

A Two-Step Dewarping of Camera Document Images

N. Stamatopoulos, B. Gatos, I. Pratikakis and S. J. Perantonis

*Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,
National Center for Scientific Research "Demokritos", GR-153 10 Athens, Greece*

<http://www.iit.demokritos.gr/cil/>

{nstam, bgat, ipratika, sper}@iit.demokritos.gr

Abstract

Dewarping of camera document images has attracted a lot of interest over the last few years since warping not only reduces the document readability but also affects the accuracy of an OCR application. In this paper, a two-step approach for efficient dewarping of camera document images is presented. At a first step, a coarse dewarping is accomplished with the help of a transformation model which maps the projection of a curved surface to a 2D rectangular area. The projection of the curved surface is delimited by the two curved lines which fit the top and bottom text lines along with the two straight lines which fit to the left and right text boundaries. At a second step, fine dewarping is achieved based on words detection. All words are pose normalized guided by the lower and upper word baselines. Experimental results on several camera document images demonstrate the robustness and effectiveness of the proposed technique.

1. Introduction

Non-linear warping is often observed in document images of bounded volumes, when captured by camera or flatbed scanning. Text in such cases is strongly distorted and influences the performance of further processing, since the contemporary OCR systems cannot handle distorted text.

Many different approaches have been proposed for document image dewarping [1-2]. These approaches can be classified into two broad categories based on (i) 3-D document shape reconstruction [3-8] and (ii) 2-D document image processing [9-20]. Approaches of the first category require specialized hardware (stereo-cameras, structured light sources, laser camera) or prior metric knowledge or they rely on restricted models to obtain the 3-D shape of the page. On the other hand, approaches in the second category use only

2-D information from camera document images. Our work is related to the second category. Previous approaches of the second category are described in the following:

Masalovitch and Mestetskiy [9] propose a method for document dewarping using the outer skeleton of text images. Long continuous branches, which define interlinear spaces of the document, are approximated by cubic Bezier curves in order to find a specific deformation of each interlinear space and then a whole approximation of the document is built. This method is sensitive to the approximation of deformation of vertical borders of text blocks which isn't so accurate. In [10], Lavialle *et al.* propose a method to straighten distorted text lines using an active contour network based on an analytical model with cubic B-splines which have been proved more accurate than Bezier curves. However, the initialization must be closed to the desired result. Zhang and Tan [11] divide the document image into shaded and non-shaded region and then use polynomial regression to model the warped text lines with quadratic reference curves. They find the text line curves by clustering connecting components and move the components to restore straight horizontal baselines. Images must be always grayscale and have a shaded region. Also, another model fitting technique [12] uses cubic splines to define the warping model of the document image. Approximate each line shape using some characteristic points of line's black objects. For more accurate dewarping, a vertical division of a document image into some partial document images is also suggested, but they did not show satisfied dewarping results in their paper. Main disadvantage of this approach is that it is hard to define such characteristics points of black objects that can give a stable approximation of the line shape. Wu and Agam [13] use the texture of a document image 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. In [14], they rely on a priori layout information and apply a line-by-line dewarping of the observed paper surface. Their method is based on the assumption that the original page contained only straight lines that were approximately equally spaced and sized and also there aren't large spaces between words. This assumption is not generic. Lu and Tan [15] approach is based on the image partition, which is implemented based on the identified VSBs as well as the x-line and the baseline of text lines. This approach fails when the distortion angle is big. Also, another segmentation based method [16] uses a novel segmentation technique to detect text lines but it is prone to segmentation errors. In another work [17], Liang *et al.* model the page surface by a developable surface, and exploit the properties of the printed textual content on the page to recover the surface shape. This process is quite slow because the restoration involves the texture flow computation and the developable surface estimation. In [18] and [19], they rely on boundaries to delimit the required dewarping and they use 2D natural cubic splines to model boundaries. Then, they apply particular types of interpolation, bi-linear blended Coons and Gordon patch model, respectively. Specifically, in [18] they use a pattern to find the boundary and the approach in [19] is highly probable to fail in the dewarping process since it deals with text line detection at the original warped document which is a well-known hard task. Finally in [20], Wu *et al.* assume that the image surface is a cylinder and generate a transformation to flatten the document image. Main disadvantage of this method is that it uses complex computation and therefore is time-consuming, while the assumption that a single cylinder fits to a deformed page is not realistic.

