If (x_0, y_0) is known, and the ring is drawn without noise, then finding ρ and w is straightforward. However, in real life we have to cope with noise, which complicates the problem. In [4], it has been proven, using Kanungo's document degradation model [6], that an elementary increment of the thickness of any primitive due to noise was very unlikely. On the other hand, we also know that a labeled skeleton, obtained with the $(3, 4)$-distance transform also gives us a lower estimate of the thickness at any skeletal point, and that the corresponding relative error decreases as the ground truth thickness increases [1]... This rapidly suggests us what to do:

1. Build a set E from the labeled skeleton as follows: for each skeletal point p with $(3, 4)$-DT value v, if p has less than 3 neighbors with value v, then add v, else add $v + 3/2$ to E;
2. Robustly estimate the thickness from E: $\hat{w} = \lfloor 2LMS(E)/3 \rfloor$-1, where LMS stands for *least median of squares*;
3. Let $\mathcal{I}$ be the source image, $|.|$ denote cardinality, and put

$$\Delta(\mathcal{X}, \mathcal{Y}) = |\mathcal{X} \cap \mathcal{Y}| - |\mathcal{X} \cap \mathcal{Y}^C|$$

for any discrete sets $\mathcal{X}$ and $\mathcal{Y}$. If $\Delta(\mathcal{R}(x_0, y_0, \rho, \hat{w}+1), \mathcal{I}) > \Delta(\mathcal{R}(x_0, y_0, \rho, \hat{w}), \mathcal{I})$ then retain $\hat{w} + 1$ as the thickness, else retain $\hat{w}$.

In other words, the above procedure determines a lower bound $\hat{w}$ of the thickness, and then checks whether it is more interesting to reconstruct the shape using a ring with thickness $\hat{w}$ or with thickness $\hat{w} + 1$.

3 Parameter Setup

An important aspect, often kept silent in the literature, is how to parametrize a given recognition method in order to obtain acceptable results. Although the method commented here uses a reduced number of parameters, we still have to provide values for all of them. Keeping the notations of [5, 4], these parameters are: the thickness f, the noise tolerance m, a lower bound τ on the probability to achieve a correct extraction, and, most important, validity bounds for circular patterns ρ_{min}, θ_{min}, and ρ_{max}.

All parameters were set more or less empirically. For τ, the arbitrary value of 0.9999 was used. On the opposite, setting m was driven by a clue observed in Liu and Dori's evaluation protocol [7]. To summarize this clue, let us simply recall some equations from [7]: on the one side, we have

$$Q_v(c) = (Q_{pt}(c).Q_{od}(c).Q_w(c).Q_{sh}(c).Q_{st}(c))^{\frac{1}{5}} \tag{1}$$

and

$$Q_{pt}(c) = \exp\left(-\frac{d_1(c) + d_2(c)}{W(g)}\right) \tag{2}$$

$$Q_{od}(c) = \exp\left(\frac{-2d_{overlap}(c)}{W(g)}\right) \tag{3}$$

which define the basic quality of a candidate vector against its ground truth g, given their overlapping vector c. On the other side

$$Q_{fr}(k) = \frac{\sqrt{\sum_{g \in G(k)} l(k \cap g)^2}}{\sum_{g \in G(k)} l(k \cap g)} \tag{4}$$

characterizes the fragmentation rate of a given candidate k. Now, consider the two following situations:

(1) We detect a given arc without fragmentation, but with poor accuracy ($d_1(c) + d_2(c) + 2d_{overlap}(c) \neq 0$);
(2) We detect a given arc with fragmentation $1 : n$, but with good accuracy ($d_1(c) = d_2(c) = 2d_{overlap}(c) = 0$).

Assuming that $Q_w(c) = Q_{sh}(c) = Q_{st}(c) = 1$, from equations 1,2, and 3, we obtain that the penalty in the former situation is

$$\exp\left(-\frac{d_1(c) + d_2(c) + 2d_{overlap}(c)}{5}\right)$$

while that in the latter is $1/\sqrt{n}$ according to equation 4. If we put $\varepsilon = d_1(c) + d_2(c) + 2d_{overlap}(c)$, then a glance at table 1 rapidly tells us what happens: situation 2 is less penalizing than situation 1 for a majority of cases, especially if we are concerned with thin vectors. Consequently, the m parameter of our method was set to 1, the smallest possible value we can supply to properly extract lines and circles without shifting.

Regarding the circular bounds $\rho_{min}, \theta_{min}, \rho_{max}$, the native implementation of our method offers to set both ρ_{min} and θ_{min} independently. For the purpose of the contest, we used a different version: the condition $(\rho \geq \rho_{min}) \wedge (\theta \geq \theta_{min})$ was replaced by a simple test on the length: to be accepted, a circular pattern must have a length of 15 pixels or more – an arbitrary, but common-sense value. We also set ρ_{max} to $\max(w/2, h/2)$, where h and w are the image's dimensions, which means that any circular pattern should always have a supporting circle

Table 1. Left: values of $Q_v = \exp(-\varepsilon/W(g))$. Right: first values of $Q_{fr} = 1/\sqrt{n}$ assuming a fragmentation ratio of $1 : n$.

$W(g)$ / ε	1	2	3	4	5
1	0.368	0.607	0.717	0.779	0.819
2	0.135	0.368	0.513	0.607	0.670
3	0.050	0.223	0.368	0.472	0.549
4	0.018	0.135	0.264	0.368	0.449
5	0.007	0.082	0.189	0.287	0.368

n	1	2	3	4	5	6
Q_{fr}	1	0.707	0.577	0.500	0.447	0.408

fully included inside the smallest square image that contains the source image itself.

Finally, f was set automatically, following the estimation procedure detailed in [4], with no prior thin/thick layer separation (q set to 1).

4 A Short Analysis

Although our system achieved the best overall performance, it is interesting to note that the concurrent systems did better in two cases: with image `8.tif` for Elliman's system, and with image `8_rn.tif` for Keysers' system. These images, as well as a rendering of the concurrent solutions, are presented at figure 2.

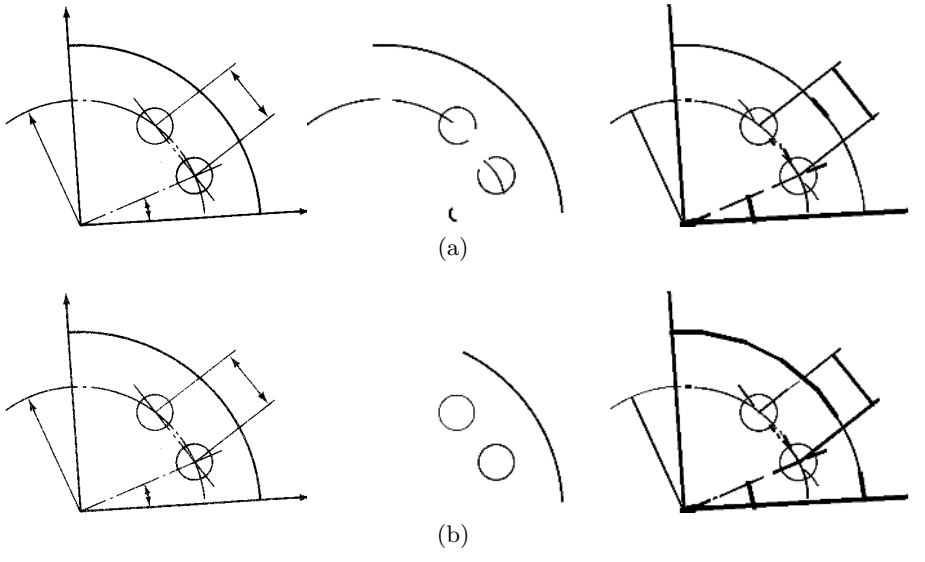


Fig. 2. Comparison of results for two particular images. (a), from left to right: source image `8.tif`, Elliman's result, our result; (b), from left to right: source image `8_rn.tif`, Keysers' result, our result.

In both cases, the lack of accuracy of our system is due to the fact that the default setting $\max(h/2, w/2)$ for the upper bound ρ_{max} was too small. As a result, in image `8.tif`, the largest arc is detected as 5 arcs and one fake segment. In image `8_rn.tif`, addition of noise worsens the situation (as m was set to 1), and this time it is detected mostly as segments. The same result can be observed on image `8_sp.tif` too.

It is also a noticeable point that other participants did not output any line in their solutions. As stated in section 3, even if a solution is fragmented or approximate but close to the ground truth, then better is to output it than keeping silent. For example, our system did not properly recognize the smallest arc in each of the `8*.tif` images, but reported a small segment instead. In image `8_rn.tif`, for example, if we remove this segment in the solution, then the VRI score drops from 0.693 to 0.687. If, furthermore, we remove all the remaining lines, then it drops to 0.675.

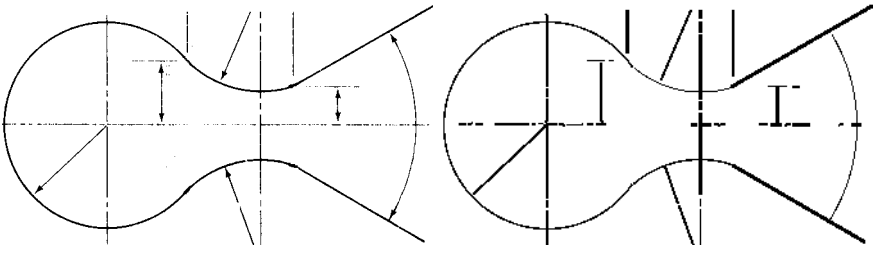


Fig. 3. The best case obtained with our system: (a) source image, (b) recognized arcs

Finally, figure 3 illustrates the best case, which occurred for image `9.tif`, and leaded to a VRI score of 0.970. The noisy versions `9_rn.tif` and `9_sp.tif` also achieve the best relative performance compared to other images. In this case, the system was well parametrized, and the result typically reflects the level of accuracy the user may expect after some suitable, circular bounds have been provided.

5 Concluding Remarks

The system we presented is actually able to extract arcs with an average VRI slightly greater than 0.8. To the best of our knowledge, it is the first time that such a result is reached since the first arc recognition contest, organized in 2001.

Besides, we believe there is still room for enhancement in future versions: although the system achieves optimal parameter estimation once the primitives are identified, the risk that the primitives have not been correctly extracted is still not null. Also, the system relies on skeletonization, and there are obvious situations in which it is still impossible to provide a correct solution given that fact. These are the two tracks currently followed to perfect it.

Acknowledgments

Most of the research presented here has been jointly supported by the French National Agency for Research and Technology (ANRT) and FS2i Corp.[2] under a CIFRE grant.

References

1. G. Borgefors. Distance Transforms in Digital Images. *Computer Vision, Graphics and Image Processing*, 34:344–371, 1986.
2. G. S. di Baja. Well-Shaped, Stable, and Reversible Skeletons from the (3,4)-Distance Transform. *Journal of Visual Communication and Image Representation*, 5(1):107–115, 1994.
3. X. Hilaire. RANVEC and the Arc Segmentation Contest. In D. Blostein and Y.-B. Kwon, editors, *Graphics Recognition – Algorithms and Applications*, volume 2390 of *Lecture Notes in Computer Science*, pages 359–364. Springer-Verlag, 2002.
4. X. Hilaire. *Segmentation robuste de courbes discrètes 2D et applications à la rétroconversion de documents techniques.* Thèse de doctorat, Institut National Polytechnique de Lorraine, 2004.
5. X. Hilaire and K. Tombre. Improving the Accuracy of Skeleton-Based Vectorization. In D. Blostein and Y.-B. Kwon, editors, *Graphics Recognition – Algorithms and Applications*, volume 2390 of *Lecture Notes in Computer Science*, pages 273–288. Springer-Verlag, 2002.
6. T. Kanungo, R. M. Haralick, H. S. Baird, W. Stuezle, and D. Madigan. A Statistical, Nonparametric Methodology for Document Degradation Model Validation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1209–1223, November 2000.
7. W.Y. Liu and D. Dori. A protocol for performance evaluation of line detection algorithms. *Machine Vision and Applications*, 9(5-6):240–250, 1997.
8. K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and Ph. Dosch. Text/Graphics Separation Revisited. In D. Lopresti, J. Hu, and R. Kashi, editors, *Proceedings of 5th IAPR International Workshop on Document Analysis Systems, Princeton (NJ, USA)*, volume 2423 of *Lecture Notes in Computer Science*, pages 200–211. Springer-Verlag, August 2002.
9. L. Wenyin, J. Zhai, and D. Dori. Extended Summary of the Arc Segmentation Contest. In D. Blostein and Y.-B. Kwon, editors, *Graphics Recognition – Algorithms and Applications*, volume 2390 of *Lecture Notes in Computer Science*, pages 343–349. Springer-Verlag, 2002.

[2] FS2i – www.fs2i.fr – 8 impasse de Toulouse, 78000 Versailles, France.

Optimal Line and Arc Detection on Run-Length Representations

Daniel Keysers and Thomas M. Breuel

Image Understanding and Pattern Recognition Research Group
German Research Center for Artificial Intelligence (DFKI GmbH)
and University of Kaiserslautern
D-67663 Kaiserslautern, Germany
{daniel.keysers, thomas.breuel}@dfki.de

Abstract. The robust detection of lines and arcs in scanned documents or technical drawings is an important problem in document image understanding. We present a new solution to this problem that works directly on run-length encoded data. The method finds globally optimal solutions to parameterized thick line and arc models. Line thickness is part of the model and directly used during the matching process. Unlike previous approaches, it does not require any thinning or other preprocessing steps, no computation of the line adjacency graphs, and no heuristics. Furthermore, the only search-related parameter that needs to be specified is the desired numerical accuracy of the solution. The method is based on a branch-and-bound approach for the globally optimal detection of these geometric primitives using runs of black pixels in a bi-level image. We present qualitative and quantitative results of the algorithm on images used in the 2003 and 2005 GREC arc segmentation contests.

Keywords: Graphics Recognition, Line Drawings, Technical Drawings, Branch-and-Bound algorithms.

1 Introduction

A large amount of technical and engineering information is still available only in paper form. The goal of graphics recognition is to make that information available in electronic format. After initial scanning, a key problem in the conversion of drawings into electronic form is the reliable detection of the geometric primitives from which those drawings are constructed [1].

This problem has been the subject of intensive research, and it has even been the subject of several graphics recognition contests (held in conjunction with the GREC workshops; please refer to [2,3] and the references therein for an overview; a report of the 2005 contest is also included in this proceedings volume).

A variety of approaches to line detection have been described in the literature, many of them based on the Hough transform [4]. A survey of various methods is given in [5]. Many approaches also first perform a vectorization, i.e. they determine a representation by poly-lines and then use this intermediate step for line or arc detection, e.g. [6,7]. One method along this line that performed very

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 369–380, 2006.
© Springer-Verlag Berlin Heidelberg 2006

well in the 2005 GREC arc segmentation contest is an extension of the method described in [8].

While these existing approaches have found numerous practical applications, they have some difficulties and limitations. Hough transforms, for example, require careful choice of parameters like bucket size in order to reduce the risk of losing solutions. Preprocessing steps like vectorization and polygonalization, are themselves complex processing steps, and, since they generally only represent approximations to the original bitmap, also potentially reduce the accuracy and reliability of the overall system.

This paper introduces an algorithm for line and circle detection that operates directly on a pixel-accurate representation of the original binary image. This avoids the complexity and potential for errors introduced by separate preprocessing steps. Furthermore, it yields optimal solutions for a broad class of well-defined geometric and statistical models of thick lines.

The basis of the algorithm is the RAST algorithm (Recognition by Adaptive Subdivisions of Transformation Space [9]), which is guaranteed to find globally optimal solutions under given error models and requires no parameters other than desired numerical accuracy to be specified by the user. RAST algorithms are closely related to (hierarchical) Hough transforms but have more desirable combinatorial properties for object recognition. Recently, effective and simple to implement variants of RAST algorithms that are based on interval arithmetic have been introduced [10]. The RAST algorithm avoids the drawback of fixed parameter discretization by subdividing regions of the parameter space adaptively only when it is possible that a desired solution exists within that region of the parameter space. The closely related adaptive Hough transform uses the same strategy, but cannot guarantee to find the global optimum because it does not keep track of intermediate scores and thus cannot recover from a decision to drop a portion of the parameter space from the search.

The work described in this paper is directly based on the principles presented in [10,11], where these principles have been applied to settings in which not pixels but the outlines of connected components were used. The application of branch-and-bound algorithms to matching problems has been described in the computer vision literature before [9,12,13,14,15,16].

In our approach, we use run length coding as the representation of the image content. Run-length coding for images is commonly used for in-memory representation of large binary images, thereby making it easy to apply the algorithm to potentially large images in memory. Run-length coding is also familiar as an important intermediate step in some binary image compression methods.

Run-length coding is not only a convenient representation in terms of memory, it also serves a semantic function: in a binary image, each line becomes a collection of adjacent runs in the run-length encoded representation. Previous approaches to line detection in images have attempted to take advantage of this representation by considering the line adjacency graph and, for example, applying Hough transform methods to the midpoints of each run [17]. We use run-length coding in a slightly different way. Rather than using the midpoints

of each run, we match the endpoints of each run against the two boundaries of a thick line model.

The presented approach separates the search algorithm from the quality function used. This modularity makes it easy to also incorporate and test other quality functions or use other representations of the image content. For example, the algorithm can also be used on a quality function using segments of a polygonal approximation of the edges present in the image.

In the rest of the paper, we first describe the framework for optimal matching and its use of interval arithmetic, followed by a description of the specific quality functions used for the detection of the thick lines and arcs. We then present results on example images of scanned technical drawings, briefly discuss the results of the GREC 2005 arc segmentation contest, and end with concluding remarks.

2 Detection Framework

Assume we are given a set $R = r_1, \ldots, r_N$ of pixel runs in the image plane, where each run $r = (x_0, x_1, y)$ is defined by its starting point (x_0, y) and its end point (x_1, y). The computation of this set from a given image is straight-forward. Note that the decision to choose runs in horizontal direction is arbitrary and only based on the most common pixel order found in the representation of digital images. The algorithms presented here would work equally well with runs of pixels in a diagonal direction.

Now we are interested in finding the best matching geometric primitives with respect to the set R. The geometric primitives are characterized by a set of parameters $\vartheta \in T$, e.g. center, radius, and thickness for a circle (cp. Section 4). Note that we avoid adding additional parameters for the start and end points of lines and arcs here, because doing so would enlarge the search space dimensionality. Instead, we evaluate all runs associated with a primitive after the search and then determine the parameters of the line or arc based on these.

