# Generating Ground Truthed Dataset of Chart Images: Automatic or Semi-automatic?

Weihua Huang[1], Chew Lim Tan[1], and Jiuzhou Zhao[1]

[1] School of Computing, National University of Singapore
3 Science Drive 2, Singapore 117543
{huangwh,tancl,zhaojiuz}@comp.nus.edu.sg

**Abstract.** Ground truthing tools mainly fall into two categories: automatic and semi-automatic. In this paper, we first discuss the pros and cons of the two approaches. We then report our own work on designing and implementing systems for generating a chart image dataset and multi-level ground truth data. Both semi-automatic and automatic approaches were adopted, resulting in two independent systems. The dataset as well as the ground truth data are publicly available so that other researchers can access them for evaluating and comparing performances of different systems.

**Keywords:** Ground truth generation, Maps and Charts Interpretation.

## 1 Introduction

Ground truthing and performance evaluation has been recognized as an important factor in advancing research in various fields. In the document analysis field, George Nagy addressed the importance of "application-oriented benchmarking" in each research area in document image recognition [1]. Ground truthed datasets that are both well established and publicly accessible are needed to evaluate and compare the performance of different image recognition and analysis systems.

As research on scientific chart recognition and understanding is a relatively young topic, there is no well established public dataset with ground truth that is specifically established for evaluating chart recognition systems. We believe that by making such a public ground truthed dataset, more attention can be drawn from other researchers that might be interested in this relatively new area. The desired dataset should have the following features:

1. The dataset should contain a sufficient number of chart images, to test the efficiency of a system working on a large scale of images.
2. The dataset should include both synthetic images and real-life images. Synthetic images are easier to generate to a large scale, while real-life images are used to present real-life effects.
3. The chart images in the dataset should cover most commonly used chart types to maintain good variety in the test images.

W. Liu, J. Lladós, and J.-M. Ogier (Eds.): GREC 2007, LNCS 5046, pp. 266–277, 2008.

4. The ground truth data should contain details in multiple aspects, so that the dataset can be used to evaluate recognition systems in various ways or at various levels.

Traditional ground truthing tools mainly fall into two categories: automatic and semi-automatic. Our work here adopts both approaches, using the automatic approach to generate synthetic images with ground truth, and the semi-automatic approach for getting ground truth from real-life images. In this paper, we are going to summarize the two systems developed by us for creating chart image datasets with ground truth. In a previous paper [2], the ground truthing system based on the semi-automatic approach was already reported. So in this paper, more emphasis will be put on the second system built based on the automatic approach.

The remaining sections of the paper cover our work in details. Section 2 surveys ground truthing works in both approaches and discusses their pros and cons. Section 3 revisits the semi-automatic system reported previously. Section 4 presents the automatic approach. Section 5 describes the final ground truthed dataset. Section 6 gives a conclusion to this paper.

## 2 Ground Truthing: Automatic vs. Semi-automatic

Most ground truthing systems reported in the literature are semi-automatic. A semi-automatic ground truthing system may involve human correction following automatic processing steps [3,4,5], or it can consist of a mixture of auto-processing steps and human inputs [8,9]. The semi-automatic approach has certain advantages. First of all, a semi-automatic system can extract ground truth data from a wide range of images with complex layout and varying types, as long as the basic processing functions are available to handle them. Secondly, a semi-automatic system is good to extract ground truth from real-life images, as human inputs or corrections can minimize the error raised from noise and distortions. Thus the resulting ground truthed dataset can reflect real-life noise and distortions. On the other hand, there are also drawbacks of the semi-automatic approach. Firstly, the process is not very efficient as it involves human effort during the process. As a result, it will be either very time consuming or very labour intensive to form a large data collection. Secondly, human verification and correction at low-level still leave certain chance to introduce inaccurate ground truth data. For example, the start point and end point of the vectorized lines may be a few pixels from the true end-points. Although the error is insignificant for most of the time, it is undesired as what we are looking for is ground "truth".