In this paper, we propose a two-step approach for efficient dewarping of camera document images, which is directly applied to the 2D space, and uses a segmentation technique that is fed by a coarse dewarping input document which reduce erroneous segmentation results and consequently improves the final OCR. At a first step, we apply a transformation model which maps the projection of a curved surface to a 2D rectangular area in order to achieve a coarse dewarping. At a second step, fine dewarping is achieved when all words are pose normalized guided by the lower and upper word baselines. Experimental results on several camera documents demonstrate the effectiveness and robustness of the proposed technique. The remainder of the paper is organized as

follows. In section 2, the proposed technique is detailed while experimental results are discussed in Section 3. Finally, conclusions are drawn in Section 4.

2. Proposed method

In our approach, we apply two main steps (coarse and fine dewarping) in order to recover camera document images which are detailed in the following sections. Before we proceed to the above steps, we apply an image binarization using the technique proposed in [21] and then we remove black and text borders based on the approach described in [22].

2.1. Coarse dewarping

In this step, we apply a simple and fast transformation model which maps the projection of a curved surface to a 2D rectangular area in order to achieve a coarse dewarping of the document image. At first, we detect text lines based on [16]. An error at text line detection won't influence the coarse dewarping. Let L_c denotes the number of lines and $LettH$ denotes the average character height. Following that, we extract the projection of the curved surface, which is defined by the two curved lines which fit the top text line and the bottom text line along with the two lines which fit to the left and right text boundaries (see Fig. 1), and then we apply the transformation model.

2.1.1. Extraction of curved surface projection

Once the text lines have been detected, we estimate the straight lines that fit to the left and right text boundaries as well as the curved lines which fit the top text line and the bottom text line. Let $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$, $D(x_4, y_4)$ denote the corner points of the projection of the curved surface (see Fig. 1).

Step A.1: Left/Right Straight Line Segment Estimation

The distinct steps we follow in order to estimate the straight line AD which corresponds to left text boundary are as follows:

Step A.1.1: Detect all the leftmost points in each text line: (x_{s_i}, y_{s_i}) , $0 < i \leq L_c$

Step A.1.2: Calculate the average value X_s of all x_{s_i} and every point (x_{s_i}, y_{s_i}) is excluded if $|x_{s_i} - X_s| > 2LettH$ in order to eliminate the leftmost

points of text lines which don't start from the beginning of the document, such as titles and subtitles, and therefore we have a better estimation of the straight line AD .

Step A.3: Least Square Estimation (LSE) method is used to get the straight line AD that fits all points (x_{s_i}, y_{s_i}) that remain after *Step A.2*.

After this process, the straight line AD , which corresponds to the left text boundary, is defined as:

$$y = a_l x + b_l \quad (1)$$

Similarly, the straight line BC , which corresponds to the right text boundary, is defined as:

$$y = a_r x + b_r \quad (2)$$

Step B: Top/Bottom Curved Line Segment Estimation

The distinct steps we follow in order to estimate the curved line AB , are as follows:

Step B.1: At this step we choose the proper text line which will be used to estimate the curved line AB . Let Dl_i represent the distance between the leftmost point of text line i and the straight line AD , also let Dr_i represent the distance between the rightmost point of text line i and the straight line BC . We choose the first text line which satisfies the following criterion:

$$(Dl_i + Dr_i) / 2 < 2LetH \quad (3)$$

This criterion guarantees that the text line selected is not too small, not a title, subtitle etc.

Step B.2: Detect all the upper points (x_{u_i}, y_{u_i}) of the text line defined in previous step.

Step B.3: Least Square Estimation (LSE) method is used to find the coefficients of a polynomial curve of third degree that fits all points of *Step B.2*.

After this process, the curved line AB is defined as:

$$y = a_{u1}x^3 + a_{u2}x^2 + a_{u3}x + a_{u4} \quad (4)$$

Similarly, the curved line DC is defined as:

$$y = a_{l1}x^3 + a_{l2}x^2 + a_{l3}x + a_{l4} \quad (5)$$

The result of extraction of curved surface projection is illustrated in Figure 1.