We perform the detection by finding the maximizing set of parameters

$$\hat{\vartheta}(R) := \arg\max_{\vartheta \in T} Q(\vartheta, R) \tag{1}$$

where the total quality $Q(\vartheta, R)$ of a parameter set is defined as the sum of local qualities

$$Q(\vartheta, R) := \sum_{n=1}^{N} q(\vartheta, r_n) \tag{2}$$

where $q(\vartheta, r)$ is a local quality function that evaluates the goodness of fit for a given run of pixels r and a set of parameters ϑ.

This maximization will be a complex task for most functional forms of Q. In many applications, such fits of parameters are carried out iteratively and heuristically, which involves the risk that the results found are only locally optimal solutions. Other methods include randomized approaches like e.g. random sample consensus [18].

We propose to employ a branch-and-bound technique [9,16,19] to perform this task that guarantees to find the globally optimal parameter set by recursively subdividing the parameter space and processing the resulting parameter hyper-rectangles in the order given by the upper bound on the total quality. Moreover, the algorithm allows us to efficiently determine the k best matches, not only the best match.

The computation of an upper bound on the quality of any geometric primitive with parameters in a hyper-rectangular region T is given by

$$\max_{\vartheta \in T} Q(\vartheta, R) \leq \sum_{n=1}^{N} \max_{\vartheta \in T} q(\vartheta, r_n), \tag{3}$$

where $\max_{\vartheta \in T} q(\vartheta, r_n)$ is straightforward to compute. Note that the computation of the upper bound is implicitly handled using interval arithmetic as described in Section 3. We can now organize the search as follows (for details see [9,19]):

1. Pick an initial region of parameter values T containing all the parameters that we are interested in.[1]
2. Maintain a priority queue of regions T_i, where we use as the priority the upper bound on the possible values of the global quality function Q for parameters $\vartheta \in T_i$.
3. Remove a region T_i from the priority queue; if the upper bound of the quality function associated with the region is too small to be of interest, terminate.[2]
4. If the region is small enough to satisfy our accuracy requirements, accept it as a solution; remove all runs involved in that solution from further consideration and continue at Step 3.
5. Otherwise, split the region T_i along the dimension furthest from satisfying our accuracy constraints and insert the subregions into the priority queue; continue at Step 3.

This algorithm will return all maximum quality matches greedily in decreasing order of total quality. Note that the optimality is only guaranteed for the best match, then for the second match excluding the already matched components, and so on. We cannot guarantee optimality for the combination of all returned matches, which is a problem of much higher complexity. To make this sequential approach practical and avoid duplicate computations, we use a matchlist representation [9]. That is, with each region kept in the priority queue in the algorithm, we maintain a list (the matchlist) of all and only those runs that have the possibility to contribute with a positive local quality to the global quality.

[1] The initial region can always be picked large enough to include all possible arcs and lines within a given image and thus it is not an additional parameter of the algorithm. If we decide to choose a smaller initial region, this is because we want to exclude certain solutions.

[2] The choice of the parameter for termination does not influence the search itself. It is rather a convenience parameter that lets the algorithm terminate before all runs have been process, if it is known that we require a certain minimum quality.

These matchlists will be large for the initially large regions of the parameter space but shrink with decreasing size of the regions T_i. The justification for the use of matchlists is that runs for which the distance from the geometric primitive is surely greater than half the maximum line thickness d that is contained within T_i do not contribute to the computation of the quality function at all. Furthermore, it is easy to see that the upper bound of a parameter space region T_i is also an upper bound for all subsets of T_i. When we split a region in Step 5, we therefore never have to reconsider runs in the children that have already failed to contribute to the quality computation in the parent.

The use of matchlists enables us to add another feature to the algorithm: Before the execution of Step 5 above, we introduce another conditional processing of the region T_i. If the geometric primitive described by T_i is associated with a matchlist of runs that can be split into two parts that are sufficiently separated (say, by 10 pixels), split the matchlist into these two parts and re-insert the region T_i into the queue twice, once with each part of the matchlist associated. Doing so avoids the unification of primitives that belong to separated regions in the image. It furthermore separates noise components from the matchlists, which then can be pruned in the further search.

After running the algorithm, the bounding boxes for the lines are determined by considering all associated runs in the matchlists and for circular arcs the smallest arc that encompasses all runs in the matchlist.

3 Use of Interval Arithmetic

One of the key points in the RAST algorithm is the calculation of the upper bound of the quality function for a subset of the parameter set. This upper bound can be determined using interval arithmetic both efficiently from a computational point of view and from the point of view of coding the actual function.

Interval arithmetic was originally proposed as a means of bounding the error of numerical computations. For detailed information about interval arithmetic and its implementation, the reader is referred to the references [20,21].

The extension of the basic numerical operators to intervals is straight-forward. Instead of a rigorous definition, let us give two examples of such extensions to illustrate the principle. The operators + and × are defined on intervals in the following way:

$$[x_0, x_1] + [y_0, y_1] := [x_0 + y_0, x_1 + y_1]$$
$$[x_0, x_1] \times [y_0, y_1] := [\min(x_0y_0, x_0y_1, x_1y_0, x_1y_1), \max(x_0y_0, x_0y_1, x_1y_0, x_1y_1)]$$

Thus, any addition of numbers from $[x_0, x_1]$ and $[y_0, y_1]$ will result in a number from $[x_0, x_1] + [y_0, y_1]$, and analogously for the multiplication operation. Due to monotonicity the extensions for log() and exp() are simple and for other functions like e.g. sin() a slightly more elaborate analysis leads to the appropriate formulation.

If the quality function used in the RAST algorithm is a composition of simpler functions like multiplication, addition, sine, minimum, square, etc., the function

can be directly implemented using the basic interval arithmetic functions. In other cases, it might be more straight-forward to base parts of the determination of the upper bound on the interval arithmetic, like e.g. the calculation of the distance of a pixel from a line, and separately code other parts.

The use of interval arithmetic instead of an explicit derivation of bounding functions, as used in prior work on branch-and-bound matching algorithms, has several advantages [10]: In many geometric matching problems of practical interest, deriving bounding functions and implementing them correctly can be difficult and error prone. This problem is substantially alleviated by using interval arithmetic, which allows us to simply replace calculations on scalars by the corresponding calculations on intervals in many cases. Furthermore, the use of interval arithmetic also makes geometric matching numerically reliable, i.e. we can guarantee that no solutions are lost due to roundoff errors. This last statement is based on the use of interval functions that guarantee to include the real-valued result within an interval bounded by finite-precision floating-point numbers as representable on a computer [20].

4 Quality Functions for the Detection of Lines and Circular Arcs

For the detection of the geometric primitives considered in this work we use the following parameterizations. The line model is parameterized by its angle φ, its distance s from the origin, and its thickness d. The circle (and circular arc) model is parameterized by the coordinate of its center point (x, y), its radius s and its thickness d.

During the search algorithm, each partition of the search space is described by a Cartesian product of intervals for the parameters, i.e. for a line by a set of the form $T = [\varphi_0, \varphi_1] \times [s_0, s_1] \times [d_0, d_1]$ and for a circle by a set of the form $T = [x_0, x_1] \times [y_0, y_1] \times [s_0, s_1] \times [d_0, d_1]$.

Given a run $r = (x_0, x_1, y)$ we consider the starting point of the run (x_0, y) and the end point (x_1, y) and determine the interval of possible distances from the center of the thick line or circle, respectively. We then compute an upper and lower bound for q in the straight-forward way. The total upper bound of the quality Q is then used in the branch-and-bound algorithm.

The local quality function for a run and a geometric primitive (line or circle) is then defined to be

$$q(\vartheta, (x_0,x_1,y)) = \max\left(0, 1-\frac{\left(\frac{d}{2}-d_\vartheta(x_0,y)\right)^2}{\sigma^2}\right) * \max\left(0, 1-\frac{\left(\frac{d}{2}-d_\vartheta(x_1,y)\right)^2}{\sigma^2}\right) \quad (4)$$

where $d_\vartheta(x, y)$ denotes the (absolute) distance of the point (x, y) from the center-line of the thick geometric primitive defined by the parameters ϑ. This quality function reaches its maximum value of one if the endpoints of the run coincide with the predicted endpoints given the parameters of the primitive. Deviations of the positions from their predicted values are tolerated up to a value of σ

with decreasing quality values. The parameter σ was set to 0.5 pixels in the experiments.

It is of course possible that we may find quality functions that are more suitable for the task discussed here and one can think of a variety of other functions measuring the goodness of match between a run and a line with finite thickness. For example, the relative overlap of runs and lines is better suited for the detection of horizontal lines based on runs than using the end points of the runs. However, this drawback can be avoided by running the algorithm once for lines $\pm\pi/4$ from the vertical axis, then rotating the remaining image elements by $\pi/2$ and repeating the detection. We experimented with a few other possibilities for the quality function like e.g. the relative overlap of a run (x_0, x_1, y) and the expected y-run of the line. Among these, the informal experiments suggested that the local quality function (4) produces the best results.

5 Results

We present experimental results for the images of the GREC arc segmentation contests that took place in the years 2003 and 2005.

5.1 Images from the 2003 Arc Segmentation Contest

We applied the algorithm to unprocessed images that were used in the fifth contest on arc segmentation at GREC 2003 to show the applicability and the quality of the results. These images show scanned technical drawings.

Each of the rows in Figure 1 shows (in that order) the original image, the union of all detected primitives, the detected circular arcs, the detected lines on the remaining runs not covered by arcs, and the remaining runs in the image that were not explained by any arc or line.

In some cases errors occur where geometric primitives extend over a larger range than visually appropriate. In that case e.g. arcs extend too far and thus prevent the detection of complete lines. It is also interesting that the remaining runs in the image often correspond to arrow-heads as these are not included in our model of the geometric primitives.

In the 2003 contest, in total 12 images were supplied for the test runs. Each of the four images shown in Figure 1 was additionally used in two version with artificially added noise. The participants in the 2003 contest obtained average VRI scores of 0.609 and 0.487, respectively [2]. Using the proposed method and the evaluation software corresponding to the 2003 contest, we obtained an average VRI score of 0.757, i.e. better than the two participating methods in 2003. Note, however, that the winning method of 2005 did not participate in 2003. Note also that although we did not specifically tune the parameters to these data, the comparison of the scores is not entirely fair because we had access to the images before actually running the evaluation.

5.2 The 2005 Arc Segmentation Contest

We participated with the presented method in the GREC 2005 arc segmentation contest. As detailed in the contest overview, which the reader can find in this

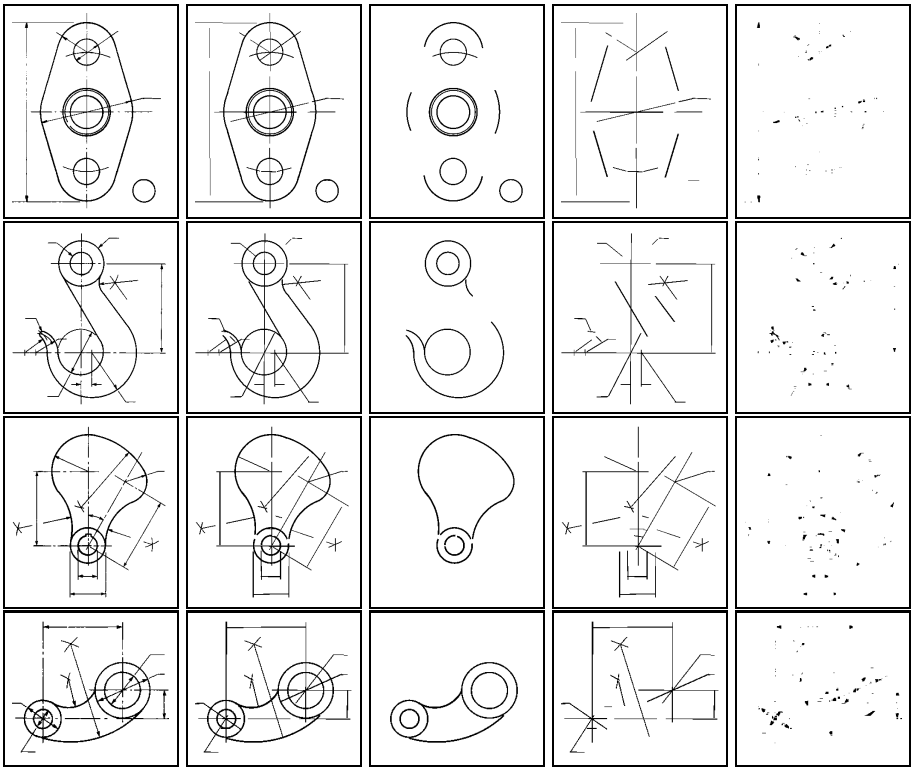


Fig. 1. Results for the detection of arcs and lines using the proposed method. Each row shows (left to right) input, detected arcs and lines overlaid, detected arcs, detected lines, remaining runs not explained.

proceedings volume, the method obtained the second place among the three participating systems with an average VRI score of 0.615. The scores of the competing systems were 0.801 for the winning method and 0.514 for the method that obtained the third place.

The contest used six images, again each in an original version and with two different types of noise added. Figure 2 shows each of the original images along with the detection result of our method and the ground truth arcs. The results for the noisy images are very similar[3]. The only preprocessing that was performed was the application of the `pbmclean` program from the NetPBM library, which removes single pixels that are not part of any connected component in the image.

Looking at the results, we can observe that the most crucial problem for the algorithm is the following: When two or more image objects touch or overlap, the pixel runs do not coincide with the expected pixel runs of the geometric primitive any more. In the present formulation of the quality function, this is not considered, which leads to the majority of the observed errors.

[3] The complete set of results can be found at http://www.iupr.org/~keysers/grec2005/grec2005.html.

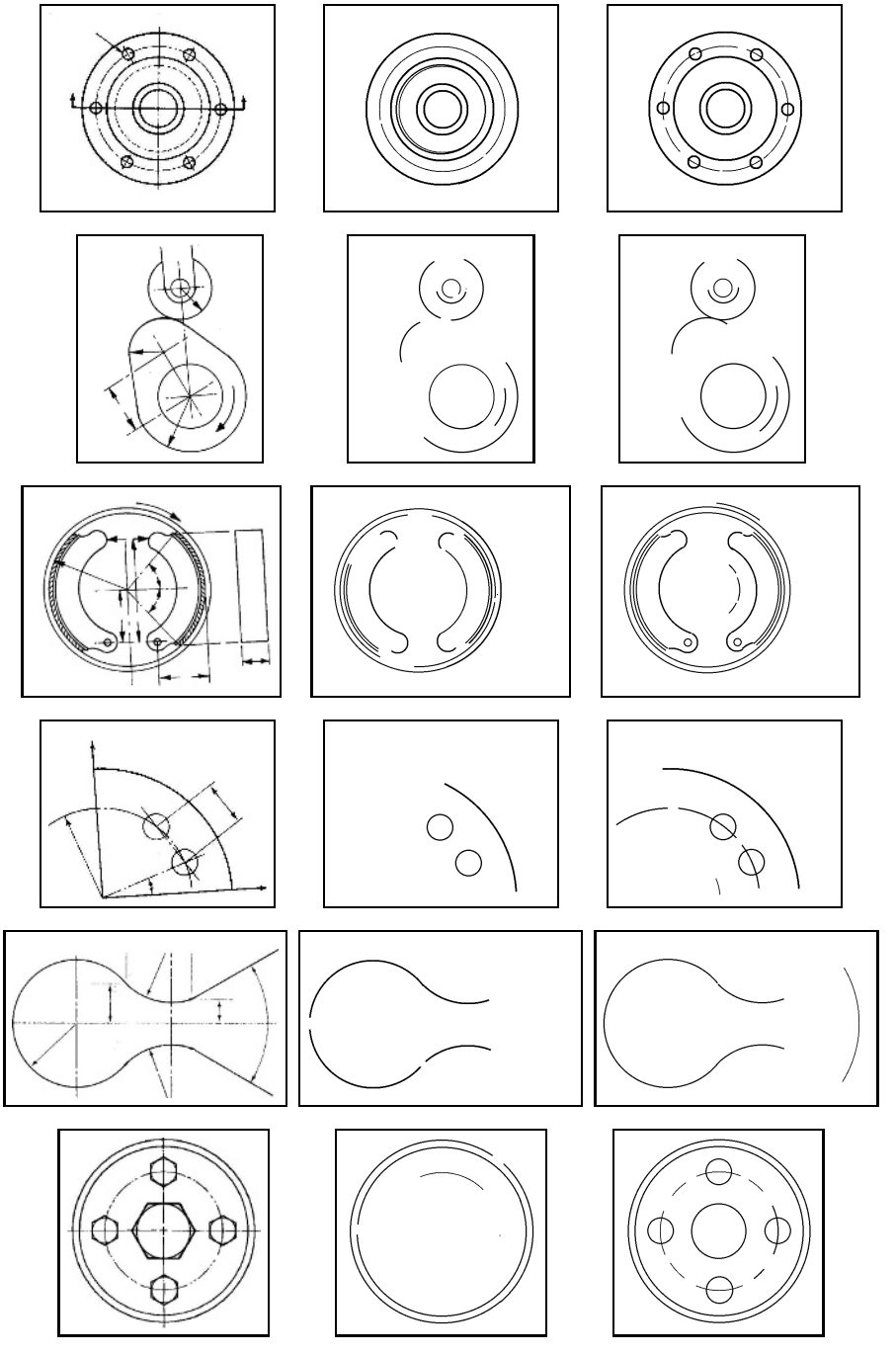


Fig. 2. GREC 2005 arc segmentation contest images (without noise), detected arcs and ground truth arcs

Note that we did not tune our method to the evaluation criterion used in the contest, which may be possible. For example, Hilaire [8] tuned the method specifically to the performance measure in the 2001 contest as he writes in the discussion of the method that "the penalty associated with *fragmentation* is probably the worst we may encounter" and that "our method should *first worry about thick, long arcs or full circles*", which are consequences of the evaluation scheme used. Note further that it is possible to obtain improvements in the score if one also outputs line segments, which our method did not do. It is possible that the the GREC 2007 arc segmentation contest will use a different evaluation scheme which would then be similar to the one discussed in [22].

6 Conclusion

We have described a simple and effective method for locating geometric primitives like lines and circular arcs in bi-level images of scanned documents. Except for a library for interval arithmetic and image file I/O, the algorithm itself is easily implemented in a few hundred lines of C++ code. Another property of the algorithm that simplifies its implementation and use is that the only parameters that the user needs to specify are bounds for the thickness of the lines, the desired accuracy of the primitives, and the minimum quality desired. The experimental results show a promising behavior of the algorithm on images of technical drawings.

