Segmentation Based Recovery of Arbitrarily Warped Document Images

B. Gatos, I. Pratikakis and K. Ntirogiannis

Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,
National Center for Scientific Research "Demokritos", GR-153 10 Agia Paraskevi, Athens, Greece
http://www.iit.demokritos.gr/~bgat/, {bgat,ipratika}@iit.demokritos.gr

Abstract

Non-linear warping appears in document images when captured by a digital camera or a scanner, especially in the case that these documents are digitized bounded volumes. Arbitrarily warped documents may have several slope changes along the text lines as well as along the words of the same text line. In this paper, a novel segmentation based technique for efficient restoration of arbitrarily warped document images is presented. The proposed technique recovers the documents relying upon (i) text lines and words detection using a novel segmentation technique appropriate for warped documents, (ii) a first draft binary image de-warping based on word rotation and translation according to upper and lower word baselines, and (iii) a recovery of the original warped image guided by the draft binary image de-warping result. Experimental results on several arbitrarily warped documents prove the effectiveness of the proposed technique.

1. Introduction

Document image acquisition by a digital camera or a flatbed scanner often results into several image distortions. Non-linear warping is a major distortion that occurs especially when the scanned documents are bounded volumes (see Fig. 1a). Warping not only diminishes document's readability but also reduces the accuracy of an OCR application.

Several techniques have been proposed for correcting the document image warping that can be classified in two main categories: (i) 2D image processing techniques ([1], [2], [3], [4], [5]) and (ii) techniques on 3D document shape reconstruction ([6], [7], [8]). Our work is related to the first category of techniques since the second category requires image capture with special camera setup as well as document surface representation by using a 3D shape model. Approaches of the first category have been reported by several authors. In [1], a deformable system to straighten curved text image is presented. Restoration is accomplished by using an active contour network based on an analytical model with cubic B-splines which have been proved more accurate than Bezier curves. A model fitting technique has also been proposed using cubic splines to define the warping model of the document image [2]. For more accurate de-warping, a vertical division of a document image into some partial document images is also suggested. Another model fitting technique [3] divides the document image into shaded and non-shaded region and then uses polynomial regression to model the warped text lines with quadratic reference curves. In [4], the texture of a document image is calculated so as to infer the document structure distortion. A mesh of the warped image is built using a non-linear curve for each text line. The curves are fitted to text lines by tracking the character boxes on the text lines. The erroneously fitted curves are detected and excluded by a post processing based on several heuristics. The approach of [5] relies on a priori layout information and is based on a line-by-line de-warping of the observed paper surface. Each letter in the input image is enclosed within a quadrilateral cell, which is then mapped to a rectangle of correct size and position in the result image.

In order to recover arbitrarily warped gray scale document images, we propose a novel technique that is based on (i) text lines and words detection using a novel segmentation technique appropriate for warped documents, (ii) a first draft binary image de-warping based on word rotation and translation according to upper and lower word baselines, and (iii) a recovery of the original warped image guided by the draft binary image de-warping result.

The remaining of this paper is structured as follows: In Section 2, we detail the proposed approach. Our experimental results are described in Section 3, while in Section 4, conclusions are drawn.

2. The proposed approach

In our approach, we address the problem of restoring arbitrarily warped gray scale document images using several distinct steps explained in the following sections. As a first step, we proceed to document image binarization of the gray scale image I_g using algorithm in [9], thus producing the binary image I_b.

2.1. Text line and word detection

In this step, a novel efficient text line and word detection technique for warped documents is introduced. First, all words are detected using a proper image smoothing. Then, horizontally neighboring words are consecutively linked in order to define text lines. For the sake of clarity, we provide the complete stepwise process in the following:

Step 1: Apply connected component labeling [10].
Step 2: Calculate the histogram with the heights of all detected connected components. The maximum value of the histogram corresponds to the average character height *H*.
Step 3: Remove noise and non-text components with height > 3*H* or < *H*/4 or width < H/4.
Step 4: Apply horizontal smoothing (RLSA [11]) with threshold *H* followed by a connected component labeling in order to detect words.
Step 6: Detect the first connected component in a top-down scanning. Set this component as word *W* with bounding box coordinates (x_1, x_2, y_1, y_2) and assign this to the first text line *L*.
Step 7: Find word W_r with bounding box coordinates $(x_1^{'}, x_2^{'}, y_1^{'}, y_2^{'})$ that neighbors at a small distance in the right side of word *W*. This is implemented as follows: from all the connected components which satisfy the condition $[y_1, y_2] \cap [y_1^{'}, y_2^{'}] \neq \varnothing$, we select the one with the smaller distance $D = x_1^{'} - x_2$ only if $0<D<5H$.
Step 8: Repeat step 7 for the new word W_r until there is no new word that can be found in the right direction.
Step 9: Repeat steps 7 and 8 for the left side of *W*.
Step 10: Label all words found in a left to right scanning order and assign them to the first text line. Furthermore, a flag is enabled to indicate that these words will not participate to further calculations.
Step 11: Repeat steps 6 to 10 for the remaining text lines until all the words are assigned to text lines.