2.1.2. Transformation model

Our goal is to generate a transformation that maps to a 2D rectangular area the projection of the curved surface, which is delimited by the curved line segments AB , DC and the straight line segments AD , BC . Let $A'(x'_1, y'_1)$, $B'(x'_2, y'_2)$, $C'(x'_3, y'_3)$, $D'(x'_4, y'_4)$ represent

the corner points of the rectangular area (see Fig. 2). Also, let $|\widehat{AB}|$ represent the arc length between points A and B and $|AB|$ represent the Euclidean distance between points A and B .

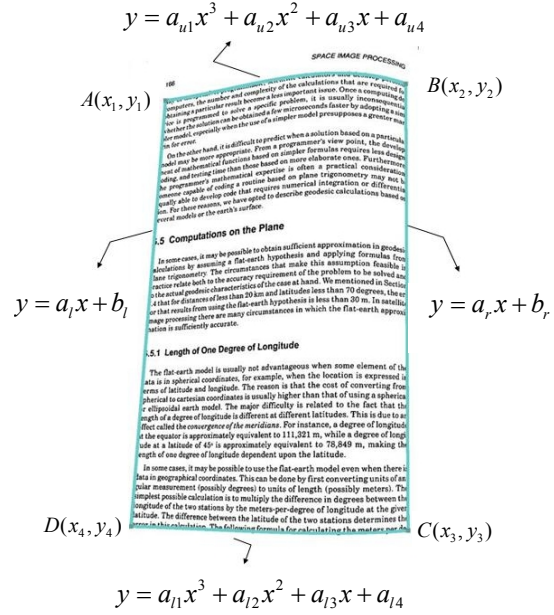


Figure 1: Extraction of curved surface projection

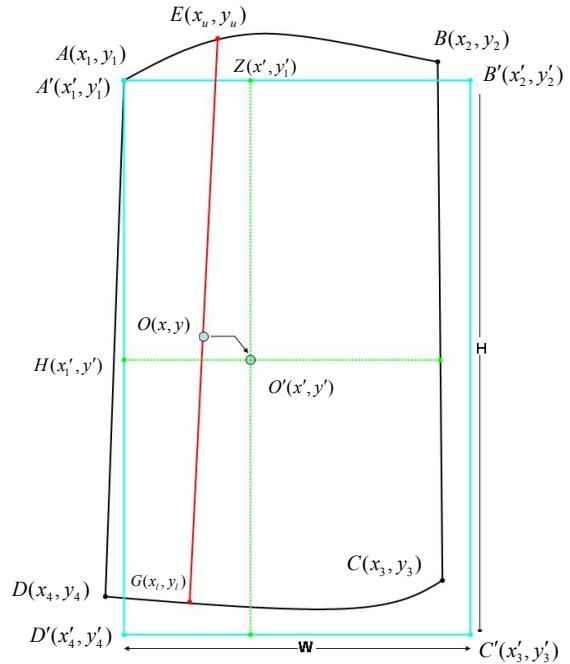


Figure 2: Transformation model

Step A: At this step we allocate the rectangular area $A'B'C'D'$:

The width W of the rectangular area is calculated as follows:

$$W = \min(|\widehat{AB}|, |\widehat{DC}|) \quad (6)$$

and the height H of the rectangular area is calculated as follows:

$$H = \min(|AD|, |BC|) \quad (7)$$

In the algorithm that follows we examine the case that $W = |\widehat{AB}|$ and $H = |AD|$ (see Fig. 2) (a similar process is applied in other cases).

The corner points of the rectangular area are calculated as follows:

$$\left. \begin{array}{l} x'_1 = x_1, \quad y'_1 = y_1 \\ x'_2 = x'_1 + W, \quad y'_2 = y'_1 \\ x'_3 = x'_2, \quad y'_3 = y'_2 + H \\ x'_4 = x'_1, \quad y'_4 = y'_3 \end{array} \right\} \quad (8)$$

Step B: Create a correspondence between the points of curved line segments AB and DC , expressed by a function \mathcal{F} defined as follows:

$$\mathcal{F}(E(x_u, y_u)) = G(x_l, y_l) \text{ if } \frac{|\widehat{AE}|}{|\widehat{AB}|} = \frac{|\widehat{DG}|}{|\widehat{DC}|} \quad (9)$$

where $E(x_u, y_u)$ represent a point on curved line segment AB and $G(x_l, y_l)$ represent a point on curved line segment DC .

Step C: Let $O(x, y)$ represent a point in the projection of the curved surface. Our goal is to calculate the new position $O'(x', y')$ of $O(x, y)$ in the rectangular area (see Fig. 2).