Note that the development of the described algorithm is work in progress, although it relies on two well-understood and tested methods, which are the RAST framework of branch-and-bound search and the use of interval arithmetic for the computation of bounds of functions defined on regions of parameter space.

To our knowledge, the algorithm presented in this paper is the first practical, globally optimal algorithm for the detection of thick lines and thick circles under well defined error models in technical drawings; other methods described in the literature generally rely on suboptimal matching methods (like the Hough transform) and approximations during preprocessing (like polygonal approximations).

Existing systems for the detection of geometric primitives often include complex, hand-tuned rules related to grouping, continuity, and plausible parameter ranges. If we identify any such cases or constraints, we can incorporate them into the quality function used by our algorithm. The optimality guarantees of the algorithm still hold (i.e., the algorithm will find the optimal results under the improved quality function), and so do the main advantages of the algorithm (few parameters, strict separation of what is being optimized from the search algorithm itself).

Acknowledgments

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project IPeT (01 IW D03).

References

1. Elliman, D.: Tif2vec, an algorithm for arc segmentation in engineering drawings. In: Proc. 4th IAPR Workshop on Graphics Recognition, Springer LNCS 2390. (2001) 359–358
2. Wenyin, L.: Report of the arc segmentation contest. In: Proc. 5th IAPR Workshop on Graphics Recognition, Springer LNCS 3088, Barcelona, Spain (2003) 364–367
3. Wenyin, L., Zhai, J., Dori, D.: Extended summary of the arc segmentation contest. In: Proc. 4th IAPR Workshop on Graphics Recognition, Springer LNCS 2390. (2001) 343–349
4. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM **15** (1972) 11–15
5. Wenyin, L., Dori, D.: From raster to vectors: Extracting visual information from line drawings. Pattern Analysis and Applications **2** (1999) 10–21
6. Dori, D., Wenyin, L.: Sparse pixel vectorization: An algorithm and its performance evaluation. IEEE Trans. Pattern Analysis Machine Intelligence **21** (1999) 202–215
7. Dosch, P., Masini, G., Tombre, K.: Improving arc detection in graphics reognition. In: International Conference on Pattern Recongition. Volume 2., Barcelona, Spain (2000) 2243–2246
8. Hilaire, X.: RANVEC and the arc segmentation contest. In Blostein, D., Kwon, Y.B., eds.: Graphics recognition—Algorithms and applications. Volume 2390 of Lecture Notes in Computer Science. Springer-Verlag (2002) 359–364
9. Breuel, T.M.: Fast recognition using adaptive subdivisions of transformation space. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition. (1992) 445–451
10. Breuel, T.M.: On the use of interval arithmetic in geometric branch-and-bound algorithms. Pattern Recognition Letters **24** (2003) 1375–1384
11. Breuel, T.M.: Robust least square baseline finding using a branch and bound algorithm. In: Proceedings of the SPIE - The International Society for Optical Engineering. (2002) 20–27
12. Hagedoorn, M., Veltkamp, R.C.: Reliable and efficient pattern matching using an affine invariant metric. International Journal of Computer Vision **31** (1999) 203–225
13. Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J.: Comparing images using the Hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence **15** (1993) 850–63
14. Jurie, F.: Solution of the simultaneous pose and correspondence problem using Gaussian error model. Computer Vision and Image Understanding **73** (1999) 357–373
15. Mount, D.M., Netanyahu, N.S., Moigne, J.L.: Efficient algorithms for robust feature matching. Pattern Recognition **32** (1999) 17–38
16. Olson, C.F.: Locating geometric primitives by pruning the parameter space. Pattern Recognition **34** (2001) 1247–1256
17. Pavlidis, T.: Algorithms for Graphics and Image Processing. W. H. Freeman & Co., New York, NY (1983)
18. Forsyth, D.A., Ponce, J.: Computer Vision: A Modern Approach. Prentice Hall, Upper Saddle River, NJ (2003)

19. Breuel, T.M.: Finding lines under bounded error. Pattern Recognition **29** (1996) 167–178
20. Hickey, T.J., Ju, Q., van Emden, M.H.: Interval arithmetic: From principles to implementation. JACM **48** (2001) 1038–1068
21. Jaulin, L., Kieffer, M., Didrit, O., Walter, E.: Applied Interval Analysis. Springer Verlag, Berlin (2001)
22. Breuel, T.M.: Representations and metrics for off-line handwriting segmentation. In: 8th International Workshop on Frontiers in Handwriting Recognition. (2002)

Report on the Second Symbol Recognition Contest

Philippe Dosch[1] and Ernest Valveny[2]

[1] Université Nancy 2, Loria (UMR 7503 CNRS-INPL-INRIA-Nancy 2-UHP)
615, rue du jardin botanique,
B.P. 101, 54602 Villers-lès-Nancy Cedex, France
Philippe.Dosch@loria.fr
[2] Computer Vision Center - Computer Science Department (UAB)
Edifici O, Campus UAB, 08193 Bellaterra, Spain
ernest@cvc.uab.es

Abstract. Following the experience of the first edition of the international symbol recognition contest held during GREC'03 in Barcelona, a second edition has been organized during GREC'05. In this paper, first, we bring to mind the general principles of both contests before presenting more specifically the details of this last edition. In particular, we describe the dataset used in the contest, the methods that took part in it, and the analysis of the results obtained by the participants. We conclude with a synthesis of the contributions and lacks of these two editions, and some leads for the organization of a forthcoming contest.

1 Introduction

1.1 General Principles of Performance Evaluation

For many areas within pattern recognition and graphics recognition, performance evaluation has become a crucial field of research work [1, 2, 3, 4]. This effort has become necessary in order to be able to compare different methods on standard datasets using metrics agreed by the research community. In general, all these evaluation works rely on several components:

- A *dataset* containing a sufficient number of representative data for the field under evaluation. Data can be either real or synthetic, depending on the application domain. It should also include several kinds and levels of degradation and deformation.
- A *ground-truth* that represents the perfect labelling of test data and therefore, the results that the participants are expected to provide.
- A *metric* to measure the distance between the ground-truth and the results provided by the participant methods.
- A *protocol* that specifies how the organizers and the participants exchange all information (input data, results, etc.) concerning the competition.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 381–397, 2006.
© Springer-Verlag Berlin Heidelberg 2006

– A set of tools for the *analysis of results*. This analysis can be led from two different viewpoints: a data viewpoint, in order to determine how each kind of input data is recognized according to different methodological approaches, and a methodological viewpoint, in order to determine the strengths and weaknesses of every method for different kinds of data.

Some of these evaluation campaigns are designed to determine a sorting of the participant methods, based on a global performance measure computed after applying each method to the whole set of data. This approach is only possible in some domains where it is realistic to compute a global performance measure according to the characteristics of the data. However, whatever the performance measures are, we strongly believe that the main objective of an evaluation framework must be the scientific analysis of the results. This analysis must be intended to determine the different qualities expected for recognition methods: robustness, genericity, precision, computational efficiency. Usually, each of these qualities must be estimated with different performance measures computed over several sets of data.

These principles being defined, we would like to point out that complete and really useful performance evaluation requires a lot of tests, led under a large number of criteria. Usually, contests can only work with a limited dataset, which means that they can play an important role as relevant milestones in the evaluation process, but they must be completed with other efforts (like regular and large tests) to allow a good understanding of the recognition methods for a particular application domain.

1.2 Symbol Recognition Contests

For performance evaluation of symbol recognition, the general principles exposed above are obviously the same. Two evaluation events concerning symbol recognition have already been held before this edition. The first one was during the 15th International Conference on Pattern Pattern Recognition (ICPR'00) [5]. The symbol library for that contest consisted of 25 electrical symbols, which were scaled and degraded with a small amount of binary noise. Following this event, a second contest was held during the fifth International Workshop on Graphics Recognition (GREC'03) [6, 7], known as the first international contest on symbol recognition, as its characteristics were closer to those expected for such an event: several application domains, more symbols, more test images, different kinds and levels of degradation and noise, ... The contest organized in the context of GREC'05 and explained in this paper was the natural continuation of this one.

As there are many factors which can influence the performance of a symbol recognition method, the main goal of these contests were not to give a single performance measure for each method, but to provide a tool to compare various symbol recognition methods under several different criteria. From an evaluation viewpoint, the question consists of determining the performance of symbol recognition methods when working on various kinds of symbols, extracted from

diverse application domains, under several constraints, with different levels of noise and degradation.

In the following of this paper, first in section 2, we will briefly review the main features of the first edition of the contest during GREC'03. Then, in section 3 we will explain the second edition of the contest: the differences with the first edition, the dataset and the development of the contest. In section 4 we will give some details about the methods that took part in it and section 5 is devoted to the analysis of their results. As we have explained before, evaluation effort should not be limited to some specific milestones but it should have a continuity over time. In section 6 we will explain the main goals of the project Épeires, a project currently under development intended to provide a stable framework for evaluation of symbol recognition. Finally, in section 7 we draw the main conclusions of the contest and some hints about future work.

2 First Edition of the Contest

To have a complete overview of the first edition of the contest, we advise to refer to [6]. Here, we will only remind the main features of this first edition:

- 50 different model symbols were used in the contest, from two different application domains (electronics and architecture) and composed of lines and arcs.
- Only segmented images of symbols were used.
- Scalability: 3 sets of symbols each with 5, 20 and 50 symbols were defined to test the robustness to scalability.
- Generation of images with rotation and scaling in order to test the invariance to geometric transformations.
- Generation of images with 9 models of binary degradations, following the model of degradation of Kanungo [8], see figure 1.
- Generation of images with 3 models of vectorial shape deformations, see figure 2.
- Generation of tests with combination of these transformations.
- Whenever possible, pixel-based and vector-based test images were provided to participants.
- 5 participants took part in the contest.
- More than 7000 test images were synthetically generated and organized in more than 70 independent tests.

The analysis of results was led according to the scientific criteria exposed above, i.e, trying to study how the performance of methods was degraded when applying transformations and noise to the original images. The main conclusions were:

- With respect to the different tests supplied, most of participant methods had very good recognition rates, usually between 95% and 100%, when dealing with 5 or 20 model symbols and few degradation and deformation.

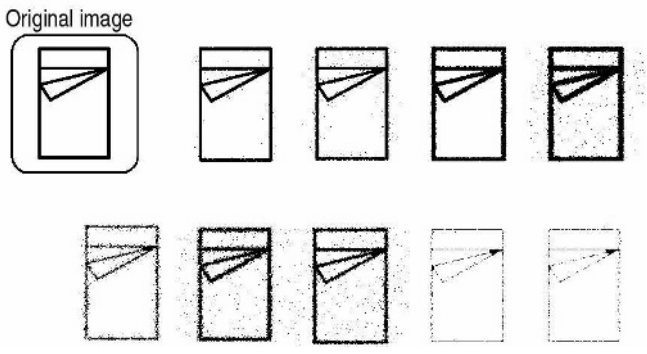


Fig. 1. The nine models of degradation used for the first edition of the contest

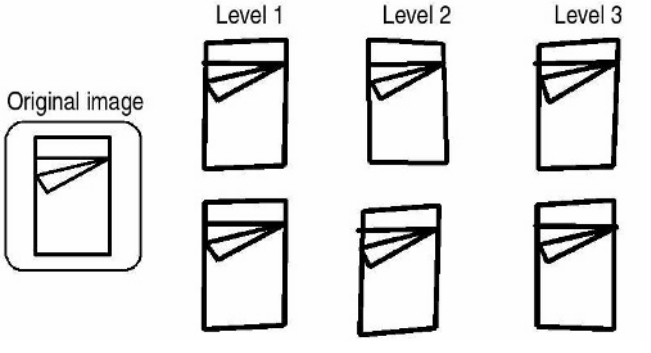


Fig. 2. The three models of deformation used for the first edition of the contest

- In general, the performance decreases with the number of symbols, for all kinds of tests.
- Methods are not fully invariant to rotation and scaling, even whenever they claim the contrary.
- Robustness to degradation: performance degrades significantly with heavy noise or when connectivity of lines is lost.
- Robustness to vectorial shape deformation: The performance decreases with the deformation, but not too much.

As a conclusion of the first edition, and as the results were quite good, we planned to increase the complexity of the data used in a future contest in these directions:

- Add more symbol models, in order to evaluate the scalability of the recognition methods.
- Add new models of noise (heavy noise), in order to evaluate more accurately their robustness.
- Define tests with non-segmented symbols, in order to evaluate the ability to localize, segment and recognize these symbols in real drawings.

3 Second Edition of the Contest

3.1 General Principles

Following the conclusions of the first edition of the contest, we have tried to set up a new edition including the new features pointed out at the end of previous section. We have been able to achieve only two of those goals. In this new edition, we have included more symbols and more models of noise. However, we have not succeeded in the inclusion of non-segmented symbols. In fact, a lot of effort is required in order to set up the evaluation of the localization and segmentation of symbols. Among the issues to be addressed we can remark the following:

- To build a dataset providing a large and enough number of real images of different domains, such as architecture drawings, electronic maps, etc. As these data are often private, it is difficult to get a dataset representative enough for such a contest.
- The definition of metrics allowing the comparison of ground-truth with the results provided by the participants. The ground-truthing itself requires a lot of time, and has to respect a very well defined methodology to be fully exploitable. In particular, we believe that the definition of the ground-truth have to include the creation itself, but also the validation by different people, in order to ensure that the ground-truth will be agreed by everyone. Incidentally, ground-truthing is a very time-consuming task, as test data have to be handled by many people to become fully exploitable.
- The design of an environment allowing the automatic processing of the results provided by participants, in order to analyze them. Whereas the evaluation of symbol recognition methods requires a reasonable framework of evaluation, the evaluation of localization and segmentation methods requires a significant bigger effort as much data have to be managed.

In this context, we are working in parallel on a project, funded by the French government, which aims at providing such an environment for the scientific community. This project, called Épeires, is briefly presented in section 6. Once this project has been fully developed, it will allow to easily define tests for symbol localization available to everybody.

In conclusion, for this edition of the contest, we have given up the idea of including non-segmented images and we have only defined some tests with segmented images, very similar to those proposed during the first edition, but with the following remarkable differences:

- The set of symbols has grown from 50 to 150 different symbols, allowing the definition of tests useful for the evaluation of the scalability of recognition methods.
- Four new degradation models have been added, allowing the generation of more noisy synthetic data. These degradation models are further explained in the next section.
- Tests have only included bitmap images for this edition. Vectorial images have not been taken into account as the selected models of degradation do not allow for a good vectorization of images.

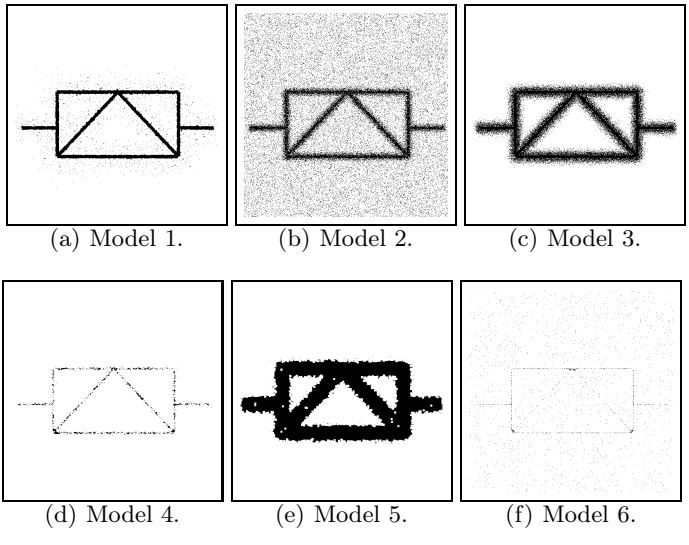
3.2 The Symbol Database

As for the first edition, two application domains have been mainly used, architecture and electronics. We have used 150 different symbols, some of them with similar shapes, grouped in four sets containing respectively 25, 50, 100 and 150 symbols, in order to evaluate the scalability of recognition methods. As we have previously said, we have only considered in this edition presegmented symbols, *i.e.* images containing one instance of one symbol, and only bitmap format.

Several transformations, global transformation and noise, have been applied on these ideal models, in order to evaluate the robustness of the recognition methods to such transformations. Global transformations include rotated and scaled images, whereas noisy images have been generated using the well-known Kanungo's method [8]. We remind that the initial purpose of this method is to modelize the noise produced by operations like printing, photocopying, or scanning. The method is formal and validated for its correctness, but the determination of the set of parameters used for the contest is more empirical.

In the first edition of the contest, we have tried to reproduce a set of degradations reproducing some realistic artifacts as those mentioned above. But for this edition, six degradation models have been defined, aiming at constituting a set of what we could call "torture models" rather than some realistic degradation models, as this issue was relatively well addressed during the first edition. This way, we can test the robustness of recognition models under very extreme conditions. Some samples of images generated with these models are shown in figure 3.

But at the same time, we have to be very careful on the conclusions we can draw on the related results. We are obviously aware that it may be dangerous to



(a) Model 1. (b) Model 2. (c) Model 3.

(d) Model 4. (e) Model 5. (f) Model 6.

Fig. 3. The six models of degradation used for the second edition of the contest

rely the performance evaluation on some too noisy synthetic data, probably too different of real images. So, if this dataset can be used to proof the robustness of the participant methods under extreme conditions, we have also to be careful on the meaning of the evaluation, especially on the capacity of these methods to work on real data.

3.3 The Contest

All information related to the second edition of the contest is available at `http://symbcontestgrec05.loria.fr/`. Most of this information, especially those points related to the formats and protocols, is the same as for the first edition. The report of the first edition [6] and the related Web site at `http://www.cvc.uab.es/grec2003/SymRecContest/index.htm` provide a good description of the contest environment.

As for the first edition, independent tests have been designed with respect to different categories (concerning the number of symbols, the kind of noise, etc.) so that each participant, according to the specificities of its method, could choose the tests he wanted to run. As some methods require training data in order to work properly during the contest, the models of all the symbols and some sample tests were made available for all participants before the contest, with the associated ground-truth. The tests provided to the participants in the contest were similar, but different from the sample tests.

For the second edition, tests have been designed according to the following categories:

- *Scalability*, with 4 categories of tests involving an increasing number of symbols: 25, 50, 100 and 150. This category is intended to evaluate the capacity of the recognition method to discriminate symbols as the number of models increases.
- *Degradation models*, with the 6 models presented in section 3.2. This category is intended to evaluate the robustness of the methods when symbols are degraded under several conditions.
- *Transformations*, by considering rotation and scaling, either alone or together. In addition, a category without any transformation has been defined. As for degradation models, this category is intended to evaluate the robustness of recognition methods under geometric transformations.

All these categories have been combined resulting in 96 different tests, with a total number of 6000 test images.