At the end of this procedure, every text line is considered as L_i while every word *j* that belongs to line L_i as W_{ij}. An example of the text line and word detection procedure is demonstrated in Fig. 1. A final task of this phase involves: (i) merging the first two words (from the left) of every text line if the width of the first word is less than 5*H* and (ii) ignoring words with width less than 2*H*. These conditions are necessary since short words cannot lead to a safe word slope detection which is necessary to the next step. On the other hand, it is important to have accurate slope detection for the first word from the left side which guides the alignment of the entire text line.

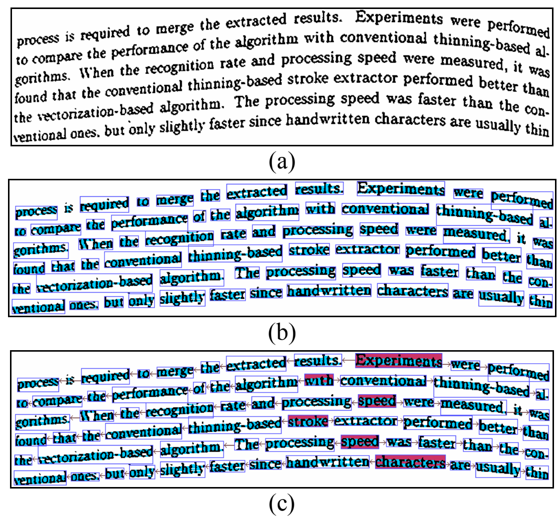


Figure 1. Example of text line and word detection: (a) original image; (b) result after horizontal smoothing for word detection (step 4) and (c) detected text lines after consecutively extracting right and left neighboring words to the first word detected (indicated with a filled box) after top-down scanning.

2.2. Word lower and upper baseline estimation

This step concerns the detection of the lower and upper baselines which delimit the main body of the words. Starting from the smoothed word image, we follow the methodology given in [12] which is used for lower baseline detection. According to this approach, a linear regression is applied on the set of points that are the lowest black pixels for each image column. In our approach, we also use a similar procedure to calculate the upper baseline. After this procedure, *upper baseline* of word W_{ij} (see Fig. 2) is defined as:

$$y = a_{ij}x + b_{ij} \tag{1}$$

Similarly, *lower baseline* of word W_{ij} (see Fig. 2) is defined as:

$$y = a_{ij}^{'}x + b_{ij}^{'} \tag{2}$$

Figure 2. Example of upper and lower baseline estimation.

2.3. Draft de-warped binary image estimation

In this step, all detected words are rotated and translated in order to obtain a first draft estimation of the binary de-warped image.

The slope of each word is derived from the corresponding baseline slopes. Upper and lower baseline slopes θ_{ij}^{u} and θ_{ij}^{l} of word W_{ij} are denoted as:

$$\theta_{ij}^{u} = \arctan(a_{ij}), \quad \theta_{ij}^{l} = \arctan(a'_{ij}) \tag{3}$$

Since the smaller absolute slope is usually the most representative, the leftmost word's slope can be defined as:

$$\theta_{i0} = \begin{cases} \theta_{i0}^{u}, \text{if } \left|\theta_{i0}^{u}\right| < \left|\theta_{i0}^{l}\right| \\ \theta_{i0}^{l}, \text{otherwise} \end{cases} \tag{4}$$

while the slope of all other words is the one with the nearest value to the slope of the previous word:

$$\theta_{ij} = \begin{cases} \theta_{ij}^{u}, \text{if } \left|\theta_{ij}^{u} - \theta_{ij-1}\right| < \left|\theta_{ij}^{l} - \theta_{ij-1}\right| \\ \theta_{ij}^{l}, \text{otherwise} \end{cases} \tag{5}$$

where j>0.

An example of detecting the words slope is given in Fig. 3b.