Firstly, we define the line EG which satisfies the following criteria:

- (1) Crosses the points $E(x_u, y_u)$ and $G(x_l, y_l)$, which belong to the curved line segments AB and DC respectively.
- (2) $\mathcal{F}(E(x_u, y_u)) = G(x_l, y_l)$
- (3) The point $O(x, y)$ belongs to the line EG .

Then, we calculate the position of $O'(x', y')$ as follows:

$$x' = x'_1 + |A'Z| \quad (10)$$

$$y' = y'_1 + |A'H| \quad (11)$$

where H is point $H(x'_1, y')$, and Z is point $Z(x', y'_1)$ and $|A'Z|$, $|A'H|$ are calculated as follows:

$$\frac{|\widehat{AB}|}{|\widehat{AE}|} = \frac{W}{|A'Z|} \Rightarrow |A'Z| = \frac{W}{|\widehat{AB}|} |\widehat{AE}| \quad (12)$$

$$\frac{|EG|}{|EO|} = \frac{H}{|A'H|} \Rightarrow |A'H| = \frac{H}{|EG|} |EO| \quad (13)$$

Consequently, we repeat *Step C* for all points which are inside the projection area of the curved surface.

Step D: Finally, all the points which are out of the projection area if the curved surface inherit the transformation of the nearest point.

Figure 3 depict three examples of coarse dewarping step. In our method the document should contain sufficient text content in order to extract the projection of the curved surface, but, as it can be observed, it can deal with documents which contain non-text content, like images, graphs, etc.

2.2. Fine dewarping

In this step, fine rectification is obtained on the word level and it is based on the algorithm described in [16]. This technique is based on text line and word segmentation. Then, all words are pose normalized guided by the lower and upper word baselines. The distinct steps followed are explained in the following sections.

2.2.1. Text line and word detection

Text line detection is applied at the resulting dewarped image after the coarse dewarping stage, thus having very high probability for a successful result.

At first, we remove the non-text components which satisfy one of the following conditions:

$$\begin{aligned} &Height > 3 * LettH \text{ or } Height < LettH / 4 \\ &\text{or } Width < LettH / 4 \end{aligned} \quad (14)$$

where *Height* denotes component's height and *Width* denotes component's width. Then, we detect text lines and words. All words are detected using a proper image smoothing. Following that, horizontally neighboring words are consecutively linked in order to detect text lines.

2.2.2. 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 using the methodology given in [23]. After this procedure, upper baseline of word W_{ij} is defined as:

$$y = a_{ij}x + b_{ij} \quad (15)$$

Next, the system randomly generates a list of confidence values δ and a fuzzy set value for each active user. Each confidence value is represented by a fuzzy term, which is an integer in the range of [0, 7]. Finally, the system populates the preference values to δ_0 by aggregating δ and δ_0 using Equation (6).

To simulate imperfect user feedback during the GA, all perfect knowledge will be turned in a noisy process. Based on the preference values in the imperfect knowledge, the system then randomly selects a set of items and assigned feature values as the user navigation behaviors. These feature values are generated by a reverse procedure of Equation (11). For example, the higher the preference value of an item is, the longer periods of view time is assigned to this item. Finally, the system recalibrates the user profile by aggregating K imperfect confidence values of navigation pattern clusters and a fuzzy set value to the imperfect user profile.

5.2 Experimental Results

We conducted several sets of experiments to verify our system performance and to compare the results of different GA parameter settings. In these experiments, we observed a significant margin of improvement after incorporating the GA in matching the user expectations in various settings. It is also shown that the performance improvement of our learning mechanism is independent of the number of users. Moreover, the improvement is linearly increased with the number of items. However, due to the space limit, in this paper, we only stress the improvements achieved by applying our learning mechanism.

The results shown for each set of experiments are averaged over many runs, where each run is executed with different seeds for the random generator functions. The parameter settings of GA operators [20, 17] are population size=30, one-point crossover, tournament selection, keep the elitism, and mutation rate = 0.25. The benchmark settings of the following figure, i.e., the number of items, the number of cut-off point, the number of fuzzy terms, the number of experts, and the number of navigation pattern clusters, are fixed as $M = 5000$, $N = 100$, $F = 8$, $E = 50$, $K = 10$, and $P = 5\%$, respectively.