4 Participants

In this edition, four participants and their method took part in the contest. Two of these participants have a paper describing their method in the current LNCS volume. In this section, we only give a brief overview of the most relevant features of the participant methods, as provided by their authors.

4.1 Jing Zhang, City University, Hong Kong

The recognition method is a statistical, pixel-based method. The method used is very similar to Su Yang's. The symbol descriptor we used is referred to as Structural Feature Histogram Matrix (SFHM), which is an improvement of Yang's SIHA in two aspects:

1. SFHM computes length ratios and angles via a symbol's centroid;
2. SFHM integrates the information of length ratios and angles.

4.2 Min Feng, City University, Hong Kong

The recognition method is a statistical and pixel-based method. The similarity is calculated by matching the point sets extracted from the symbols. The assumption in the method is many to many correspondence, which reduces the time complexity into O(n2). However, the new similarity function is not invariant to rotation. To recognize rotated symbols, we compute their angular distributions and align them by their orientations. The whole recognition procedure consists of three steps: image compression, denoising and recognition. Firstly, the input images are compressed in order to cut down the number of foreground points. After compressed, each pixel indicates the density of the foreground points in the original image. Secondly, a novel denoising technique is utilized to remove the noises from the compressed images. Finally, the above similarity function is used to compute the similarity between each pair of preprocessed test symbol and model symbol, and then for each test symbol the best matched symbol model is outputted.

4.3 Wan Zhang, City University, Hong Kong

The method is a statistical approach, where a symbol is represented by a 2D joint density estimated from a set of points sampled from the skeleton of the symbol. Matching two symbols is then equivalent to determining whether the two symbols have a similar probability distribution or not. In other words, if the points on the test symbol fit the density of the symbol model well, we can determine that the test symbol is similar to the model one. By adopting the Kullback-Leibler (KL) divergence as a distance of the two distribution densities, the similarity of the two symbols can be measured. In the first preprocessing module, a freeware (Ras2Vec) is selected to finish the vectorization processing of the binary images and obtain the skeletons of images. Furthermore if necessary, a few preprocessing techniques will be applied to reduce the noise and help to improve the robustness. The method is independent of the position of the symbol, and easy to be extended for rotation-invariance and scale-invariance.

4.4 Andyardja Weliamto, Nanyang Technological University, Singapore

This recognition system is based on the statistical approach. It assumes that at the end of the preprocessing step we have single pixel thin line. The system consists of several steps:

1. Preprocessing/filtering: adaptive noise preprocessing using morphological, convolution and thresholding for different noise models (noise model classification).
2. Feature selection/feature vector composition based on Fisher Discriminant Analysis.
3. Classification based on the k-Nearest Neighbor with Mahalanobis distance.

The problem of the system in the contest was that we did not have enough time to verify the linearity of preprocessing image among different noise models. We also needed to test some parameters that deal with the training system. The constraints of the system are: first, it is difficult to make the preprocessing of the image linear among different noise models without experimentally testing and second, the feature vector should be unique with higher discriminant factor. That is why the recognition rate drop since the preprocessing step fails. Incorporating better adaptive noise reduction preprocessing of images increases the recognition rate of the system by 10%. Another problem was that some symbols have similar radial feature. Therefore, we need to introduce some new features based on the angular feature and as a result of it the recognition rate increases by 6%.

5 Analysis of Results

5.1 Introduction

As a preamble of this section, we want to point out that this analysis is related to the dataset defined, which contains only synthetic data, degraded using some set of parameters for Kanungo's method, as explained in section 3.2. Even if some of the generated data seems close to real data, as those represented in various technical documents, other images are rather far from real or realistic data. The purpose of this kind of contest is obviously to determine what methods work on real data, as this is the typical way they are used in real applications. But as building a set of real data, with a representative and sufficient number of images, is complex for several reasons (availability, rights, work force), the current edition is partially based on exaggerated noisy data. This is a more practical way to proof the robustness of recognition methods, but it also implies that we have to keep aware of these evaluation conditions, and therefore, be careful on the interpretation of results.

Moreover, in this edition, some of the tests have been designed with a low number of images for some categories, as the participants had to run their method on all tests the day before GREC. It leads sometimes to strange recognition rates, as tests did not contain a significative enough number of images, and maybe some images were "easier" to recognize in some of the sets.

These two constraints, the use of synthetic data and the restricted number of images involved, lead of course to some limitations for the determination of the more generic and robust method/approach in symbol recognition.

Another important remark we also want to point out is that participant methods should not integrate *a priori* knowledge about degradation or transformation

models, if the goal has to be the evaluation of the genericity. From a scientific viewpoint, the most important task is to evaluate the core of symbol recognition methods, and not frameworks integrating pre or post processing steps dedicated to the degradation models. Even if some labels are provided for each proposed test, either training or final, they are essentially tips to easily determine if a recognition method is adapted to a test, not to allow an adaptation of the method to the test. As this principle was not explicitly defined for this edition, it may be another limitation, as some methods included specific preprocessing depending on the kind of noise detected in the test.

In the following, we will discuss the results provided by the participants, from several viewpoints, taking into account the above stated limitations and bearing in mind that our main goal is not to give an absolute winner, but to show the robustness of the methods to the different evaluation criteria.

5.2 Clean Images

The first degradation model was designed to simulate some clean images, *a priori* very close from real ones. The results obtained for the tests using this model are presented in figure 4 (see the first column). Recognition rates correspond to the average rate for all participant methods. For comparison, figure 5 shows the results obtained by the best participant method, those of Feng Min, with an average recognition rate of 94.88% (see the first column too). The results are very good, for all methods. Even if the recognition rate decreases a bit when the number of symbols increases, the average recognition rate is equal to 98% when dealing with 150 different symbols. So we can conclude that symbol recognition

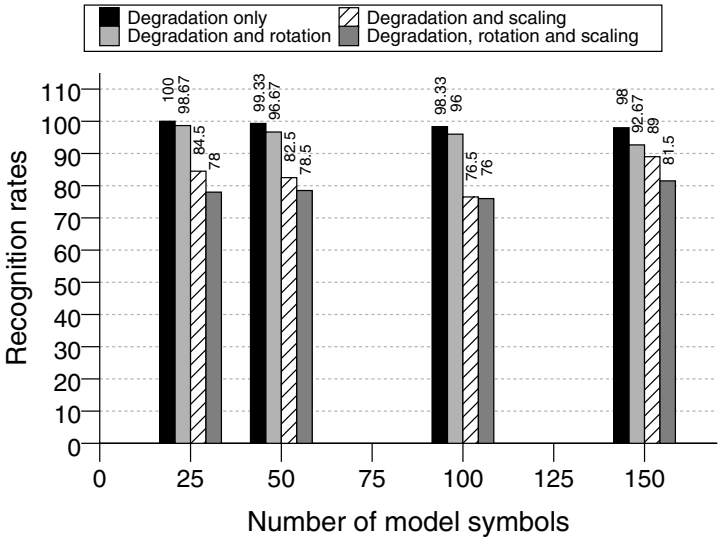


Fig. 4. Synthetic chart showing the average recognition rate obtained by all participant methods for the first degradation model with all combinations of rotation and scaling

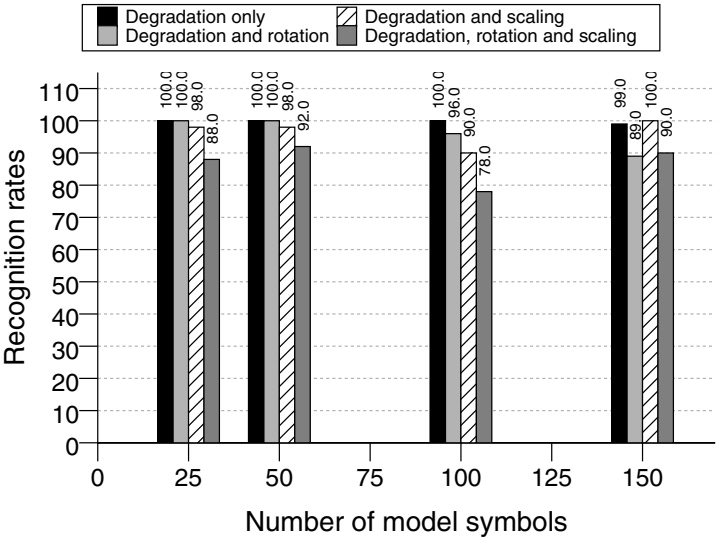


Fig. 5. Results obtained by the best participant method, those of Feng Min, for the first degradation model

is quite mature with these contest conditions, close to ideal real ones. The next step in order to evaluate the scalability of the methods under these conditions is probably to propose tests with a very larger number of symbols, maybe 1000.

5.3 Clean Images with Transformations

Still working with the first degradation model, close in our opinion to ideal real images, tests have been defined with rotation, scaling, and a combination of both transformations. The corresponding results are also presented on figure 4 and 5 (second, third and fourth columns). When dealing only with rotation, the recognition rate decreases and this tendency is accentuated when the number of symbols increases. The average recognition ability is almost reduced by approximately 5% when dealing with 150 symbols with respect to the same tests without any transformation. Similar remarks can be done with the tests related to scaling only and the combination of both transformations, leading respectively to approximate reductions of 9% and 16%.

The tendencies shown on the synthetic chart (figure 4), presenting only average recognition rates, are similar to results obtained by each participant. For Feng Min (figure 5), we can however see that the results obtained with the test dealing with scaling only and 150 model symbols are better than the others related to this scalability category. As this test contains only 50 test images, which is probably not representative enough with respect to the 150 model symbols, it is however difficult to formally interpret this result.

But from a general viewpoint, transformations clearly still impact recognition quality.

5.4 Scalability

Testing scalability with respect to the number of considered model symbols is one of the main objectives of the symbol recognition contests. For this edition, figure 6 presents a synthetic chart of the average results obtained for all degradation models, according to the number of model symbols. The performance clearly decrease when the number of symbols increases, with some variations according to the kind of tests. For tests including degraded images without any transforma-

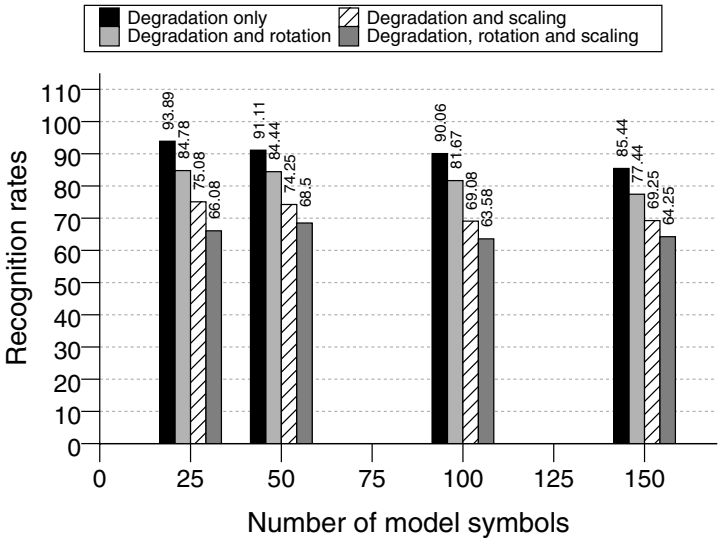


Fig. 6. Synthetic chart for all degradation models

tion, the loss of recognition is about 8.5% when the number of symbols evolves from 25 to 150. This loss decreases when transformations are added. It is about 7.3% when rotation is added, about 5.8% when scaling is added and only about 1.8% when both of these transformations are added. This is a bit surprising, as we intuitively expect that the more constraints are added, the more performance decreases.

In general, the decrease seems to be linear with respect to the number of symbols involved. A larger number of model symbols has to be considered in further events related to performance evaluation to allow a more detailed analysis of scalability impact.

5.5 Participants Method and Degradation Models

The last chart, presented in figure 7, shows the average recognition rates obtained by each participant for each degradation model. The recognition rates obtained for each degradation model is rather different from one participant to another. This fact shows that, according to the recognition approach, methods are more

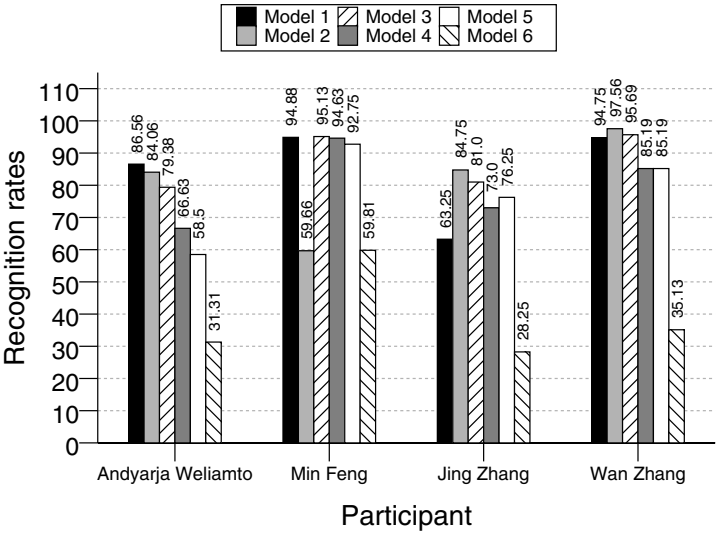


Fig. 7. Synthetic chart for each participant with respect to the degradation models

or less sensitive to the kind of degradation. It reminds the importance of using several degradation models for this kind of performance evaluation. It would be interesting to have more details on each participant method to have a better understanding of this behavior. The model 6 appears to be the more difficult to recognize in general. This is probably because the corresponding images are very degraded, with loss of connectivity and a global noise with pixel density close to that of the symbol itself. The only method having a recognition rate higher than 50% for this model is that of Min Feng, with a recognition rate equal to 59.81%.

Min Feng is also the global winner of the contest, with an average recognition rate of 83.33% for all the proposed tests, as shown in table 1. However, as previously stated, the main purpose of this contest is not to determine a winner, but rather to study the evolution of the recognition rates according to the test characteristics. But it appears that 83.33% is a good overall recognition rate considering the proposed tests, most of them designed to be "torture tests", *i.e.* exaggerated noisy data proposed to assess the robustness of the participant methods.

Table 1. Overall results for each participant

Participant	**Average recognition rate**
Andyardja Weliamto	70.28%
Min Feng	83.33%
Jing Zhang	67.65%
Wan Zhang	82.82%

5.6 Synthesis

Following the experience of the first edition of the symbol recognition contest, this second edition has been organized in order to propose some difficult tests on segmented symbols. It appears that the recognition rates are quite good with respect to some of the degradation models proposed. In particular, in conditions close to the ideal ones, an average recognition rate of 98% has been reached by the participant methods. As a conclusion, one can say that symbol recognition, in the conditions defined for the contest, is quite mature, even if participants methods are not fully invariant to transformations like rotation and scaling.

As expected, performances generally decrease when the number of model symbols increase and when transformations are added. Therefore, more models have to be proposed to accurately measure the scalability with really large sets of symbols. And more data, representative of other application domains, have to be supplied too, in order to evaluate the robustness of the participant methods to different domains and kinds of symbols.

Now, we think that the next important challenge is to organize tests about symbol localization, that is to say, symbol recognition on images including several instances of different symbols in their real context, connected to other lines or elements of a drawing. Tests about symbol recognition are still interesting, but only in some particular aspects, such as, for example, scalability with a large number of model symbols.

We plan to define forthcoming tests about symbol localization thanks to the Épeires project presented in the next section.

6 The Épeires Project

The Épeires Project[1] is funded by the French Ministry of Research in the context of the Techno-Vision Campaign[2]. Its purpose is the construction of a complete environment providing tools and resources for performance evaluation of symbol recognition and localization. The aim is to estimate their capacity to recognize and to localize symbols in a generic way, according to various criteria: application domain, modelization, number of symbols involved, document quality, etc. The consortium is currently composed by 6 laboratories (City University of Hong Kong, CVC Barcelona/Spain, LI Tours/France, L3I La Rochelle/France, LORIA Nancy/France and PSI Rouen/France) and also 2 French companies (France Télécom R&D and Algo'Tech Informatique). This environment is intended to be used by the whole scientific community.

Document analysis generally deals with two main kinds of symbols: structured symbols and logos. In the Épeires Project, we intend to consider symbol recognition as a whole, without making any particular distinction between them. Participant methods should be subsequently tested on both kinds of symbols. The Épeires Project is organized along 3 main directions:

[1] http://www.epeires.org/

[2] http://www.recherche.gouv.fr/appel/2004/technovision.htm

- *Development of a database of test images*, in order to get a large variety and a large number of test data, possibly free of rights. Images will be proposed in clean and degraded versions, to test the robustness of the recognition methods. A ground-truth will be associated with each image, using a collaborative software (currently under development) connected to the information system of the project.
- *Design of metrics and protocols* specifying how the results will be analyzed.
- *Performance evaluation of the methods supplied by the participants.* It will determine the methods providing the best recognition and/or localization rates on the documents of the test database. It will also be the opportunity to measure the strengths and weaknesses of the methods. As for the contests, the goal will not only be to determine the most reliable chains of applications from a synthetic viewpoint, but also to understand the influence of the different approaches on the quality of the results.

As a result, it is planned to provide at least 1000 model symbols and 100000 test images. We hope we will provide to the community a great tool for performance evaluation of symbol recognition and localization.

7 Conclusion and Next Steps

As a conclusion of the two contests on symbol recognition organized during GREC'03 and GREC'05, we would like to point out the following issues:

- More information is needed from the participants to better understand the recognition rates. We expect that they give a more detailed description of their methods, and they give more feedback on their results. We plan to provide tools to assist these descriptions and discussions.
- We have to provide facilities allowing to spread and analyze the results of evaluation campaigns, for the further contests as well as for any campaign related to symbol recognition and localization. We hope that the Épeires project will supply such facilities very soon.
- More data, free of use, are still required, as performance evaluation cannot be fully suitable without a large number of heterogeneous data. It is a call for the community, as we all need these data to make evaluations on our methods.
- No new degradation models based on Kanungo method are needed. After these two contests, we have defined 15 different models, from more realistic noise to "torture models", and we think it is enough. It is more interesting to support new kind of noises, like scratches, or to mix the existing models in blind tests.
- Next campaigns must include blind tests in order to ensure that participant methods are not adapted to the particular data of the contest. We would like to be sure that participants address the good goal: design generic symbol recognition methods, working with all kind of (noisy) symbols, and not only those provided in the context of these contests.

- Campaigns of evaluation must be led more regularly than every 2 years. If we fully want to integrate performance evaluation as a main part of each research on symbol recognition method, we need a stable environment for evaluation events with more heterogeneous data.
- And of course symbol localization must be addressed as it is currently on of the main challenging issues for the symbol recognition community.