The rotation of the word W_{ij}(x,y) is calculated as follows:

$$\left.\begin{aligned} y^{r} &= (x - x_{ij}^{\min}) * \sin(-\theta_{ij}) + y * \cos(\theta_{ij}) \\ x^{r} &= x \end{aligned}\right\} \tag{6}$$

where $W_{ij}^{r}(x^{r}, y^{r})$ is the rotated word and $x_{ij}^{\min}$ is the left side of the bounding box of the word W_{ij}. An example of correcting the skew of the words is given in Fig. 3c.

After word rotation, all the words of every text line, except from the leftmost, must be vertically translated in order to restore horizontal alignment. The rotation and translation of the word W_{ij}(x,y) is done as follows:

$$\left.\begin{aligned} y^{rs} &= y^{r} + d_{ij} \\ x^{rs} &= x \end{aligned}\right\} \tag{7}$$

where $W_{ij}^{rs}(x^{rs}, y^{rs})$ is the rotated and translated word and y^{r} is denoted at eq. 6.

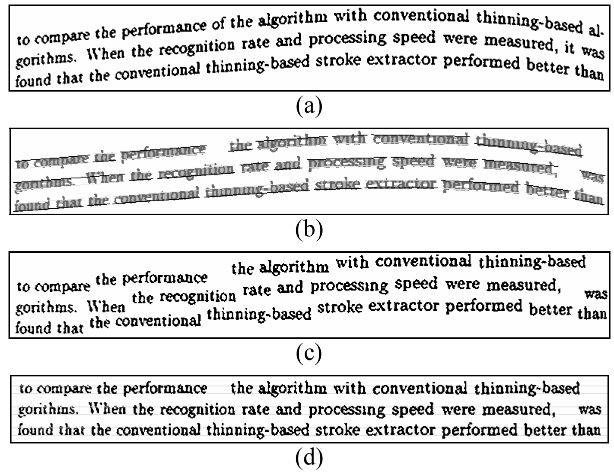


Figure 3. Example of draft binary image de-warping: (a) original image; (b) word slope detection based on upper and lower baselines; (c) word skew correction and (d) final de-warped document image after word alignment.

d_{ij} corresponds to the vertical word translation and is given by the following formula:

$$d_{ij} = \begin{cases} y_{i0}^{ru} - y_{ij}^{ru}, \text{if } \left|\theta_{ij}^{u} - \theta_{ij-1}\right| < \left|\theta_{ij}^{l} - \theta_{ij-1}\right| \\ y_{i0}^{rl} - y_{ij}^{rl}, \text{otherwise} \end{cases} \tag{8}$$

where:

$$y_{ij}^{ru} = (a_{ij} x_{ij}^{\min} + b_{ij}) * \cos(\theta_{ij}) \tag{9}$$

and

$$y_{ij}^{rl} = (a'_{ij} x_{ij}^{\min} + b'_{ij}) * \cos(\theta_{ij}) \tag{10}$$

The reason of having two alternatives of translation is that each word may be rotated either by its lower baseline or upper baseline slope. Hence, it has to be translated so that its lower or upper baseline is aligned with the lower or upper baseline of the leftmost word of the text line. An example of the draft binary image de-warping steps is given in Fig. 3.

During this step, we store the transformation factors (T_{xy}, Θ_{xy}, X_{xy}) for each pixel that is rotated and translated according to eq.7. These factors will be used at the next step of our algorithm for the final image recovery (see section 2.4). The distinct steps we follow in order to construct the binary de-warped image, are as follows:

Step 1: Initialization of dewarped binary image:

$$I_{b\_dew}(x,y) = 0, \; x \in [1, x_{\max}], y \in [1, y_{\max}] \tag{11}$$

Step 2: Initialization of transformation factors:

$$T_{xy} = \Theta_{xy} = X_{xy} = \text{NULL}, \; x \in [1, x_{\max}], y \in [1, y_{\max}] \tag{12}$$

Step 3: Dewarped binary image calculation:

$$\begin{aligned} &\forall (x,y): W_{ij}(x,y) = 1 \Rightarrow \\ &I_{b\_dew}(x,y') = 1, \\ &\text{where } y' = (x - x_{ij}^{\min}) * \sin(-\theta_{ij}) + y * \cos(\theta_{ij}) + d_{ij} \\ &\text{AND } T_{xy'} = d_{ij} \text{ AND } \Theta_{xy'} = \theta_{ij} \text{ AND } X_{xy'} = x_{ij}^{\min} \end{aligned} \quad (13)$$

where $x_{ij}^{\min}$ is the left side of the bounding box of the word W_{ij} , θ_{ij} is the slope of the word W_{ij} calculated from the upper and lower baseline slopes (see eq. 5) and d_{ij} is defined in eq. 8.