On the other hand, there are also fully automatic ground truthing systems, such as [6,7]. An automatic ground truthing system usually makes use of existing document/graphics generation packages to create datasets and captures intermediate results as the ground truth. Through literature review, we found out that automatic ground truthing is used when the targeted ground truth data only require high level details, such as the number of cells in a table or the font type of the text string. If low-level details are to be included, such as

the boundary lines of a cell in a table or the bounding box of a character, then semi-automatic approach seems to be a better choice unless such details are directly available. A typical automatic ground truthing system is computationally efficient and thus is good for the generation of a dataset with a large scale. Furthermore, the ground truth data obtained through automatic process are highly accurate. But the automatic approach also has some drawbacks. Firstly, the amount of ground truth data that can be automatically obtained is restricted, and the low-level details may not be accessible. Secondly, if the system relies on a certain graphics generation package, then the dataset created only reflects the characteristics of that package, resulting in lack of variety in the dataset created. Last but not least, the system produces synthetic images, which do not contain real-life noise and effects. To alleviate this drawback, a degradation module such as [10] is needed to introduce deformations, distortions and noise to the final images produced.

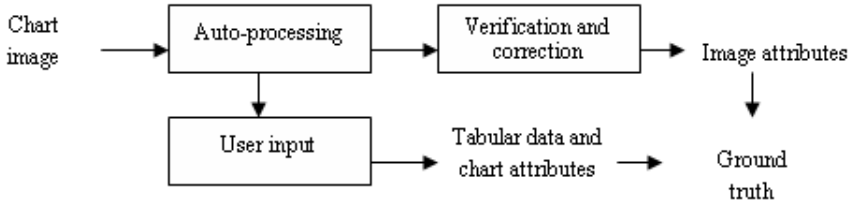


**Fig. 1.** Semi-automatic ground truthing

## 3 The Semi-automatic System Revisited

As shown in Figure 1, the system developed based on semi-automatic approach [2] accepts real-life images that were downloaded from the web or scanned in. Basic image processing techniques are performed to automatically extract image attributes of the graphical symbols in the input chart image. As the image processing techniques applied are imperfect dealing with noisy real-life images, the result obtained may be erroneous. Thus in the following step, the user needs to verify the result and make corrections when necessary. Since the structural and semantic information of the input image is not available, the user takes another responsibility which is to input these types of information. Through investigation, we found out that getting direct input from the user is more efficient than an automatic recognition of the structural and semantic information followed by corrections, as keying in high-level information is convenient for the user. The resulting ground truth information is defined to be multi-level, including the pixel level, the text level, the vector level and the chart level. By having multi-level ground truth, the dataset can be used not only by chart recognition systems but also by other systems focusing on different levels, such as text recognition systems or graphical symbol extraction systems. More details of the multi-level ground truth can be referred to in [2], which also suggested the metrics for

**Table 1.** Statistics of the dataset generated using semi-automatic approach

| **Image Information** | |
|---|---|
| Chart type | Number of images |
| Bar chart | 80 |
| Pie chart | 60 |
| Line chart | 60 |
| **Ground Truth Information** | |
| Entity | Quantity |
| Text level | |
| Text block | 4212 |
| Word | 5692 |
| Graphics level | |
| Straight line | 10165 |
| Arc | 129 |
| Chart level | |
| Chart title | 151 |
| X-axis label | 1820 |
| Y-axis label | 1308 |
| Bar | 1719 |
| Wedge | 401 |
| Polyline vertex | 681 |

performance evaluation at individual level as well as the overall performance measure. The dataset reported in [2] initially contained 120 chart images with ground truth. The chart images are of 4 different types: 2D bar chart, 2D pie chart, 3D pie chart and Line Chart. The ground truth data are of two different formats: plain text and XML format. The dataset has since been expanded to 200 chart images, with the same types and ground truth formats. Table 1 summarizes some statistics about the dataset and ground truth generated.