For the major part of these remarks, we hope that the ÉPEIRES project will be able to provide such a complete framework to the community.

Acknowledgment

The authors would like acknowledge the participants for their participation to the contest and for their contribution to this article. They also would like to acknowledge the French Ministry of Research for the funding of the ÉPEIRES project as a part of the Techno-Vision campaign. This work has also been partially supported by the Spanish project CICYT TIC2003-09291.

References

1. Kong, B., Phillips, I.T., Haralick, R.M., Prasad, A., Kasturi, R.: A benchmark: Performance evaluation of dashed-line detection algorithms. In Kasturi, R., Tombre, K., eds.: Graphics Recognition: Methods and Applications, Selected Papers from First International Workshop on Graphics Recognition, GREC'95. Springer, Berlin (1996) 270–285 Volume 1072 of Lecture Notes in Computer Science.
2. Chhabra, A., Phillips, I.: The second international graphics recognition contest - raster to vector conversion: A report. In Tombre, K., Chhabra, A., eds.: Graphics Recognition: Algorithms and Systems, Selected Papers from Second International Workshop on Graphics Recognition, GREC'97. Springer, Berlin (1998) 390–410 Volume 1389 of Lecture Notes in Computer Science.
3. Chhabra, A., Philips, I.: Performance evaluation of line drawing recognition systems. In: Proceedings of 15th. International Conference on Pattern Recognition. Volume 4. (2000) 864–869 Barcelona, Spain.
4. Wenyin, L., Zhai, J., Dori, D.: Extended summary of the arc segmentation contest. In Blostein, D., Kwon, Y., eds.: Graphics Recognition: Algorithms and Applications, Selected Papers from Fourth International Workshop on Graphics Recognition, GREC'01. Springer, Berlin (2002) 343–349 Volume 2390 of Lecture Notes in Computer Science.
5. Aksoy, S., Ye, M., Schauf, M., Song, M., Wang, Y., Haralick, R., Parker, J., Pivovarov, J., Royko, D., Sun, C., Farneboock, G.: Algorithm performance contest. In: Proceedings of 15th. International Conference on Pattern Recognition. Volume 4. (2000) 870–876 Barcelona, Spain.
6. Valveny, E., Dosch, P.: Symbol recognition contest: a synthesis. In Lladós, J., Kwon, Y.B., eds.: Graphics Recognition: Recent Advances and Perspectives – Selected papers from GREC'03. Volume 3088 of Lecture Notes in Computer Science. Springer-Verlag (2004) 368–385

7. Valveny, E., Dosch, P.: Performance Evaluation of Symbol Recognition. In: Proceedings of the 6th IAPR International Workshop on Document Analysis Systems, Florence, (Italy). Volume 3163 of Lecture Notes in Computer Science. (2004) 354–365
8. Kanungo, T., Haralick, R.M., Baird, H.S., Stuetzle, W., Madigan, D.: Document Degradation Models: Parameter Estimation and Model Validation. In: Proceedings of IAPR Workshop on Machine Vision Applications, Kawasaki (Japan). (1994) 552–557

Symbol Recognition Using Bipartite Transformation Distance and Angular Distribution Alignment

Feng Min, Wan Zhang, and Liu Wenyin

Department of Computer Science,
City University of Hong Kong, China
{emmwcity, wanzhang, csliuwy}@cityu.edu.hk

Abstract. In this paper, we present an integrated system for symbol recognition. The whole recognition procedure consists of image compression, denoising and recognition. We present a pixel-based method to calculate similarity between two symbols using the *bipartite transformation distance* after they are aligned by their angular distributions. The proposed method can overcome some shortcomings of other pixel-level methods. We also propose a new denoising technique in our system to improve the recognition precision and efficiency. Evaluation results on test sets provided by the 2nd IAPR contest on symbol recognition show good performance of the system in recognizing symbols with degradation and affine transformation.

1 Introduction

Symbol recognition is an important problem in many application fields, such as recognition of engineering drawings [4], circuit diagrams [5], and handwritten characters [8]. As there are many factors which can influence the performance of a recognition approach, such as affine transformations, distortion and degradation, it is necessary to provide a universal system to preprocess and recognize symbols under all kinds of circumstance.

In this paper we present a pixel-based approach to symbol recognition, which is a brute-force method. The *bipartite transformation distance* is proposed in this paper to calculate similarity in our method. However, it is not invariant to rotation transformation. To recognize symbols with rotation transformation, we compute their angular distributions and align them by their orientations.

We also describe our symbol recognition system used in the GREC 2005 contest. Our system utilizes certain techniques to predigest the input symbol to achieve higher recognition accuracy, including image compression and edge extraction. Various kinds of noise may exist in the input symbols, which severely affect similarity computing. Hence, we also propose a denoising technique to remove noise from the symbols. Experiments show that the integrated system can recognize the symbols effectively and efficiently.

The structure of this paper is described as follows. Related work is reviewed in Section 2. Section 3 presents the proposed methodology. In Section 4, we

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 398–407, 2006.
© Springer-Verlag Berlin Heidelberg 2006

illustrate the steps in our symbol recognition system. Evaluation results are briefly presented in Section 5. Finally, we draw conclusions and discuss future work in Section 6.

2 Related Work

One of the most popular classes of recognition methods is pixel-based method. The representative approaches are Fourier descriptors [9], moment invariants [3], ring projection [8] and shape contexts [1]. These descriptors are invariant to affine transformation but each of them has the shortcomings mentioned in [1], [7]. Fourier descripter only describes the external contour and ignores the internal contour and is difficult to be extracted from a real image. Moment invariants are rotation, scale and translation invariant, however, they are unable to provide a detailed profile about a symbol's structure. Hence, their discrimination power is limited. Shape context is a robust descriptor, but its rotation invariance depends on the tangent at every pixel, which is sensitive to deformation.

Intuitively shape matching can be done by matching the point sets extracted from the images. Let A and B be point sets of sizes n and m extracted from two shapes accordingly. A frequently used dissimilarity measure is Hausdoff distance [6]. It is defined as infimum of the distances of the points in A to B and the points in B to A. If the two point sets have the same size, i.e., $m = n$, *minimum weight (distance) matching* can be applied [6]. A *minimum weight matching* minimizes the sum of the weights of the edges of the matching. The difference between our approach and the approaches mentioned in [6] is that the assumption in our method is many to many correspondence among the points and different distance functions are implemented.

3 Symbol Similarity

3.1 Normalizing Symbol

Assume that a 2D symbol is represented as a binary image. Its grey-scale image, $S(x, y)$, can be discretized into binary values as follows:

$$S(x,y) = \begin{cases} 1 & \text{if (x,y) is foreground pixel of the symbol} \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

We can derive the centroid of the symbol, as denoted by $cen(x_0, y_0)$, which is the geometric center of the symbol:

$$\begin{aligned} x_0 &= \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x * S(x,y)\,\mathrm{d}x\,\mathrm{d}y}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S(x,y)\,\mathrm{d}x\,\mathrm{d}y} \\ y_0 &= \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} y * S(x,y)\,\mathrm{d}x\,\mathrm{d}y}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S(x,y)\,\mathrm{d}x\,\mathrm{d}y}, \end{aligned} \tag{2}$$

and subsequently, translate the origin of our reference coordinate to this centroid. Next, we define

$$R_{avg} = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} r(x,y) * S(x,y)\,\mathrm{d}x\,\mathrm{d}y}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S(x,y)\,\mathrm{d}x\,\mathrm{d}y}, \tag{3}$$

where $r(x,y)$ is the distance between (x,y) and centroid. In other words, R_{avg} is the average radius of all the points in the symbol. To normalize the symbol, we scale it into fixed size by the following formula

$$S'(x,y) = S(x * R_{avg}, y * R_{avg}), \tag{4}$$

where $S'(x,y)$ represents the normalized symbol. After normalized, the symbols matching is invariant to scaling.

3.2 Similarity Between Two Symbols

Generally the similarity between two symbols is inverse to the difference between them. In this paper, the difference between two symbols is referred to as *bipartite transformation distance*, which is something like the dissimilarity function in shape matching [2].

Firstly, we define the distance from point $p(x,y)$ to symbol S as the distance between $p(x,y)$ and the closest point $p_s(x_s,y_s)$ in S,

$$dis(p,S) = \min_{S(p_s)\neq 0} distance(p,p_s). \tag{5}$$

Given two symbols S_1, S_2, the cost of transforming S_1 into S_2 is defined as follows,

$$cost(S_1,S_2) = \int_R dis(p,S_2)^2 * S_1(p)\,\mathrm{d}p, \tag{6}$$

where R is the range of the coordinate of S_1. In other words, the cost is equivalent to the amount of work required to transform one symbol into the other. The transformation approach is to move each point in S_1 to S_2 through the shortest route, and therefore the cost is equivalent to the sum of the distances between each point pair. We normalize the cost by the area of S_1 and derive the *unipartite transformation distance* as follows,

$$unidis(S_1,S_2) = \frac{cost(S_1,S_2)}{\int_R S_1(p)\,\mathrm{d}p}, \tag{7}$$

where $unidis(S_1,S_2)$ is the *unipartite transformation distance* from S_1 to S_2 and the denominator is the number of foreground pixels in S_1.

Unipartite transformation distance cannot be considered directly as the difference between symbols because of partial matching. For example, if S_1 is similar to part of S_2, the *unipartite transformation distance* will be very small, but in fact, the two symbols are probably very different and the *reverse transformation distance* is quite large.

Hence we adopt *bipartite transformation distance* to estimate the difference between S_1 and S_2 instead, which is defined as,

$$bidis(S_1, S_2) = \frac{cost(S_1, S_2) + cost(S_2, S_1)}{\int_R S_1(p)\,\mathrm{d}p + \int_R S_2(p)\,\mathrm{d}p}. \tag{8}$$

The similarity between S_1 and S_2 is defined as follows,

$$sim(S_1, S_2) = \frac{1}{1 + bidis(S_1, S_2)}. \tag{9}$$

$Sim(S_1, S_2)$ is between 0 and 1. When two symbols are the same, the *bipartite transformation distance* is 0 and the similarity is 1.

3.3 Symbols with Rotation

If symbols are rotated, we cannot use the above method directly because it is not rotation-invariant. A simple way is to try rotating one of the symbols several times and find the maximum similarity between them. Since calculating similarity after each rotation costs too much, it is necessary to design a low-cost measure to normalize the orientations of the symbols. We can also apply the gradient-descent algorithm in order to improve the recognition accuracy.

We transform the original reference Cartesian coordinate system into the polar coordinate system based on the following relations:

$$\begin{cases} x = \gamma \cos\theta \\ y = \gamma \sin\theta \end{cases}. \tag{10}$$

Hence,

$$S(x, y) = S(\gamma \cos\theta, \gamma \sin\theta),$$

where $\gamma \in [0, \infty)$, $\theta \in [0, 2\pi)$.

For any fixed $\theta \in [0, 2\pi)$, we then compute the following:

$$g(\theta) = \frac{\int_0^\infty r * S(\gamma \cos\theta, \gamma \sin\theta)\,\mathrm{d}\gamma}{\int_0^\infty S(\gamma \cos\theta, \gamma \sin\theta)\,\mathrm{d}\gamma}. \tag{11}$$

The resulting $g(\theta)$, which can be viewed as a 1-D symbol that is directly transformed from the original 2-D symbol, shows the distribution of the foregound pixels in symbol along angle θ. We assume that $g(\theta)$ is a periodic function and its period is 2π. If the symbol is rotated by angle φ, its distribution function can be easily represented as $g(\theta - \varphi)$. We do not need to recompute the function $g(\theta)$ after each rotation and therefore this feature can be utilized to judge whether two symbols are in the same orientation.

Given two symbols and their distribution functions, we define the difference between the two distributions:

$$diff\,(\varphi) = \int_0^{2\pi} [g_1(\theta - \varphi) - g_2(\theta)]^2\,\mathrm{d}\theta, \tag{12}$$

where φ is the angle by which the symbol S_1 is rotated. Our idea is that one of the two symbols must be rotated to make their difference minimum. The rotated angle is denoted as φ_{min}. Our algorithm to find φ_{min} is shown as follows:

Step 1. Input S_1 and S_2, initialize $run\_length = 2 * \pi/60$, $\varphi = 0.0$, $diff_{min} = \infty$, $\varphi_{min} = 0.0$;
Step 2. Compute the distribution g_1 and g_2 using Eq(11). In practice, g_1 and g_2 are represented by discrete arrays;
Step 3. Compute $diff(\varphi)$ using Eq(12);
Step 4. If $diff(\varphi) \geq diff_{min}$, then go to step 7;
Step 5. $diff_{min} = diff(\varphi)$, $\varphi_{min} = \varphi$;
Step 6. Utilize the gradient descent algorithm to minimize the difference;
Step 7. $\varphi = \varphi + run\_length$;
Step 8. If $\varphi < 2 * \pi$, then go to step 3;
Step 9. Return φ_{min}.

4 Symbol Recognition System

To demonstrate the effectiveness of our approach, we implement a symbol recognition system. Given a test symbol, our system can find the best matching one from a set of models.

The input to the system is a set of test symbols and a set of model symbols, which are represented in binary images. We then apply some preprocessing techniques to them and compute similarity between any pair of them. Finally, we output the best matched model for each test symbol as the result. The steps in our system are outlined as follows:

Step 1. Compress the input image. After compressed, each pixel indicates the density of the foreground pixels in the original image.
Step 2. Utilize a newly proposed approach to remove noise from the image.
Step 3. Compute the similarity between any pair of test symbols and models.
Step 4. Output the best matched model for each test symbol.

Next, we present the steps in more details.

4.1 Image Compression

Because our method to compute similarity is based on pixels, the number of pixels in a symbol gives a great impact on its time complexity. Therefore, the aim of preprocessing is to decrease the number of pixels while the information in the image loses slightly.

Assume that the input image is a $512 * 512$ 1-bit bitmap, as shown in Fig.1. It is represented as $S(x, y)$, defined in Eq. 1. We compress the original image into a grey-scale image by the following formula:

$$D(x', y') = \frac{\sum_{x=\eta*x'}^{\eta*x'+\eta-1} \sum_{y=\eta*y'}^{\eta*y'+\eta-1} S(x, y)}{\eta^2}, \tag{13}$$

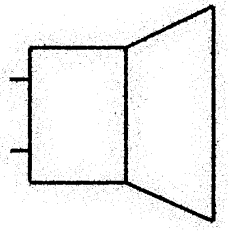


Fig. 1. A sample of $512 * 512$ input image

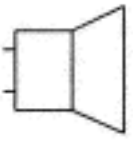


Fig. 2. $52 * 52$ compressed image

where $D(x', y')$ is the compressed image shown in Fig.2, $x' \in [0, \lceil \frac{512}{\eta} \rceil]$, $y' \in [0, \lceil \frac{512}{\eta} \rceil]$, and η^2 is the scaling factor. The value of $D(x, y)$ indicates the density of black pixels in the original image. We can see that the size of the image becomes very small after compression, while little information is lost. Hence, we can compute the similarity based on the compressed images to accelerate the whole process. Moreover, it is also a pre-requisite of our denoising approach.

4.2 Denoising

The input images, which contain the symbols, may have various kinds of noise. Noise may severely affect similarity computing, even making the result rather different from the correct answer. Hence, before computing the similarity, we must remove noise from the images.

In our denoising approach, $unidis(S_1, S_2)$ defined in Eq. 7 is the core function. Given a compressed image $D(x, y)$, we denoise it in the following steps:

Step 1. Build another image S_b of the same size with all pixels being black.
Step 2. Compute $unidis(S_b, D)$.
Step 3. If $unidis(S_b, D)$ is smaller than threshold ω, the denoising process ends. In our system, the experimental value of ω is set to 0.023.
Step 4. Remove the minimal D-valued pixel(s) in the compressed image D, turn to Step 2.

We assume that the density of noisy pixels is always lower than that of the foreground pixels in the symbols. Further since the $D(x, y)$ indicates the density of foreground pixels in the original image, removing the minimal D-valued pixels in the compressed image is equal to denoising. When the image is full of noise, the $unidis(S_b, D)$ is rather high. We go on denoising the image until $unidis(S_b, D)$ is lower than the threshold value ω, which is the result of training with a large set of test data. Fig.3 and Fig.4 show two denoising samples.

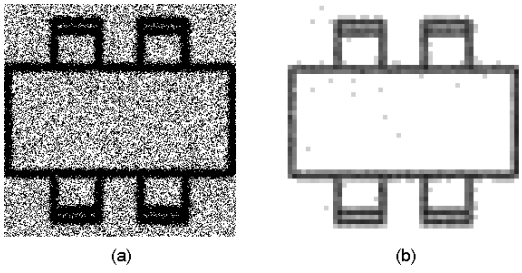


Fig. 3. A symbol full of global noise and its denoised image

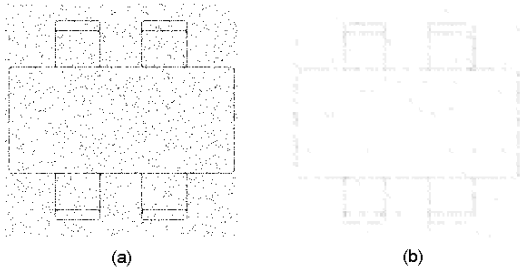


Fig. 4. A much degraded symbol with loss of connection and its denoised image

4.3 Computing Similarity

Finally we utilize the approach proposed in Section 3 to compute the similarity $sim(D_1, D_2)$ between each pair of preprocessed test symbol and model. After that, the system outputs the best matched model for each test symbol.

5 Evaluation

Tab.1 shows the result which our system obtains in the GREC 2005 contest. Our system achieves the highest overall score while the time it costs is the shortest. The tests are grouped according to the type and degree of degradation applied to the images involved in that test. Six degradation models are used and the test symbols may be rotated/scaled.

Among the six degradation models, our system achieves the highest score except for the second model. The reason is that the parameters in our denoising process are not adapted in such kind of noise. A large amount of noise remains after preprocessing, and therefore the result of similarity computing is rather bad. Our accracy in the sixth degradation model is not high either, while other systems score even worse due to severe degradation.

Regardless of noise, neither rotation transformation nor scaling transformation affects much the recognition rate of our system. However, if both exist, our system's accuracy is affected.