2.4. Final recovery of the warped image

In this step, we proceed to a complete restoration of the original gray scale warped image guided by the draft binary de-warping result of the previous stage. Since the transformation factors for every pixel in the final binary de-warped image have been already stored, the reverse procedure is applied on the gray scale pixels in order to retrieve the final gray scale de-warped image. For all pixels that transformation factors have not been allocated, the transformation factors of the nearest pixel are used.

The following steps are followed:

Step 1: Initialization of dewarped gray scale image:

$$I_{g\_dew}(x,y) = 0 \quad (14)$$

for $x \in [1, x_{\max}], y \in [1, y_{\max}]$

Step 2: Replace NULL values of the transformation factors with the corresponding values of the nearest pixels:

$$\begin{aligned} &\textit{if } I_{b\_dew}(x,y) = 0 \Rightarrow \\ &T_{xy} = T_{x'y'} \text{ AND } \Theta_{xy} = \Theta_{x'y'} \text{ AND } X_{xy} = X_{x'y'} \\ &\text{where } x', y' : \ \underset{\substack{x', y' \\ I_{b\_dew}(x', y') = 1}}{\arg\min} \left(|x - x'| + 2 * |y - y'| \right) \end{aligned} \quad (15)$$

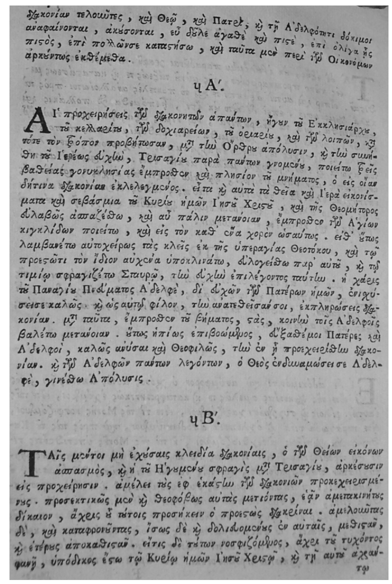
The minimum distance used in order to define the nearest pixel is biased to the x-axis in order to achieve a preference to the pixels belonging to the same text line.

Step 3: Final dewarped gray scale image calculation:

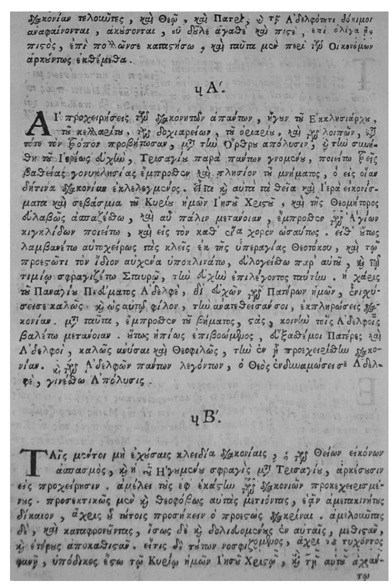
$$\begin{aligned} &I_{g\_dew}(x,y) = I_g(x,y'), \\ &\text{where } y' = \frac{\mathrm{y} - T_{xy} - (x - X_{xy}) sin(-\Theta_{xy})}{\cos(\Theta_{xy})} \end{aligned} \quad (16)$$

3. Experimental results

The proposed algorithm was tested using several arbitrarily warped documents. We mainly focused on historical arbitrarily warped documents as well as on documents with major distortions. Some representative results are shown in Fig. 4-5. As we observed, the experimental results indicate the effectiveness of the proposed technique. Some problems that appear in our experiments are mainly due to erroneous word baseline detection.

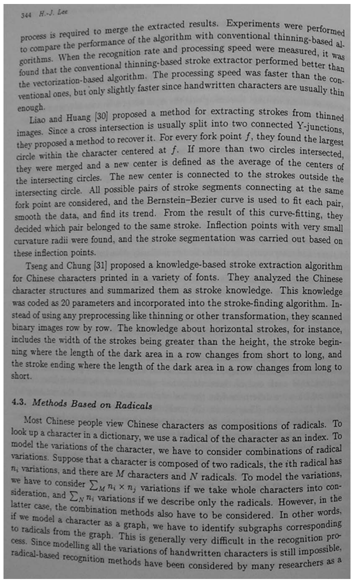


(a)



(b)

Figure 4. Recovery of a historical arbitrarily warped document: (a) original image and (b) de-warped image.