## 4 The Automatic System

As mentioned in section 2, both the semi-automatic approach and the automatic approach have pros and cons. One obvious problem with the semi-automatic system introduced is its low efficiency. It cost several minutes to process one chart image. Thus the dataset created is fairly small in terms of number of images. To expand the dataset to a reasonably large scale, we also implemented an automatic system. One possible way to achieve automatic ground truthing is to decode the graphics generation software and capture the intermediate data as the ground truth data. However, through investigation of some existing graphics packages, such as Microsoft Excel and PSTricks, we found that this task was not easy to achieve as most graphics generation software reveals only high-level details. Low-level details such as vector information and text bounding boxes cannot be obtained unless reverse engineering is applied. Thus we decided to implement an automatic system on our own. The system should generate chart

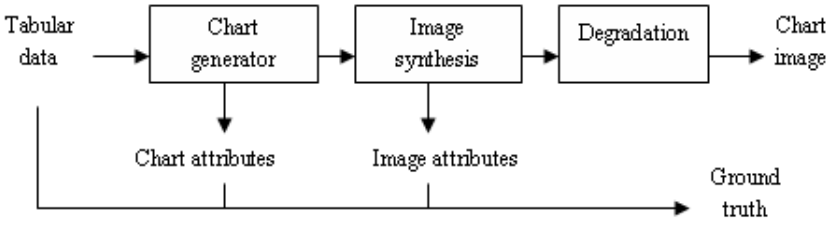


**Fig. 2.** Automatic ground truthing

images and store detailed ground truth data at all levels that were mentioned in the previous section. The major steps in the automatic system are shown in Figure 2. By comparing this figure with Figure 2, we can see that the automatic ground truthing process focuses on "generation" while the semi-automatic ground truthing process focuses on "extraction".

### 4.1 The Chart Generator

Randomly generated tabula data (label plus value) is used as the basis for chart generation. The data is passed into the chart generator to create a chart of a certain type chosen by the user. The current version of the system generates four common types of chart: 2D bar chart, 3D bar chart, 2D pie chart and 3D pie chart. Each chart type consists of a set of essential components, which can be further decomposed into text entities and regular graphical entities. Each graphical entity is represented as a combination of graphical primitives following geometric constraints. To draw a generated chart as an image, the drawing functions in the Windows GDI+ library are called to draw the graphical primitives such as line segments and arcs. The thickness of a line or an arc can be specified by user. GDI+ library also provides functions to render text strings in an image and estimate the bounding box of each text string. Figure 3 illustrates how a chart is decomposed and converted into an image. Note that the existence of axis is type-dependent. If a chart type does not require axis, such as a pie chart, then the system does not include it. Drawing 3D charts is more complicated than drawing 2D charts, in our approach the following steps are carried out:

Step 1: Draw a 2D version of the chart.
Step 2: Construct 3D chart based on the 2D version, using geometric transformations. To draw a 3D bar chart from its 2D version, translation is used. To draw a 3D pie chart from the 2D version, perspective distortion and translation are both applied.

### 4.2 The Degradation Module

For each chart generated, a clean synthetic image is created through rasterization. The degradation module is applied on the clean image to add less-than-ideal effects to simulate real-life image quality. Our degradation module is based