Table 1. The recognition accuracy of our system for different test sets used for the GREC 2005 contest

	mod 1	mod 2	mod 3	mod 4	mod 5	mod 6
without rotation and scaling	**0.997**	0.700	**0.997**	**0.983**	**0.980**	**0.867**
only with rotation	**0.950**	0.637	**0.950**	**0.967**	**0.977**	**0.513**
only with scaling	**0.965**	0.385	**0.960**	**0.930**	**0.915**	**0.620**
with rotation and scaling	**0.870**	0.635	**0.885**	**0.895**	**0.830**	**0.335**

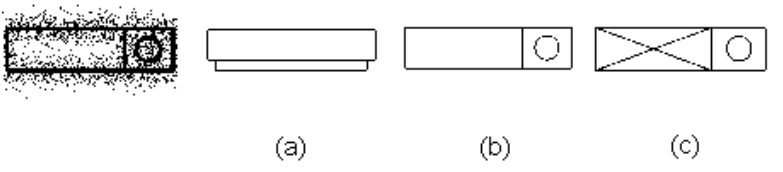


Fig. 5. Similarity between test symbol and three model symbols. (a) 0.993341 (b) 0.999193 (c) 0.995259.

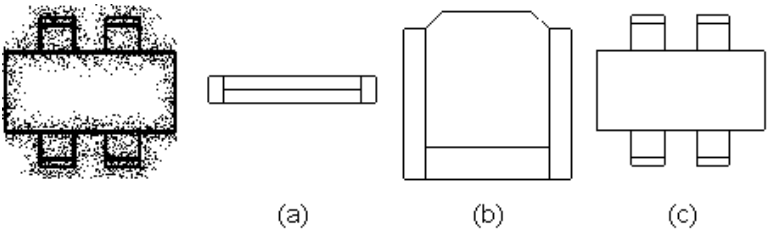


Fig. 6. Similarity between test symbol and three model symbols. (a) 0.869658 (b) 0.990119 (c) 0.999074.

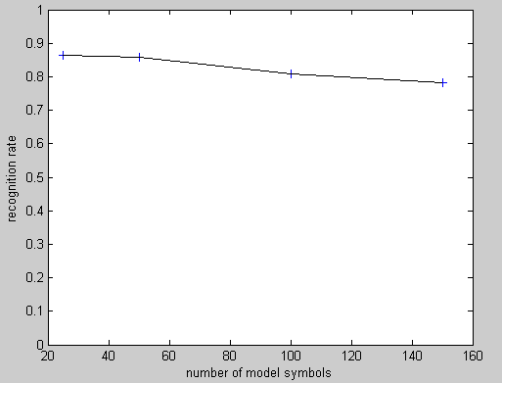


Fig. 7. Result with different numbers of model symbols

Fig.5 and Fig.6 are two groups of symbols from the contest. In each group, the leftmost symbol is the test symbol with some noise and the rest three symbols are model symbols. All the three model symbols in the first group are very similar to the test symbol, which may be even confusing to human beings, especially (b) and (c). The similarities given by our system is reasonable, which reflect the real ranking result of the similarities among them. In the second group, the result looks also very good.

Fig.7 shows the curve of average recognition rates over the number of model symbols of a test set. There are 150 different model symbols available in GREC 2005. Each test consists of 4 sets, with 25, 50, 100, 150 symbol models, respectively. As shown above, the recognition rate decreases slightly as the number of symbol models increases. We can conclude that each additional model symbol reduces the discrimination ability of our system by 0.07% averagely.

6 Conclusion

We have presented a symbol recognition system, which employs computing bipartite transformation distance, rotating symbols according to the angular distribution, compressing images and a newly-proposed denoising approach. The evaluation in GREC 2005 shows that the combination of these techniques is effective and efficient.

We note that there are still some improvements to be pursued in future work. First, we can utilize the vectorial descriptions of the model symbols. Currently, vectorial descriptions are ignored. If we can retrieve some descriptors from them, the recognition precision may be improved greatly. Second, the angular distribution in our method has some features suitable for other symbol recognition approaches. It is potential and will be studied in the future.

Acknowledgement

The work described in this paper is fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 1147/04E].

References

1. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.
2. P. J. Best and N. D. Mckay. A method for registration of 3d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.
3. M.K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory*, 8:179–187, 1962.
4. Y. Luo and W.Y. Liu. Engineering drawings recognition using a case-based approach. In *Proceedings of Seventh ICDAR*, volume 1, pages 190–194, Edinburgh, UK, 2003.

5. A. Okazaki, S. Tsunekawa, T. Kondo, K. Mori, and E. Kawamoto. An automatic circuit diagram reader with loop-structure-based symbol recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(3):331–341, 1988.
6. R. C. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. Technical report uu-cs-1999-27, Utrecht University, the Netherlands, 1999.
7. S. Yang. Symbol recognition via statistical integration of pixel-level constraint histograms: a new descriptor. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:278–281, 2005.
8. P.C. Yuen, G.C. Feng, and Y.Y. Tang. Printed chinese character similarity measurement using ring projection and distance transform. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 12(2):209–221, 1998.
9. C. Zahn and R. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans. Computers*, C-21(3):278–281, 1972.

Robust Moment Invariant with Higher Discriminant Factor Based on Fisher Discriminant Analysis for Symbol Recognition

Widya Andyardja Weliamto[1], Hock Soon Seah[1], and Antonius Wibowo[2]

[1] School of Computer Engineering, Nanyang Technological University,
Block N4, Nanyang Avenue, Singapore 639798
{ph810023, ashsseah}@ntu.edu.sg
http://www.ntu.edu.sg/sce
[2] Department of Electrical Engineering, Bandung Institute of Technology,
Ganesha 10, Indonesia 40132
antonius232@students.itb.ac.id

Abstract. In this paper, we propose a robust moment invariant which has a higher discriminant factor based on Fisher linear discriminant analysis that can deal with noise degradation, deformation of vector distortion, translation, rotation and scale invariant. The proposed system for the symbol recognition consists of 3 steps: 1) degradation model preprocessing step, 2) a different normalization for the second moment invariant and a measure for roundness and eccentricity for feature extraction step, 3) k-Nearest Neighbor with Mahalanobis distance compared to Euclidean distance and k-D tree for classifier. A comparison using multi-layer feed forward neural network classifier is given. An improvement of the discriminant factor around 4 times is achieved compared to that of the original normalized second moments using GREC 2005 dataset. Experimentally we tested our system with 3300 training images using k-NN classifier and on all 9450 images given in the dataset and achieved recognition rates higher than 86 % for all degradation models and 96 % for degradation models 1 to 4.

1 Introduction

The computational power of computer is increasing tremendously. This allows us to accomplish difficult tasks for pattern matching, symbol recognition, character recognition, finger print recognition, speech processing, etc. The moment invariants proposed by [3] is a well-known method for pattern recognition [4][5][6], but experimental results show a significant deviation value due to noise degradation, deformation, rotation and scale changes. Previous work [8] gave a revised fundamental theorem of moment invariants. Our work is inspired by [7] that gave the method of normalization to determine invariants. We proposed other normalization that yields a higher discriminant factor. The normalizer is derived from the property of SVD decomposition, which is the best linear unbiased estimator [4] for least square optimization. We define the roundness and eccentricity as a result of normalization, give

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 408 – 421, 2006.
© Springer-Verlag Berlin Heidelberg 2006

its correlation with the normalized second moment invariants [3] and evaluate our normalization using various classifiers such as k-nearest neighbor, k-D tree and neural network classifier to show its separability. In this paper we apply our proposed robust moment invariant for the symbol recognition and compare the results with [2]. Our solution is close to the previous work [1], which uses the statistical method with histogram of pixel-based descriptor and Manhattan distance. Our method to solve this problem does not use a histogram, but it is a statistical method based on binary image pixel position using three steps processing: image preprocessing, feature extraction and classification as discussed in Section 2. In Section 3 we discuss the experimental results and the conclusion is found in Section 4.

2 The Proposed Framework

2.1 Preprocessing Step

Noise Filtering and Degradation Level Measurement

The first step in our recognition system is a noise filtering and degradation level measurement. For an 8-bit image, a simple technique to measure the degradation level is by filtering the image with a low-pass filter kernel [1 4 6 4 1]/16 for both vertical and horizontal directions, and binarizing it with a threshold value t_h = 36/256. The threshold value range is around t_h (for pepper noise) to $2.5t_h$ (for hard pencil noise). The degradation level d_l is obtained from the number of pixels in the smooth area after filtering and dividing by the number of pixels in the original image.

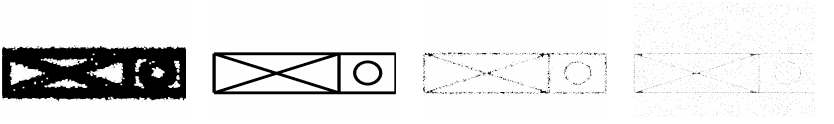
$$d_l = \frac{\sum_x \sum_y \left(f(x,y) * g(x,y,\sigma) > t_h \right)}{\sum_x \sum_y f(x,y)} \tag{1}$$

where $f(x,y)$ is the image and $g(x,y,\sigma)$ is the Gaussian low pass filter kernel with a 5 × 5 mask. This measurement allows us to detect how bad the noise level is. We used this measurement on GREC 2003 dataset [11]. For the worst degradation levels 7, 8, and 9, we have different preprocessing tasks. The degradation level value is around one for ideal-test and it will be higher for a noisy image. The highest degradation level appears when pixels in the noisy and thin lines merge after filtering.

This approach works quite well on GREC 2003 dataset. Using only one measurement, the degradation level measurement, we are able to separate all noise types and levels quite well. Unfortunately on GREC 2005 dataset [13], we are unable to separate noise type well enough using only degradation level measurement. So we combined degradation level measurement with line width measurement, noise distribution measurement and noise level measurement proposed in [12] to describe the noise features of each individual engineering drawing. With these measurements we have more degree of freedom to describe noise type and level in engineering drawings.

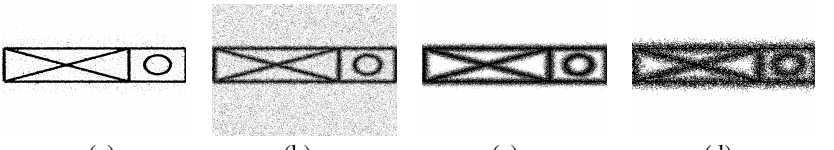
Based on the measurement results of primitives and noise we classify noise type and noise level of the drawing. From the combination of noise level measurement and ratio of primitives line width and image resolution we categorized the drawing into 4 categories, which are image with thick primitives (Fig. 1a), normal primitives (Fig. 1b), thin primitives (Fig. 1c), and very thin primitives (Fig. 1d).

Using a combination of noise distribution and degradation level measurement, we categorized noise type into 4 categories of noise type and distribution, which respectively are almost noiseless image (Fig. 2a), Gaussian noise combined with some high frequency noise (Fig. 2b), hard pencil noise concentrated around primitive (Fig. 2c), and hard pencil noise and high frequency noise combined with some Gaussian noise concentrated around the primitive (Fig. 2d).



(a) (b) (c) (d)

Fig. 1. Example of 4 categories of primitive thickness



(a) (b) (c) (d)

Fig. 2. Example of 4 categories of noise type and distribution

Morphological Operation

The next step of preprocessing is applying some morphological operations such as erosion, opening, closing, dilation, and thinning different categories of primitives and noise type. In this symbol recognition case, we experimentally categorize the images into 12 categories according to the primitives and noise type. Their corresponding image preprocessing tasks are shown in Table 1.

Table 1. The preprocessing task for different degradation levels and models

image type	preprocessing steps
I	open(0.25 lineW)→ close(0.4 lineW)→ idealizing line width→ thinning + clean
II	idealizing line width→ thinning + clean
IIA	close(0.25 lineW)→ open(0.3 lineW)→ idealizing line width→ thinning + clean
IIB	median filter(1.5 lineW)→ open(0.4 lineW)→ idealizing line width→ thinning + clean
IIC	erode(0.35 lineW)→ close(0.6 lineW)→ idealizing line width→ thinning + clean
IID	median filter(1.5 lineW)→ open(lineW)→ idealizing line width→ thinning + clean
III	close(0.25 lineW)→ open(0.3 lineW)→ idealizing line width→ thinning + clean
IIIA	close(2 lineW)→ dilate(0.1 lineW)→ close(lineW)→ idealizing linewidth→ thinning+clean
IIIB	median filter(1.5 lineW)→ open(0.25 lineW)→ idealizing line width→ thinning + clean
IIIC	median filter(1.5 lineW)→ close(0.25 lineW)→ open(0.3 lineW)→ idealizing line width→ thinning + clean
IVA	dilate(lineW)→ close(4 lineW)→ idealizing line width→ thinning + clean
IVB	filter→ dilate(1.5 lineW)→ close(4 lineW)→ component labelling→ remove small component→ idealizing line width→ thinning + clean

In the image type column of the above table, image types (I-IV) correspond to the thickness of image primitives, where (I) is thick primitive, (II) is normal primitive, (III) is thin primitive, and (IV) is very thin primitive, and (A-D) correspond to noise type and distribution. *LineW* is the line width of primitive and the diameter of disc structuring element used in open, close, erode, or dilate operation is shown in the bracket after each operation.

2.2 Feature Extraction

1. Roundness measurement

First, we propose a roundness measurement as the minimum singular value divided by the maximum singular value.

$$\text{roundness} = \sigma^2_{\min} / \sigma^2_{\max} \tag{2}$$

The roundness value is between 0 and 1. For example, a line has roundness equal to 0 with the smallest singular value equals to 0 and a disk has roundness equal to 1 with its singular values equal to identity. We compare this with the measurement for roundness or compactness defined by A.K. Jain [4] as follows:

$$\text{roundness} = \text{perimeter}^2/(4\pi \text{ area}) \tag{3}$$

Our measurement is more general as the object can be any shape. Whereas the equation (3) above may give value greater than 1 for a square shape and it depends on two parameters: the perimeter and area. The problem is that the measurement of area is not robust due to object deformation. Our proposed measurement is statistically robust. It gives the best estimation of roundness between 0 and 1, a higher discriminant factor and invariant to translation, rotation and scale changes. Next, we will show its relationship with the current moment invariant in Hu [3] and introduce a correction for normalization to achieve the same result as our proposed definition. Before that we will give a short explanation of singular values from the covariance matrix. A 2-dimensional unnormalized covariance matrix **X** is defined as follows:

$$\text{cov}(\mathbf{X}) = \mathbf{X}\mathbf{X}^{\mathrm{T}} = \begin{bmatrix} x-\bar{x} \\ y-\bar{y} \end{bmatrix} \begin{bmatrix} x-\bar{x} & y-\bar{y} \end{bmatrix} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \tag{4}$$

The singular values obtained from Singular Value Decomposition SVD are as follows:

$$\mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}} = \text{SVD}(\mathbf{X}\mathbf{X}^{\mathrm{T}}) \quad \text{and} \quad \mathbf{D} = \text{diag}(\sigma^2_{\max}, \sigma^2_{\min}) \tag{5}$$

$$\sigma^2_{\max} = \left| (a+c) + \sqrt{(a-c)^2 + 4b^2} \right| / 2 \tag{6}$$

$$\sigma^2_{\min} = \left| (a+c) - \sqrt{(a-c)^2 + 4b^2} \right| / 2 \tag{7}$$

where **D**, the diagonal matrix of singular values, is always positive and represents the variances of data distribution among its axes. The first singular value is the maximum singular value, which is the norm of covariance matrix. Next, we propose to use this norm as a normalizer of the second moments invariant [3]. The value of the square root will represent the eccentricity of data distribution. **U** and **V** are the orthogonal matrices that give the best rotation orientation along its principal axis. Now we review

the well-known feature in statistical-based pattern recognition, i.e., moment invariants. The central moments μ_{pq} and the normalized central moments μ_{pq}' in Hu [3] are defined as:

$$\mu_{pq} = \Sigma_x \Sigma_y \ (x - \overline{x}\ )^p \ (y - \overline{y}\ )^q f(x,y) \tag{8}$$

$$\mu_{pq}' = \mu_{pq} / \mu_{00}^{\gamma} \quad \text{and} \quad \gamma = (p+q+2) / 2 \tag{9}$$

We focus on two of the seven invariant moments, which are the second moments:

$$\phi_1 = \mu_{20} + \mu_{02} \tag{10}$$

$$\phi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \tag{11}$$

The first equation ϕ_1 corresponds to roundness that gives 100 % correlation if and only if the normalized equation for the second moments is as follows:

$$\mu_{pq}' = \mu_{pq} / \sigma^2_{\max} \tag{12}$$

Next, the second equation ϕ_2 corresponds to eccentricity. It happens because $\mu_{20} = a$, $\mu_{02} = c$ and $\mu_{11} = b$. So the normalized first equation ϕ_1' equals to:

$$\phi_1' = (\mu_{20} + \mu_{02}) / \sigma^2_{\max} = (\sigma^2_{\max} + \sigma^2_{\min}) / \sigma^2_{\max} = 1 + \text{roundness} \tag{13}$$

and the normalized second equation ϕ_2' equals to:

$$\begin{aligned} \phi_2' &= ((\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2) / (\sigma^2_{\max})^2 = ((\sigma^2_{\max} - \sigma^2_{\min}) / \sigma^2_{\max})^2 \\ &= (1 - \text{roundness})^2 = \text{eccentricity}^2 \end{aligned} \tag{14}$$

Table 2 shows our experimental result with the symbol recognition database from GREC 2003 contest [11], which yielded clean images after using a simple preprocessing and GREC 2005 contest [13], which yielded noisier images even after using a complex preprocessing. The proposed normalized equations ϕ_1' and ϕ_2' have 5.2 and 2.6 times higher discriminant factor or 2.2 and 1.6 times higher recognition rate than the first original equations ϕ_1 and ϕ_2, respectively.

