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.

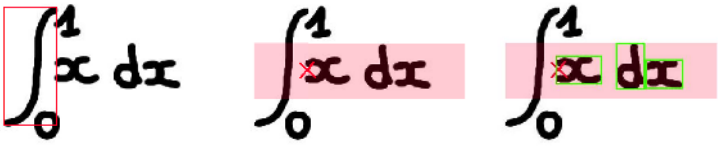
First, the operator `AT` makes it possible to choose a reference position (a point) and a research zone, depending on the last component found. Then, the operator `TERM_CMP` extracts a component:

- 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.

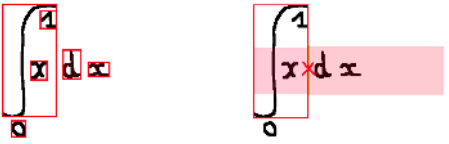
That is why we proposed to include new operators for this grammar, based on the use of the neighbourhood graph that has been presented in Sect.2.3. In this graph, the relative position of two graphical components is then given by the existence of an edge in the graph and the associated distance, which 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:

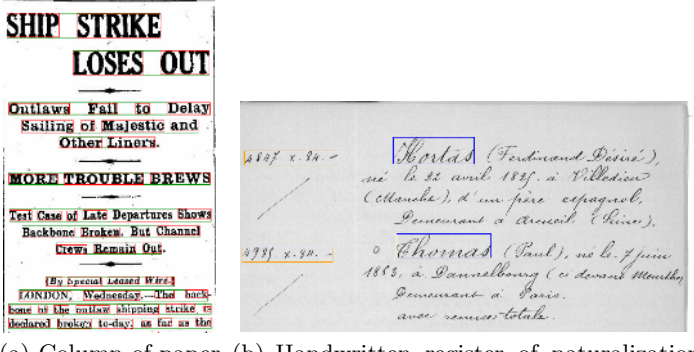
```
calculateStatDistGraph Area FavoriteDirection RequiredStatistic
```

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, Voronoı-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.

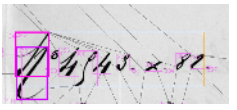


Fig. 5. Example of overlapping bounding boxes

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.