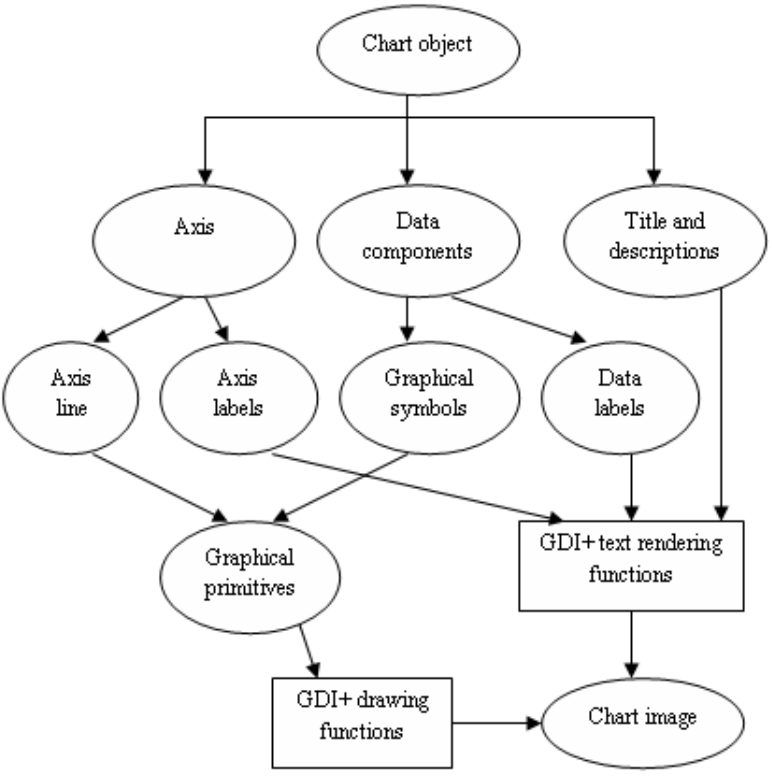


**Fig. 3.** Drawing chart image using GDI+ functions

on the degradation model proposed by Baird [10]. The original model listed 10 parameters. Considering the problem domain we are dealing with, we only adopt a subset of them. As listed in Table 2, the parameters included in our degradation module are used to perform the following tasks: rotation (skew angle), shearing, edge distortion, Gaussian noise and motion blur.

**Table 2.** Overview of the parameters in the degradation module

| Parameter | Data Type | Range | Meaning |
|---|---|---|---|
| $\beta$ | Real | $(-\pi, \pi)$ | Skew angle, measured in degrees |
| $\lambda$ | Real | [-1, 1] | Horizontal shearing factor |
| $L$ | Integer | [0, 10] | Degree of edge distortion |
| $v$ | Integer | [0, 5] | Radius of motion blur |
| $\theta_{\mathrm{b}}$ | Real | $(-\pi, \pi)$ | Angle of motion blur, measured in degrees |
| $\sigma$ | Real | [0, 50] | Degree of Gaussian noise |

– Rotation. Rotation is a deformation operation. The whole chart is rotated to add a skew angle to the image. For each pixel (x, y) in the image plane, if the skew angle is $\beta$, then the new pixel location (x', y') in vector form is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{1}$$

– Shearing. Shearing is a common deformation type that changes the shape of a geometric object. The shearing process requires one parameter, the shearing factor $\lambda = \cot\alpha$, and a pixel (x, y) will be mapped to the new location:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & \cot\alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2}$$

– Edge distortion. In real-life, distortions are very likely to occur along the edges of lines or regions, mainly due to the reproduction process such as scanning or faxing etc. To simulate edge distortion, we adopt a convolution method based on [11], with the modification that besides pixel-adding in the original method, pixel-reduction is also performed. Here pixel-adding means a pixel change from fore-ground color to background color and pixel-reduction means vice versa. A parameter L here is used to controls the degree of edge distortion.

– Motion blur. Motion blur most often occurs during a camera-based capturing process. The modeling of motion blur is based on [12]. Let f(x, y) be the input image, and H(x, y) be the blurring function. With two parameters $v$ = the level of motion blur and $\theta$ = the angle of the motion blur, the blurred image g(x, y) is generated as:

$$g(x,y) = \sum_{n=1}^{width} \sum_{m=1}^{height} f(x-n, y-m)H(n,m) \tag{3}$$

where

$$H(x,y) = \begin{cases} \frac{1}{2v+1} \text{ , if } 0 \le |x| \le (2\text{v} + 1) * \cos\theta \\ and \quad 0 \le |y| \le (2\text{v} + 1) * \sin\theta \\ 0 \qquad \text{, otherwise} \end{cases} \tag{4}$$

– Gaussian noise. Gaussian noise models the thermal noise in electronic imaging systems. To generate Gaussian noise, the crucial step is to obtain a Gaussian (normal) distribution, a random variant with its probability density function as:

$$p(X) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} \tag{5}$$

Here we use an algorithm called ran0 [13] to realize the polar method [14] for obtaining a standard normal variable X0. To add Gaussian noise, each pixel Gij in the original image is added with a value $\sigma$X0. $\sigma$ is a parameter that controls the level of noise.