Table 2. The improvement of the proposed normalized second moments

No.	Parameter	ϕ_1'	ϕ_1	ϕ_1'/ϕ_1	ϕ_2'	ϕ_2	ϕ_2'/ϕ_2
1	Discriminant factor GREC03	22.8	3.2	7.1	30.5	6.1	5.0
2	Discriminant factor GREC05	21.2	5.2	4.1	24.5	14.9	1.6
	Average discriminant factor	22.0	4.2	5.2	27.5	10.5	2.6
1	Recognition rate GREC03	50.9 %	25.4 %	2.0	50.0 %	36.5 %	1.4
2	Recognition rate GREC05	21.2 %	6.9 %	3.1	19.0 %	7.6 %	2.5
	Average recognition rate	36.1 %	16.2 %	2.2	34.5 %	22.1 %	1.6

2. Radius min-max ratio

The radius min-max ratio is defined as the ratio between minimum radius and maximum radius $r_{\min}/r_{\max}$, where r is the Euclidean distance of each pixel to the centroid:

$$r = \sqrt{(x - \overline{x})^2 + (y - \overline{y})^2} \tag{15}$$

3. The compactness is defined as the perimeter in Eq. 19 divided by the bounding box area in the principal axis.

$$\text{compactness} = (\Sigma\, r) \,/\, ((y'_{\max} - y'_{\min})\ (x'_{\max} - x'_{\min})) \tag{16}$$

where x' and y' are the rotated image in the principal axis and obtained from:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{U}'^{\mathrm{T}} \begin{bmatrix} x - \bar{x} \\ y - \bar{y} \end{bmatrix} \tag{17}$$

$$\mathbf{U}' = \mathrm{diag}(1, \det(\mathbf{U}))\, \mathbf{U} \tag{18}$$

where $\mathbf{U}$ is the rotation matrix correspond to the normalization of orientation to the principal axis and a correction $\mathbf{U}'$ is needed to remove the ambiquity since the singular value is always positive and the determinant of $\mathbf{U}$ may not always equal to one.
The perimeter is defined as the sum of all the distance of each pixel to the centroid:

$$\text{perimeter} = \int \sqrt{x^2(t) + y^2(t)}\; dt \equiv \Sigma\, r \tag{19}$$

where t is necessarily the boundary parameter but not necessarily its length.

4. Normalized pixel-perimeter is defined as the number of pixels, N, multiplied by the perimeter and divided by the maximum singular value.

$$\text{Normalized pixel-perimeter} = \mathrm{N}\, (\Sigma\, r) \,/\, \sigma^2_{\max} \tag{20}$$

5. Bounding box ratio = $(y'_{\max} - y'_{\min}) \,/\, (x'_{\max} - x'_{\min})$ (21)

6. Normalized perimeter is defined as the perimeter square divided by the number of pixels and the maximum singular value.

$$\text{Normalized perimeter} = \text{perimeter}^2 / (\mathrm{N}\sigma^2_{\max}) = (\Sigma\, r)^2 / (\mathrm{N}\sigma^2_{\max}) \tag{22}$$

7. Average standardized radius is defined as the mean of radii divided by its maximum radius.

$$\text{Average standardized radius} = (\Sigma\, r) \,/\, (\mathrm{N}\, r_{\max}) \tag{23}$$

8. Normalized perimeter square is defined as the perimeter square divided by the maximum singular value.

$$\text{Normalized perimeter square} = \text{perimeter}^2 / \sigma^2_{\max} = (\Sigma\, r)^2 / \sigma^2_{\max} \tag{24}$$

9. Inverse normalised perimeter is defined as the inverse of normalized perimeter (see Eq. 22).

$$\text{Inverse normalised perimeter} = (\mathrm{N}\sigma^2_{\max}) \,/\, (\Sigma\, r)^2 \tag{25}$$

10. Normalized second moment invariant for eccentricity is defined as:

$$\text{Eccentricity}^2 = ((\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2) \,/\, (\sigma^2_{\max})^2 \tag{26}$$

11. Average normalized angular is defined as the average of angular pixel distribution of every point around its centroid divided by its maximum value.

$$\text{Average normalized angular} = \Sigma\, \mathrm{P}(\theta) \,/\, (\mathrm{N_a} \max(\mathrm{P}(\theta))) \tag{27}$$

where $\mathrm{N_a}$ is the number of angular bin and $\mathrm{P}(\theta)$ is a probability density function of angular pixel distribution.

12. Average normalized radius is defined as the average of radial pixel distribution of every point around its centroid divided by its maximum value.

$$\text{Average normalized radius} = \Sigma\, \mathrm{P}(r) \,/\, (\mathrm{N_r} \max(\mathrm{P}(r))) \tag{28}$$

where $\mathrm{N_r}$ is the number of radial bin and $\mathrm{P}(r)$ is a probability density function of radial pixel distribution.

13. Average principal-axis-norm radius is defined as the mean of radii divided by the principal-axis length.

$$\text{Average principal-axis-norm radius} = (\Sigma\, r) \,/\, (\mathrm{N}\,(x'_{\max} - x'_{\min})) \tag{29}$$

2.3 Feature Selection

The feature selection is based on the discriminant factor d_f and its correlation to other features using Fisher discriminant analysis. The discriminant factor is obtained from the ratio between standard deviation "between class" σ_b and standard deviation "within class" σ_w. The first feature must have a high discriminant factor, and the next feature should have the lowest correlation toward zero as shown in Table 3.

Equation (30) until (33) is the corresponding equation for the discriminant factor and its variables. n_c is the number of classes which is 150 symbols from GREC 2005 contest which is 2.5 times the number of symbols in GREC 2003 contest. μ_b is the mean between classes which is the mean of the mean within class μ_{wj}.

Table 3. The comparison of discriminant factor, its correlations to the roundness and the recognition rate

No. feature	Feature name	GREC 2005 dataset					GREC 2003 dataset				
		σ_b	σ_w	d_f	cor. to no. 1	recog. rate %	σ_b	σ_w	d_f	cor. to no. 1	recog. rate %
1	roundness	.303	.0143	21.2	1.000	21.2	.299	.0131	22.8	1.000	50.9
2	radius min/max	.172	.0108	15.9	0.223	12.0	.180	.0122	14.8	0.193	34.0
3	compactness	.153	.0238	6.4	-0.147	8.8	.180	.0248	7.3	-0.287	36.0
4	pixelperimnorm	.189	.0228	8.3	0.767	9.1	.181	.0234	7.7	0.778	31.9
5	boundbox ratio	.237	.0123	19.3	0.948	16.6	.255	.0140	18.2	0.946	39.3
6	perimnorm	.159	.0077	20.5	0.976	19.6	.161	.0069	23.4	0.982	53.5
7	avgstdradius	.107	.0104	10.3	0.563	10.4	.089	.0095	9.4	0.621	26.1
8	perimsqrnorm	.133	.0348	3.8	0.823	7.3	.136	.0202	6.7	0.836	37.9
9	invperimnorm	.187	.0087	21.6	-0.946	19.9	.200	.0072	27.8	-0.941	53.6
10	eccentricity	.271	.0111	24.5	-0.966	19.1	.287	.0094	30.5	-0.968	50.0
11	avgangularbin	.172	.0241	7.1	0.471	7.4	-	-	-	-	-
12	avgradiibin	.081	.0254	3.2	-0.096	4.4	-	-	-	-	-
13	avgpanradius	.067	.0070	9.5	0.628	11.7	-	-	-	-	-

$$d_f = \sigma_b / \sigma_w \tag{30}$$

$$\sigma_b = (\Sigma_j\, (\mu_{wj} - \mu_b) \,/\, (n_c - 1))^{1/2} \tag{31}$$

$$\mu_b = (\Sigma_j\, \mu_{wj}) \,/\, n_c \tag{32}$$

$$\sigma_w = (\Sigma_j\, \sigma_{wj}) \,/\, n_c \tag{33}$$

2.4 Classifier

As a comparison, we use the simplest k-nearest neighbor (k-NN) classifier with Mahalanobis distance and compare its performance with Euclidean distance. To show the separability power of our robust moment invariant, we use k-D tree [9][10] classifier and multi-layer feed-forward neural network classifier [5] as discussed in Section 3. The average standard deviation within class σ_w is utilized as a weight to give a higher separability for a lower standard deviation among the features. A sample s is assigned

as class C_j if and only if the Mahalanobis distance d_M between the sample feature vector $\mathbf{s}_i$ and the mean within class μ_{wij} for all features i among classes j is minimum, where $i = 1\ldots n_f$ and $j = 1\ldots n_c$, where n_f is the number of features.

$$s \subset C_j \quad \text{iff} \quad \min_j d_M(\mu_{wj}, \mathbf{s}) = \min_j (\Sigma_i ((\mu_{wij} - \mathbf{s}_i)^2) / \sigma_{wi})^{1/2} \tag{34}$$

2.5 Training

Preprocessing results may not totally recover noisy images. To increase robustness in noisy images, we choose the image preprocessing with broken lines instead of continuous lines detection. Moreover, we also provide a filtering scheme in the training process of k-NN that removes the training images with unexpected groups of pixels. This selection process is based on a training error, which is the mean of distance between the feature vector of a sample image and a corresponding ideal or model image. In this case, we experimentally choose the threshold value of training error t_e to be around 0.0025 with the assumption that all features of element vector have values between 0 and 1. This will cover most of the good preprocessing results in the training set T to avoid mis-training with degrading images or over training such as the k-D tree result as shown in the Section 3.

$$s \subset T \quad \text{iff} \quad (\Sigma_i ((\mu_{wi} - \mathbf{s}_i)^2) / \sigma_{wi}) / n_f \; < \; t_e \tag{35}$$

3 Experimental Results

Using the simplest k-NN-based classifier, the best recognition rate can be achieved is around 99 % with a 7-feature vector (1-2-7-4-3-5-9) for GREC 2003 dataset and around 86 % with 8-feature vector (1-2-7-11-5-12-13-4) for GREC 2005 dataset as shown in Table 4. The highest recognition rate in GREC 2005 image dataset is achieved with a 7-feature vector by over training with degrading images. Hence, erroneous recognition may occur for the recognition of better preprocessing result, for

Table 4. The percentage of recognition rate vs the number of features

No. of features	Feature code	GREC 2005 image dataset			GREC 2003 image dataset		
		Ek-NN	Mk-NN	kD tree	Ek-NN	Mk-NN	kD tree
1	1	21.17	21.17	87.54	50.93	50.93	64.56
2	1-2	61.03	60.94	92.91	88.80	88.87	93.71
3	1-2-7	77.48	77.96	96.28	94.76	94.94	98.58
4a	1-2-7-4	78.41	79.53	95.51	97.78	98.25	99.71
5a	1-2-7-4-3	75.35	77.14	94.98	98.22	98.88	99.93
6a	1-2-7-4-3-5	78.17	79.82	95.79	98.47	98.98	99.93
7a	1-2-7-4-3-5-9	78.65	80.40	95.78	98.48	99.07	99.94
4b	1-2-7-11	80.84	81.97	96.20	-	-	-
5b	1-2-7-11-5	83.89	84.74	96.33	-	-	-
6b	1-2-7-11-5-12	84.13	85.50	96.32	-	-	-
7b	1-2-7-11-5-12-13	84.90	85.99	96.44	-	-	-
8	1-2-7-11-5-12-13-4	84.46	85.93	96.37	-	-	-

example from the ideal-test image dataset. This means that the subtotal testing recognition rate from the degradation model 1 to 4 may decrease as shown in Table 5. Similar problem may occur with other method such as k-D tree, but this is a kind of over training that does not represent the statistical-based classifier using Fisher discriminant analysis.

The recognition rate using Mahalanobis distance (Mk-NN) is 0.5 % to 3 % higher compared to those using Euclidean distance (Ek-NN) for the number of features greater than three. The highest recognition rate of 99.9 % for GREC 2003 dataset is obtained using k-D tree closest point with 2534 training images to construct a binary-tree classifier. The speed to construct a binary-tree classifier is very fast so that we can use more training set images. This highest recognition rate can be achieved since we incorporate the most extreme degradation, distortion, rotation and scale changes in the training set and it shows that our robust moment invariants have a higher separability power.

Table 5. The percentage of recognition rate distribution for GREC 2005 image dataset

		Robust Moment Invariant enhanced							RMI contest result as reference						
dataset	group	mod1	mod2	mod3	mod4	mod5	mod6	total	mod1	mod2	mod3	mod4	mod5	mod6	total
grec05	100	100	100	93	100	85	62	90.0	98	93	98	96	67	56	84.7
	150	100	100	98	98	87	56	89.8	98	87	98	93	58	56	81.7
	25	100	100	88	100	92	58	89.7	100	100	100	100	88	70	93.0
	50	100	100	100	100	96	60	92.7	98	94	100	100	72	66	88.3
grec05-rot+scl	100	82	94	98	74	64	42	75.7	78	76	50	38	48	8	49.7
	150	86	96	90	72	66	40	75.0	80	66	64	26	30	12	46.3
	25	82	92	98	60	64	42	73.0	78	82	40	28	44	12	47.3
	50	84	96	96	74	68	48	77.7	74	80	56	40	44	8	50.3
grec05-Rot	100	99	99	95	99	87	36	85.8	97	92	98	92	64	41	80.7
	150	99	100	96	97	84	29	84.2	94	91	96	85	59	24	74.8
	25	96	100	100	100	92	40	88.0	100	100	98	98	76	36	84.7
	50	98	98	100	100	90	44	88.3	92	96	98	96	70	40	82.0
grec05-Scl	100	92	94	96	84	68	56	81.7	64	66	78	42	54	18	53.7
	150	96	96	90	74	76	54	81.0	82	66	70	44	48	20	55.0
	25	92	100	94	78	80	60	84.0	80	82	60	38	56	20	56.0
	50	98	98	98	82	88	58	87.0	72	74	66	50	58	14	55.7
subtotal testing	A	95.1	98.1	95.6	89.3	81.5	48.4	84.7	88.6	85.4	83.0	71.6	59.2	33.9	70.3
sample-test-rot	100	99	100	95	98	89	27	84.7	-	-	-	-	-	-	-
	150	99	100	97	95	85	25	83.5	-	-	-	-	-	-	-
	25	100	100	100	100	72	28	83.3	-	-	-	-	-	-	-
	50	98	98	98	100	90	18	83.7	-	-	-	-	-	-	-
sample-test	100	100	100	96	100	81	66	90.5	-	-	-	-	-	-	-
	150	100	99	97	100	87	66	91.5	-	-	-	-	-	-	-
	25	100	100	96	92	88	68	90.7	-	-	-	-	-	-	-
	50	100	100	100	96	96	62	92.3	-	-	-	-	-	-	-
subtotal training	B	99.5	99.6	96.9	98.0	86.4	45.1	87.6	-	-	-	-	-	-	-
total dataset	A+B	96.6	98.6	96.1	92.4	83.2	47.2	85.7	-	-	-	-	-	-	-

Table 6. The percentage of performance class comparison with GREC 2005 contest results

dataset	RMI enhanced			RMI contest result			ZW contest result			FM contest result		
	mod 1-4	mod 5-6	total	mod 1-4	mod 5-6	total	mod 1-4	mod 5-6	total	mod 1-4	mod 5-6	total
grec05	98.6	73.8	90.3	96.4	64.2	85.7	97.8	75.2	90.3	91.9	92.3	92.1
grec05-rot+scl	85.9	54.3	75.3	59.8	25.8	48.4	88.5	48.5	75.2	82.1	58.3	74.2
grec05-rot	98.3	61.5	86.1	94.5	49.8	79.6	95.8	51.2	80.9	87.6	74.5	83.2
grec05-scl	91.4	67.5	83.4	64.6	36.0	55.1	90.3	66.0	82.2	81.0	76.8	79.6
subtotal testing	94.5	65.0	84.7	82.2	46.6	70.3	93.8	60.8	82.8	86.5	77.1	83.3
sampletest-rot	98.9	55.5	83.9	-	-	-	-	-	-	-	-	-
sampletest	98.1	76.0	91.2	-	-	-	-	-	-	-	-	-
subtotal training	98.5	65.7	87.6	-	-	-	-	-	-	-	-	-
total	95.9	65.2	85.7	-	-	-	-	-	-	-	-	-

The GREC 2005 image dataset consists of 150 symbols as shown in Figure 3 and total 9450 images which consist of 150 ideal-test images, 3300 training images (sample test set), and 6000 testing images. Whereas, the GREC 2003 image dataset is a sub-set of GREC 2005 image dataset. The GREC 2005 dataset is designed to test the scalability of symbol recognition system to cope with the increasing number of symbols with more severe degradation levels, but less degradation number of models. The corresponding results of the preprocessing step for the various degradation models are shown in Figure 4. We found that the feature vector for GREC 2003 dataset, which mainly based on the radii measurement, is not sensitive to the angular pixel variation

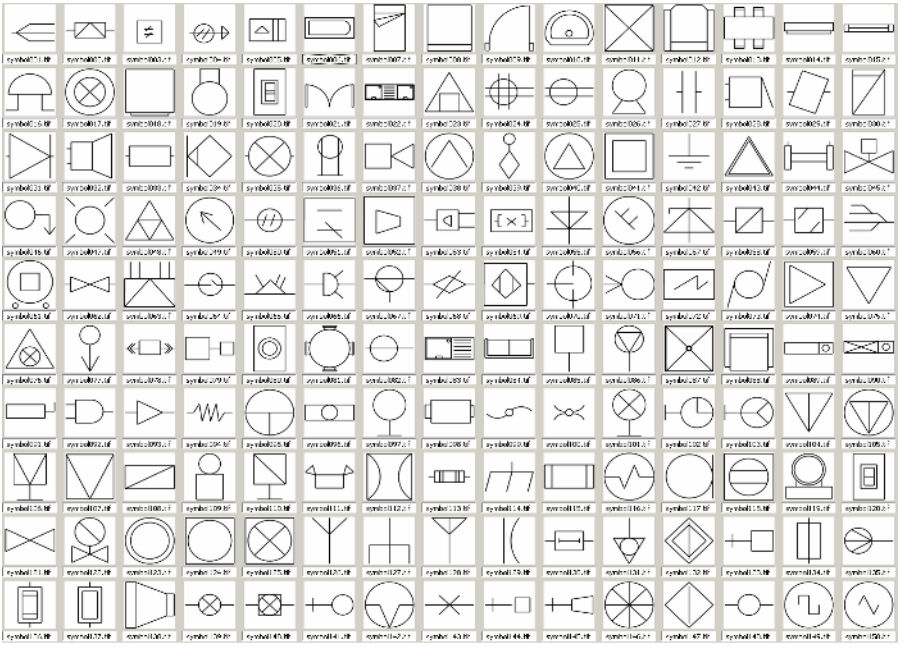


Fig. 3. The 150 symbols in GREC 2005 image dataset

that exists in GREC 2005 dataset. This case was not detected in the GREC 2005 contest since we did not have better noise preprocessing which resulted in a 70 % recognition rate. Adopting the Adaptive Noise Reduction preprocessing [12] yields 10 % improvement. To increase the separability power for better recognition rate, therefore, we have to add some additional features such as average normalized angular based on angular pixel distribution (feature number code 11) and radial pixel distribution to increase radii sensitivity (feature number 12 and 13). These three features yield 6 % additional gain up to 86 %.

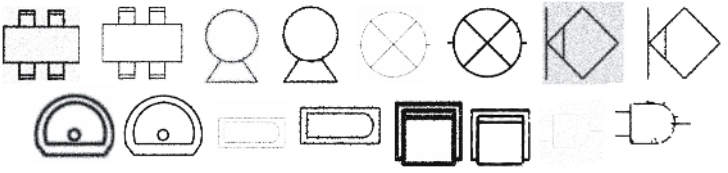


Fig. 4. The preprocessing results for the various degradation models degrad 4-7-9 & mod 2 to 6

The system has false alarm detection on the similar objects as shown in Figure 5 for symbol 11-87, 98-115, 20-120, 136-137, 149-150-49, 43-75, 48-23.