344 H.-J. Lee

process is required to merge the extracted results. Experiments were performed to compare the performance of the algorithm with conventional thinning-based algorithms. When the recognition rate and processing speed were measured, it was found that the conventional thinning-based stroke extractor performed better than the vectorization-based algorithm. The processing speed was faster than the conventional ones, but only slightly faster since handwritten characters are usually thin enough.

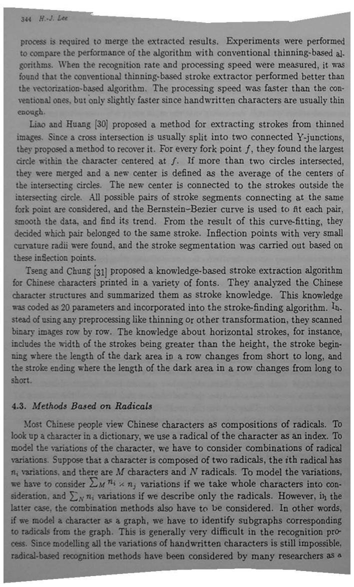
Liao and Huang [30] proposed a method for extracting strokes from thinned images. Since a cross intersection is usually split into two connected Y-junctions, they proposed a method to recover it. For every fork point f, they found the largest circle within the character centered at f. If more than two circles intersected, they were merged and a new center is defined as the average of the centers of the intersecting circles. The new center is connected to the strokes outside the intersecting circle. All possible pairs of stroke segments connecting at the same fork point are considered, and the Bernstein–Bezier curve is used to fit each pair, smooth the data, and find its trend. From the result of this curve-fitting, they decided which pair belonged to the same stroke. Inflection points with very small curvature radii were found, and the stroke segmentation was carried out based on these inflection points.

Tseng and Chung [31] proposed a knowledge-based stroke extraction algorithm for Chinese characters printed in a variety of fonts. They analyzed the Chinese character structures and summarized them as stroke knowledge. This knowledge was coded as 20 parameters and incorporated into the stroke-finding algorithm. Instead of using any preprocessing like thinning or other transformation, they scanned binary images row by row. The knowledge about horizontal strokes, for instance, includes the width of the strokes being greater than the height, the stroke beginning where the length of the dark area in a row changes from short to long, and the stroke ending where the length of the dark area in a row changes from long to short.

4.3. *Methods Based on Radicals*

Most Chinese people view Chinese characters as compositions of radicals. To look up a character in a dictionary, we use a radical of the character as an index. To model the variations of the character, we have to consider combinations of radical variations. Suppose that a character is composed of two radicals, the ith radical has n_i variations, and there are M characters and N radicals. To model the variations, we have to consider $\sum_M n_i \times n_j$ variations if we take whole characters into consideration, and $\sum_N n_i$ variations if we describe only the radicals. However, in the latter case, the combination methods also have to be considered. In other words, if we model a character as a graph, we have to identify subgraphs corresponding to radicals from the graph. This is generally very difficult in the recognition process. Since modelling all the variations of handwritten characters is still impossible, radical-based recognition methods have been considered by many researchers as a

(a)



344 H.-J. Lee

process is required to merge the extracted results. Experiments were performed to compare the performance of the algorithm with conventional thinning-based algorithms. When the recognition rate and processing speed were measured, it was found that the conventional thinning-based stroke extractor performed better than the vectorization-based algorithm. The processing speed was faster than the conventional ones, but only slightly faster since handwritten characters are usually thin enough.

Liao and Huang [30] proposed a method for extracting strokes from thinned images. Since a cross intersection is usually split into two connected Y-junctions, they proposed a method to recover it. For every fork point f, they found the largest circle within the character centered at f. If more than two circles intersected, they were merged and a new center is defined as the average of the centers of the intersecting circles. The new center is connected to the strokes outside the intersecting circle. All possible pairs of stroke segments connecting at the same fork point are considered, and the Bernstein–Bezier curve is used to fit each pair, smooth the data, and find its trend. From the result of this curve-fitting, they decided which pair belonged to the same stroke. Inflection points with very small curvature radii were found, and the stroke segmentation was carried out based on these inflection points.