### 4.3 The Ground Truth Generation

The initial tabular data become the semantic level ground truth. The vector information of the lines recorded during drawing process becomes the vector level ground truth. The text strings and their bounding boxes form the text level ground truth. The chart entities created during chart generation are also recorded to form another part of the chart level ground truth. An extra part of the ground truth contains the parameters used by the degradation module. This part of information was not obtainable using the semi-automatic approach.

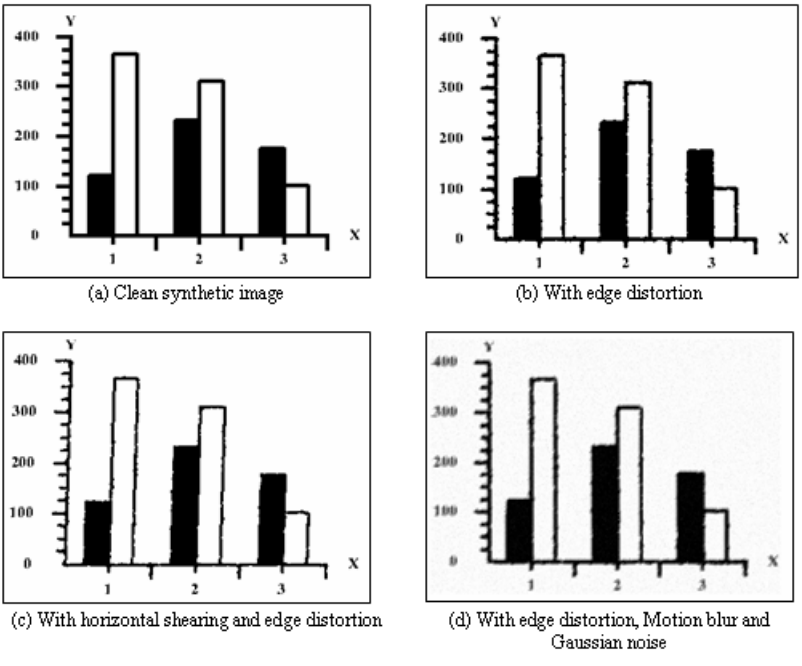


**Fig. 4.** Sample synthetic image and degradation effects

## 5 The Final Ground Truthed Dataset

### 5.1 Dataset Description

The final dataset contains two subsets from the two works we have done: a collection of real-life images and a collection of synthetic images. For the real-life collection, 200 images were collected and the corresponding ground truth data were also extracted using the system presented in [2]. In the synthetic collection produced using this automatic system, 400 clean images were created for each of the four chart types. For a clean image, one of the eight different combinations of degradation effects was added to create a noisy version. Example of a synthetic image and its corresponding degraded versions are shown in Figure 4. Thus the

**Table 3.** The final data set

| | Real | | | Synthetic | | |
|---|---|---|---|---|---|---|
| Chart type | Scanned | Downloaded | **Total** | Clean | Noisy | **Total** |
| Bar chart | 61 | 19 | **80** | 800 | 800 | **1600** |
| Pie chart | - | 60 | **60** | 800 | 800 | **1600** |
| Line chart | 14 | 46 | **60** | - | - | **-** |
| **Total** | **75** | **125** | **200** | **1600** | **1600** | **3200** |

final dataset contains 3200 chart images, with ground truth data in XML format. We put them together with the first dataset produced using the semi-automatic system, resulting in the final dataset with a total of 3400 chart images and their corresponding ground truth data. Some statistics about the complete dataset are shown in Table 3.