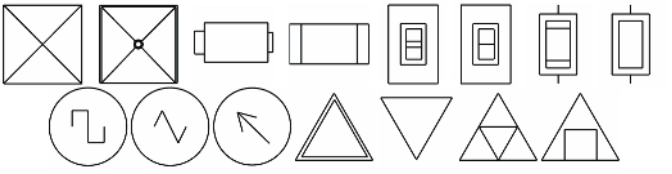


Fig. 5. False alarm detection on the similar objects for GREC 2005 dataset

Figure 6 shows the result of GREC 2003 dataset with the most difficult test images to be recognized are those with degradation 7 and scale-rotation. It also shows that our robust moment invariants are 100 % scale invariant, but it is slightly dependent on rotation changes since the orientation measurement used as the normalization in the principal axis is not as stable as the scale from the diagonal matrix of SVD.

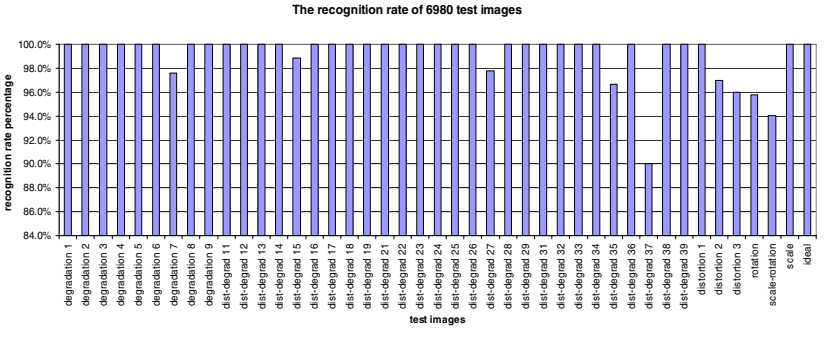


Fig. 6. The recognition result of 6980 images using Mk-NN classifier for GREC 2003 dataset

Figure 7 shows the result of GREC 2003 dataset that the symbols 1, 2, 4, 11, 47, 48 and 51 have adjacent distance such as 11-48-33, 51-15, 47-42, 2-5, 4-3 and 1-35, or severe degradation and distortion level. These happen since we use the centroid of each class to calculate the distance from the sample. However, using k-D tree classifier these cases do not happen since the constructed tree is trained using all the supporting vectors. It shows that all symbols can be separated well using k-D tree and our proposed moment invariant features.

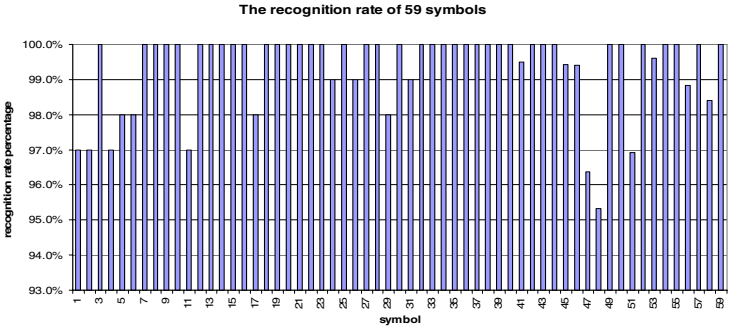


Fig. 7. The recognition result of 59 symbols using Mk-NN classifier for GREC 2003 dataset

A comparison result has also been done for multi-layer feed-forward neural network with a lower recognition rate of 91.48 % using 3 layers, 15:30:59 neurons, 59 ideal-test training set images, tangent sigmoid activation function for the hidden layer, pure linear activation function for the output layer and Levenberg-Marquadt optimization. This lower recognition rate happens because zero degradation, distortion, rotation and scale test images are trained with a tradeoff between the convergent speed and memory limitation. Figure 8 shows the result of GREC 2003 dataset that the convergence rate of the network training is relatively fast using a small training set and

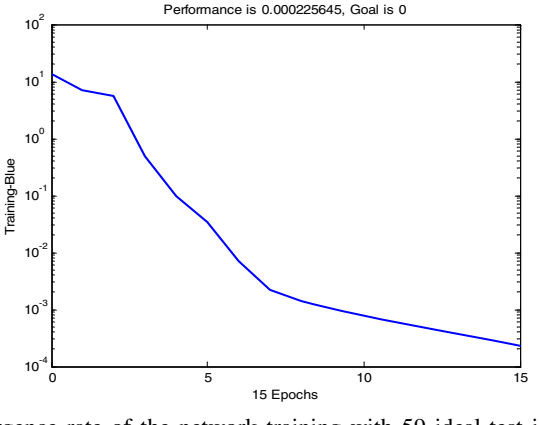


Fig. 8. The convergence rate of the network training with 59 ideal-test images and 10:30:59 neurons using Levenberg-Marquadt optimization for GREC 2003 dataset

achieved a Mean Square Error less than 0.001. It means that the separability power of all features is strong. The separability power based on the roundness is also shown in [14] for the first four features value: roundness, radius min/max, average standardized radius and normalized pixel-perimeter. The symbols that have the same values on 2 features would not have the same values for another feature. Most of the symbol can be separated using these 4 features with the recognition rate of 99 % as shown in Table 4.

4 Conclusion

In this paper, we propose novel robust moment invariants by normalizing the second moments using the norm of covariance matrix and defining the roundness and eccentricity measurement to yield a higher discriminant factor of around 5 times by average the test results on GREC 2003 and GREC 2005 datasets. We experimentally evaluate the proposed framework using various classifiers such as k-NN based on Fisher discriminant analysis, k-D tree and neural network and show that by using Mahalanobis k-NN, higher separability with improved recognition rate can be achieved, which is around 86 % for GREC 2005 dataset and 99 % for GREC 2003 dataset. Note that our feature vector is still maintained to be less than 10 features, which is useful for any object recognition based on statistical and geometrical analysis.

5 Discussion

Our strategies to achieve general symbol recognition are: First, to deal with various noises processing model is to classify noise model automatically in some measurements with smooth step changing value in various model degradations. The more noise measurement model we have, the better noise preprocessing model we get. For real implementation, other problem will happen, i.e., the localization of symbols as the next challenge. Second, the roundness measurement is inspired by the scale space problem, the properties of singular value decomposition and invariant to scale and rotation. Basically the proposed roundness is a dimensionless measurement for the second moment or covariance measurement in x-y axes, which will then be transformed into principal/major axis using SVD. Singular value itself is a stable measure regardless of the orientation, but it still has a dimension. The maximum and minimum singular values are the covariance along major and minor axis respectively. The normalization of minimum singular value by maximum singular value is performed so that the dimension can be eliminated and the measurement always stable between 0-1.

References

1. Yang, S.: Symbol Recognition via Statistical Integration of Pixel-Level Constraint Histograms: A New Descriptor. IEEE Trans Pattern Analysis & Machine Intelligent, vol. 27, no. 2, (Feb. 2005) 278-281.
2. Valveny, E., and Dosch, P.: Symbol Recognition Contest: A Synthesis. In: Llados, J. and Kwon, Y.B. (eds.): Graphics Recognition: Recent Advances and Perspectives, GREC 2003. Lecture Notes in Computer Science, vol. 3088. Springer-Verlag, Berlin Heidelberg New York (2004) 368-385.

3. Hu, M.K.: Visual Pattern Recognition by Moment Invariants. IRE Trans. Information Theory, vol. 8, (1962) 179-187.
4. Jain, A.K.: Fundamentals of Digital Image Processing. Prentice-Hall Inc. (1989).
5. Gonzales, R.C. and Woods, R.E.: Digital Image Processing, Prentice-Hall Inc. (2002).
6. Mindru, F., Tuytelaars, T., Gool, L.V., and Moons, T.: Moment Invariants for Recognition under Changing Viewpoint and Illumination. Computer Vision and Image Understanding, vol. 94, (2004) 3-27.
7. Rothe, I., Susse, H., and Voss, K.: The Method of Normalization to Determine Invariants. IEEE Trans. Pattern Analysis and Machine Intelligent, vol. 18, no. 4, (Apr. 1996) 366-376.
8. Reiss, T.H.: The Revised Fundamental Theorem of Moment Invariants. IEEE Trans. Pattern Analysis and Machine Intelligent, vol. 13, no. 8, (Aug. 1991) 830-834.
9. DeBerg, M., vanKreveld, M., Overmars, M., and Schwarzkopf, O.: Computational Geometry: Algorithms and Applications, Springer Verlag, (2000).
10. http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=7030&objectType=file
11. http://www.cvc.uab.es/GREC2003/SymRecContest/index.htm
12. Zhang, J., Zhang, W., and Wenyin, L.: Adaptive Noise Reduction for Engineering Drawings based on Primitives and Noise Assessment. Sixth IAPR Int. Workshop on Graphics Recognition GREC 2005, (Aug. 2005) 136-145.
13. http://symbcontestgrec05.loria.fr/index.php
14. Weliamto, W.A., and Seah, H.S.: Robust Moment Invariant with Higher Discriminant Factor for Symbol Recognition. Sixth IAPR Int. Workshop on Graphics Recognition GREC 2005, (Aug. 2005) 68-76.

Graphics Recognition: The Last Ten Years and the Next Ten Years

Karl Tombre

LORIA-INPL, École des Mines de Nancy, Parc de Saurupt,
CS 14234, 54042 Nancy CEDEX, France
Karl.Tombre@loria.fr

Abstract. GREC'05 marked the 10th anniversary of the series of international workshops on graphics recognition, for which the first edition was held in Penn State in 1995. At the end of the workshop, a panel discussion was held to take a broader view of our field, to examine the shift in issues addressed by research teams, and define some challenges for the coming years. This paper tries to summarize the results of the panel discussion.

1 Traditional Issues, Emerging Themes and Failures

1.1 Topics for Which Interest Seems to be Drifting Away

Looking at the research themes of the groups active in graphics recognition, it appears that the interest for some themes has more or less faded away. This is especially the case for document image analysis methods for pre-processing and text/graphics separation. The question naturally arises whether these issues are considered to be solved. Although we definitely have state-of-the-art methods for these problems, it probably cannot be said that these methods can be considered as definitive solutions. However, the problem has shifted in some way:

- Some applications aim at full reconstruction of the document in a high-level CAD or SIG system. In that case, users need excellent low-level results and the existing methods are not good enough. Hence, pragmatically, the low-level segmentation tools are performed manually or semi-automatically.
- Other applications do not aim at full reconstruction but rather at providing tools for browsing and indexing large documentation databases. In this case, the available methods are usually deemed to be sufficient and no strong need is present for more work on segmentation.

Several panelists noted that there seems to be less interest in building complete systems, at least from a scientific point of view. This is not due to an applicative lack of interest for such systems, but is rather an indication of the maturity to which we have come in several areas. However, there is a belief that there are still open problems in building complete, robust systems and researchers are encouraged not to forget this aspect.

W. Liu and J. Lladós (Eds.): GREC 2005, LNCS 3926, pp. 422–426, 2006.
© Springer-Verlag Berlin Heidelberg 2006

1.2 Topics Which Have Been Along for Some Time and Are Still Very Much Relevant

A number of issues have been constantly present in our research field and remain so. *Vectorization*, i.e. raster-to-vector conversion, is still addressed and there is room for improvement, as the arc detection contest held at the workshop made clear. While traditional problems like the vectorization of archival material may be less present, new issues arise with hand-made sketches, for instance. Among the directions mentioned for further improvement, let us cite the direct processing of the grayscale image, as it seems that we are slowly coming to the limit of the accuracy we can get out of the black and white pixels of a binary image. Have we really made progress? At least, we have certainly now a better understanding of what the good techniques are and what their limitations are.

Symbol recognition is a topic receiving a lot of attention. While we are reasonably good at recognizing fully segmented, simple symbols, as illustrated by the symbol recognition contest held at the workshop, there are a number of open issues with handwritten symbols, complex symbols made by the combination of simple symbols and textual annotations, and symbols which can not easily be segmented out of their context. The scalability of symbol recognition methods, i.e. their ability to discriminate between several hundred different symbols, also remains an open problem, where we will probably need hybrid approaches incorporating both structural representations and classification techniques.

Interesting questions also arise with *sketching* and *online graphics recognition* tools. These can be used for querying existing documents or as interactive tools for a designer. The design of such tools leads to specific challenges due to the interactive process involved in the recognition and to the large variability of handwriting. On the other hand, online data provide more information than plain raster images.

1.3 Emerging Themes

There are also new issues, or issues which have received much more interest recently, mostly because of the applications they are related to. This includes taking into account *new media*, such as documents available in electronic format but with little or no structure or semantics (typical example is a PDF document), online sketching, paper ink and e-paper. It is felt that the problem of handling legacy documents will stay with us for a long time, at least with PDF and HTML documents. In that context, there is a special interest in *cultural heritage documents*, with all the specificities they represent.

Let us also mention *information retrieval and spotting* applications where there is an increasing interest in adding graphics features to the indexes and keys for navigating large information databases. We could speak of "graphics search" within document sets. Especially for legacy documents, the focus is shifting from recognition to search. *Symbol spotting* should be mentioned explicitely in this context; the idea is to be able to quickly localize instances of a possible symbol, even without having a library of known models to match against. In an ideal case, the user should be able to delineate a graphics area of interest and do a

query for similar areas within the document or in the whole document database. This could also be qualified as unsupervised or dynamic recognition of symbols, and relies on the ability to compute quickly and efficiently a number of general signatures which can be used for indexing and querying. It also necessitates taking into account relevance feedback from the user.

In these areas, there is an overlap between our community and other research and technical communities, interested in content-based image retrieval, trademark logo recognition, layout-based retrieval, etc. We should be eager to build bridges with these communities to take advantage of each other's progress.

1.4 [Putative] Failures

Our community has also had some relative failures, with respect to the hopes and plans generated and discussed at previous workshops. One of the most visible is our inability to gather around a common base of software. There is a lot of knowledge in the community, but various groups often prefer developing their own versions of various state of the art algorithms, instead of "plugging" their own work into some standard software environment. Thus, the knowledge remains partly fragmented and everybody spends a lot of time reprogramming existing methods. This is not because of lack of open software environments, but rather a common syndrome in many research groups leading to think that "home-made is better". There was no clear consensus at the panel discussion on how to avoid this or converge towards some more satisfactory solution.

2 Some "Hot" Topics

We spent some time during the panel discussion debating some of the topics perceived as being "hottest" in our field. The following lines try to summarize the (sometimes heated, but always constructive) discussions we had.

2.1 Vectorization

A first question to address with vectorization is its definition. What is a good raster-to-vector conversion? If we say that it means looking for the central lines of the raster image, we end up with the problem of having a clear definition of what the central lines are. Do we look for line fragments or for arcs? How do we discriminate between these two without contextual knowledge.

Some panelists stressed that a vectorization system has to be universal to be of any real interest, but there are actually two main choices. The first is to aim at a universal, non-contextual system which has to adopt some compromises between arc and line segment hypotheses, between simplicity and precision, etc. The second is to have an application-driven method; in that case there may be contextual knowledge about the presence and nature of arcs, the precision or the speed needed, the possibility or not to have some kind of user interactions, the presence of free curves, etc.

Therefore, despite the contests organized at the workshop, there seems not to be any universally approved way of defining what a good vectorization is supposed to be.

Still, it is felt that with respect to automatic, universal, non-contextual vectorization, we are close to optimal methods when dealing with black-and-white images. Further progress will either include working on the gray-level image to achieve subpixel precision and better curve segmentation, or progress in the seamless integration of user input and contextual knowledge into application-specific methods.

2.2 Analysis of Complete Documents

In many cases, graphics recognition is just a part of a broader picture where the aim is to analyze complete documents, also containing text, logos, illustrations, etc. The analysis itself can be for document image understanding purposes, but also (and actually more and more often) for indexing purposes, to let a user browse through a large document set and quickly retrieve or spot relevant information.

One area with increasing focus is that of heritage documents which have often been scanned in large digitization campaigns, the need appearing afterwards for tools to organize these scanned documents and for browsing through them. In some cases, there is a real problem with the image quality, as the digitization had been performed solely with the purpose of having document images readable by a human, not necessarily resolutions good enough for document image processing and analysis. When document analysis people are involved in such projects from the very beginning, an important recommendation for them is to see to that the digitizing aspects are not neglected and that the resolution with which the information is scanned and stored is good enough. On historical documents, a resolution of at least 600 dpi should be requested.

Another application domain with a large potential in the future is that of electronic documents available with little or no structure, such as PDF or HTML documents. Specific challenges arise for large-scale processing of such documents.

2.3 Performance Evaluation: The Contests

Organizing contests has been one of the strong points of our workshops since the first edition. We have also seen lately that the very fact that these contests have been organized, has driven research groups to publish their methods with reference to the contest data and evaluation methods, also at other conferences or in journals.

But there are some drawbacks and pitfalls, which were discussed at length during the panel debate. One of the controversial issues is the use of *noise models*. They are felt to be necessary to model real problems. But they have also led to participant methods which try to reduce the noise by "guessing" more or less which parameters of the noise model were applied to the data. Then the question arises: Do we actually test the quality of the de-noising method or the recognition capabilities of the method? Should we limit ourselves to real data

and not use synthetic data obtained through noise models applied to perfect data? One of the problems is the extreme cost of building groundtruth on real data...

It was also felt that there are too few participants in these contests. Many people do not take the extra step to set up everything and compare their results in an objective way with that of others. Also, there were no commercial tools this time. Several solutions were explored for getting more people into participating, including offering rewards or letting people compete anonymously.

Still, besides the contest which is more or less a "one-shot" event, the work on performance evaluation also allows us to make reference data and objective evaluation tools available to the community. The aim is to have regular benchmarking campaigns where we can really get beyond the point of having a winner of a contest, to get a better understanding of the strengths and weaknesses of various approaches taken for recognition tasks.

3 Conclusion

At the end of the panel, workshop attendants were asked to cite topics which would be discussed at GREC'2015. Here are some of the answers, without any further comments:

- Vectorization
- Same program as GREC'95
- Geometry-based or shape-based recognition
- Knowledge-based recognition
- Hardware and software technology evolution

Acknowledgments

Special thanks to the panelists, in alphabetical order Thomas Breuel, Alex Gribov, Josep Lladós, Gerd Maderlechner, Jean-Marc Ogier, and Liu Wenyin, and to all other attendees who contributed to this enriching panel discussion and hence to the conclusions drawn up here.

Author Index