Tseng and Chung [31] proposed a knowledge-based stroke extraction algorithm for Chinese characters printed in a variety of fonts. They analyzed the Chinese character structures and summarized them as stroke knowledge. This knowledge was coded as 20 parameters and incorporated into the stroke-finding algorithm. Instead of using any preprocessing like thinning or other transformation, they scanned binary images row by row. The knowledge about horizontal strokes, for instance, includes the width of the strokes being greater than the height, the stroke beginning where the length of the dark area in a row changes from short to long, and the stroke ending where the length of the dark area in a row changes from long to short.

4.3. *Methods Based on Radicals*

Most Chinese people view Chinese characters as compositions of radicals. To look up a character in a dictionary, we use a radical of the character as an index. To model the variations of the character, we have to consider combinations of radical variations. Suppose that a character is composed of two radicals, the ith radical has n_i variations, and there are M characters and N radicals. To model the variations, we have to consider $\sum_M n_i \times n_j$ variations if we take whole characters into consideration, and $\sum_N n_i$ variations if we describe only the radicals. However, in the latter case, the combination methods also have to be considered. In other words, if we model a character as a graph, we have to identify subgraphs corresponding to radicals from the graph. This is generally very difficult in the recognition process. Since modelling all the variations of handwritten characters is still impossible, radical-based recognition methods have been considered by many researchers as a

(b)

Figure 5. Recovery of a document with major distortions: (a) original image and (b) de-warped image.

4. Conclusions and future work

In this paper we present a novel segmentation based technique for efficient restoration of arbitrarily warped document images. Our approach is based on (i) text lines and words detection using a novel segmentation technique appropriate for warped documents, (ii) a first draft binary image de-warping based on word rotation and translation according to upper and lower word baselines, and (iii) a recovery of the original warped image guided by the draft binary image de-warping result. The experimental results on several arbitrarily warped documents indicate the effectiveness of the proposed technique. After observing some problems that occurred, we focus our future work to develop a more efficient word baseline detection algorithm.

Acknowledgements

This research is carried out within the framework of the Greek Ministry of Research funded R&D project POLYTIMO [13] which aims to process and provide access to the content of valuable historical books and handwritten manuscripts.

References

[1] O. Lavaille, X. Molines, F. Angella & P. Baylou, "Active Contours Network to Straighten Distorted Text Lines", Proc. Int. Conf. Image Processing, 2001, pp. 1074-1077.

[2] H. Ezaki, S. Uchida, A. Asano & H. Sakoe, "Dewarping of document image by global optimization", Proc. ICDAR'00, 2005, pp. 500-506.

[3] Z. Zhang & C. L. Tan, "Correcting document image warping based on regression of curved text lines", Proc. ICDAR'03, 2003, pp. 589–593.

[4] C. Wu & G. Agam, "Document image de-warping for text/graphics recognition", SSPR&SPR 2002, LNCS 2396, 2002, pp. 348-357.

[5] A. Ulges, C.H. Lampert & T.M. Breuel, "Document image dewarping using robust estimation of curled text lines", Proc. ICDAR'05, 2005, pp. 1001- 1005.

[6] C.L. Tan, L. Zhang, Z. Zhang & T. Xia, "Restoring Warped Document Images through 3D Shape Modeling", IEEE Trans. Pattern Analysis and Machine Intelligence 28(2), 2006, pp. 195-208.

[7] M.S. Brown & W.B. Seales, "Image Restoration of Arbitrarily Warped Documents", IEEE Trans. Patt. Analysis and Machine Intelligence, 26(10), 2004, pp. 1295-1306.

[8] H. Cao, X. Ding & C. Liu, "Rectifying the Bound Document Image Captured by the Camera: A Model Based Approach", Proc. Int. Conf. Comp. Vision, 2003, pp. 71-74.

[9] B. Gatos, I. Pratikakis & S.J. Perantonis, "Adaptive Degraded Document Image Binarization", Pattern Recognition, 39, 2006, pp. 317-327.

[10] L. Shapiro & G. Stockman. Computer Vision (Prentice Hall, 2001)

[11] F.M. Wahl, K.Y Wong & R.G. Casey, "Block Segmentation and Text Extraction in Mixed Text/Image Documents", Computer Graphics and Image Processing, 20, 2006, pp. 375-390.

[12] U.V. Marti & H. Bunke, "Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system", Int. Journal of Pattern Recognition and Artifical Intelligence, 15(1), 2001, pp. 65–90.

[13] POLYTIMO project, http://iit.demokritos.gr/cil/Polytimo, 2007.