### 5.2 Discussions

In automatic ground truth generation, there is a trade off between the complexity of the implementation and the level of details to be kept in the ground truth data. If only tabula data are required, then the generation process is very simple: use a graphical package to create electronic charts and then convert it into image format. However, the ground truth will only be useful when evaluating a chart interpretation system that returns tabula data. Besides the tabula data itself, other metrics are also relevant and important to the performance evaluation of a system that deals with chart images, including the accuracy of graphical symbol construction, the accuracy of text segmentation and recognition etc. Thus to provide measurement for these metrics, the ground truth should be more enriched to include low-level information about graphical symbols, text bounding boxes and text strings etc. As mentioned at the beginning of section 4, the low-level information is not directly obtainable from commercial graphical packages. Thus to obtain such information, we need to implement our own functions for drawing and recording.

The accuracy of the automatically generated ground truth data is relatively higher than those generated using the semi-automatic system. However, some ground truth data may still be slightly erroneous. More specifically, the bounding box returned by the GDI+ function Graphics.MeasureString() does not reflect the true bounding box of a text string, due to the limitation of the way GDI+ computes the width of the text using hinting and anti-aliasing. The bounding box returned by the current implementation is a bit wider than the truth bounding box. The problem may be solved in the new version of the system, using alternative ways of measuring the width of text strings.

The current version of the system only takes the major chart components into consideration, including: chart axes, data components, titles and labels etc. Although these are the essential components for interpreting a chart, there are other important components to be included. For example, legends are very important in a chart with multiple data series. Grid lines may also be included because

they are very often used in real-life charts. Besides, the random text generation unit in the current system only generates very simple text strings such as numeric strings etc. Random alphabetic labels, or even sentence based descriptions should be generated. The points mentioned above will be covered as our future work.

## 6 Issues on Performance Measure

An important issue raised with the ground truthed dataset is how the data can be used to measure the performance of a system. The system to be evaluated does not need to perform all the tasks and generate all the data to match with the ground truth. It can be a line detection system, a text recognition system or an image understanding system. Thus performance score needs to be defined from multiple aspects.

Performance evaluation issue on pixel level and vector level were well described by Liu et al [6]. We also proposed ways to perform evaluation on higher levels in [2]. Below are some of them re-visited:

At a higher level which is the chart level, the detection rate of graphical data components can be obtained by calculating the data component recovery index:

$$DRI = \mu D_d + (1 - \mu)(1 - F_d) \tag{6}$$

where $\mu$ is the relative importance of detection and 1-$\mu$ is the relative importance of the false alarm. And here:

$$D_d = \frac{\Sigma_{k \in C_g} D_d(k) S(k)}{\Sigma_{k \in C_g} S(k)} \tag{7}$$

where $D_d$ is the overall detection rate, $D_d(k)$ is the detection rate for ground truth component k and S(k) is the size of ground truth component k, $C_g$ is the set of graphical data components in the ground truth.

$$F_d = \frac{\Sigma_{k \in C_d} F_d(k) S(k)}{\Sigma_{k \in C_d} S(k)} \tag{8}$$

where $F_d$ is the overall false alarm rate, $F_d(k)$ is the false alarm rate of the detected component k, $C_d$ is the set of graphical data components detected. $D_d(k)$ and $F_d(k)$ are defined as:

$$D_d(k) = \frac{S(C_d(k) \cap C_g(k))}{S(C_g(k))} \tag{9}$$

$$F_d(k) = 1 - \frac{S(C_d(k) \cap C_g(k))}{S(C_d(k))} \tag{10}$$

where $C_d(k)$ is the detected component and $C_g(k)$ is the ground truth component.

For evaluation of text recognition results, well known IR metrics precision P and recall R are used instead of detection rate and false alarm. Calculation of the precision and recall for character recognition is straightforward:

$$P = \frac{|Ch_g \cap Ch_d|}{|Ch_d|} \tag{11}$$

$$R = \frac{|Ch_g \cap Ch_d|}{|Ch_g|} \tag{12}$$

where $Ch_g$ is the set of characters in the ground truth text and $Ch_d$ is the set of characters recognized. To evaluate the accuracy of text blocks detected, a slight change needs to be made to equation (11) and (12). Instead of the intersection between two sets, the overlap between two corresponding bounding boxes should be calculated.

The overall performance score S may be defined as:

$$S = \Sigma_{i=1}^{n} w_i S_i \tag{13}$$

where $Si$ is the individual score at a single level i, and $w_i$ is the weight assigned to each $S_i (\Sigma w_i = 1)$. The weights are used to address the aspects that a system emphasizes on. For example, equation (13) is applicable for a system focusing on only one task, by turn off other performance measures (setting all other weights to zero).

## 7 Conclusion and Future Work

This paper covered our work on constructing a public dataset of chart images and generating multi-level ground truth data for the images. Two approaches were adopted to implement two independent ground truthing systems: the semi-automatic approach and the automatic approach. As the semi-automatic system was reported before, this paper emphasized more on the automatic system which was developed more recently. This paper also discussed the pros and cons of both approaches, and suggested that the ideal way of constructing a large dataset with ground truth is to combine the results of the two approaches. The resulting dataset with ground truth data is publicly accessible, through URL:

http://www.comp.nus.edu.sg/∼huangwh/GroundTruth/dataset.html

**Acknowledgement.** This research is supported by A*STAR grant 0421010085 and NUS URC grant R252-000-202-112.

## References

1. Nagy, G.: Twenty years of Document Image Analysis in PAMI. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 38–62 (2000)
2. Yang, L., Huang, W.H., Tan, C.L.: Semi-automatic ground truth generation for chart image recognition. In: Bunke, H., Spitz, A.L. (eds.) DAS 2006. LNCS, vol. 3872, pp. 324–335. Springer, Heidelberg (2006)

3. Haralick, R.M., et al.: UW English document image database I: A database of document images for OCR research. UW CD-ROM
4. Haralick, R. M. et al: UW-II English/Japanese Document Image Database: A Database of Document Images for OCR Research, http://www.science.uva.nl/research/dlia/datasets/uwash2.html
5. Phillips, I.: Users' reference manual. CD-ROM, UW-III Document Image Database-III (1995)
6. Wang, Y., Haralick, R.M., Phillips, I.T.: Automatic Table Ground Truth Generation and a Background-Analysis-Based Table Structure Extraction Method. In: 6th Int. Conf. on Document Analysis and Recognition, ICDAR 2001, Seattle, pp. 528–532 (2001)
7. Zi, G., Doermann, D.: Document Image Ground Truth Generation from Electronic Text. In: 17th Int. Conf. on Pattern Recognition, ICPR 2004, vol. 2, pp. 663–666 (2004)
8. Yacoub, S., Saxena, V., Sami, S.: PerfectDoc: A Ground Truthing Environment for Complex Documents. In: 8th Int. Conf. on Document Analysis and Recognition, vol. 1, pp. 452–456 (2005)
9. Suzuki, M., Suzuki, S., Nomura, A.: A Ground-Truthed Mathematical Character and Symbol Image Database. In: 8th Int. Conf. on Document Analysis and Recognition, vol. 2, pp. 675–679 (2005)
10. Baird, H.S.: Document Image Defect Models. In: Proceedings of IAPR Workshop on Syntactic and Structural Pattern Recognition, Murray Hill, NJ; Reprinted in: Baird, H.S., Bunke, H., Yamamoto, K.: Structured Document Image Analysis, pp. 546–556. Springer, New York (1990)
11. Zhai, J., Liu, W.Y., Dori, D., Li, Q.: A Line Drawings Degradation Model for Performance Characterization. In: 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland (2003)
12. Gonzalez, R.C., Wintz, P.: Digital Image Processing, 2nd edn. Addison-Wesley Publishing Company, Reading (1987)
13. William, H.P., Saul, A.T., William, T.V., Brian, P.F.: Numerical recipes in C++: The Art of Scientific Computing. Cambridge University Press, New York (2002)
14. Ross, S.M.: A Course in Simulation. Macmillan Publishing Company, New York (1990)