Figure 2 demonstrates the improvements achieved by our learning mechanism. The X-axis of Figure 2 depicts the number of generations. The Y-axis of Figure 2a illustrates the similarity distance (between query results and perfect user feedback) computed by Equation (10). The Y-axis of Figure 2b represents average satisfaction computed by Equation (11), respectively. In this experiment, Figure 2 indicates that the accuracy of web-list improves drastically after incorporating the learning mechanism. As observed, the average satisfaction increases nearly 50% within ten generations of learning. Overall, in the majority of our experiments, GA can acquire nearly ideal user profiles, i.e., the rankings in the retrieved web-list has 90% similarity to those in the perfect web-list, within 40 generations. It should be noted that Yoh

Next, the system randomly generates a list of confidence values δ and a fuzzy set value for each active user. Each confidence value is represented by a fuzzy term, which is an integer in the range of [0, 7]. Finally, the system populates the preference values to δ_0 by aggregating δ and δ_0 using Equation (6).

To simulate imperfect user feedback during the GA, all perfect knowledge will be turned in a noisy process. Based on the preference values in the imperfect knowledge, the system then randomly selects a set of items and assigned feature values as the user navigation behaviors. These feature values are generated by a reverse procedure of Equation (11). For example, the higher the preference value of an item is, the longer periods of view time is assigned to this item. Finally, the system recalibrates the user profile by aggregating K imperfect confidence values of navigation pattern clusters and a fuzzy set value to the imperfect user profile.

5.2 Experimental Results

We conducted several sets of experiments to verify our system performance and to compare the results of different GA parameter settings. In these experiments, we observed a significant margin of improvement after incorporating the GA in matching the user expectations in various settings. It is also shown that the performance improvement of our learning mechanism is independent of the number of users. Moreover, the improvement is linearly increased with the number of items. However, due to the space limit, in this paper, we only stress the improvements achieved by applying our learning mechanism.

The results shown for each set of experiments are averaged over many runs, where each run is executed with different seeds for the random generator functions. The parameter settings of GA operators [20, 17] are population size=30, one-point crossover, tournament selection, keep the elitism, and mutation rate = 0.25. The benchmark settings of the following figure, i.e., the number of items, the number of cut-off point, the number of fuzzy terms, the number of experts, and the number of navigation pattern clusters, are fixed as $M = 5000$, $N = 100$, $F = 8$, $E = 50$, $K = 10$, and $P = 5\%$, respectively.

Figure 2 demonstrates the improvements achieved by our learning mechanism. The X-axis of Figure 2 depicts the number of generations. The Y-axis of Figure 2a illustrates the similarity distance (between query results and perfect user feedback) computed by Equation (10). The Y-axis of Figure 2b represents average satisfaction computed by Equation (11), respectively. In this experiment, Figure 2 indicates that the accuracy of web-list improves drastically after incorporating the learning mechanism. As observed, the average satisfaction increases nearly 50% within ten generations of learning. Overall, in the majority of our experiments, GA can acquire nearly ideal user profiles, i.e., the rankings in the retrieved web-list has 90% similarity to those in the perfect web-list, within 40 generations. It should be noted that Yoh

Figure 4: Recovery of a warped document: (a) Original Image; (b) after coarse dewarping step; (c) after fine dewarping step; (d),(e),(f) enlarged image positions of (a),(b),(c), respectively

As a measure of success, we evaluate the dewarping results by carrying out OCR testing on original and restored images. We perform the OCR test using ABBYY FineReader Pro 8.0 [25]. Table 1 illustrates the average OCR rate before and after dewarping using SET-1 and Table 2 illustrates the average OCR rate before and after dewarping using SET-2. Our experimental results demonstrate the success of our dewarping method, since it has achieved remarkable improvements on OCR accuracy. Also, our approach takes approximately 10 seconds on average to process one page. Some representative results are shown in Figures 5, 6 and 7.

Table 1: OCR rate using SET-1

Without Dewarping	78.20%
After Dewarping using method [16]	92.12%
After Dewarping using the proposed method	98.52%

Table 2: OCR rate using SET-2

Without Dewarping	76.69%
After Dewarping using method [16]	90.41%
After Dewarping using method [14],[24]	86.07%
After Dewarping using the proposed method	98.53%

4. Conclusion

In this paper, we propose a two-step approach for efficient dewarping of camera document images which suffer from warping due to distortion and surface deformation. At a first step, we apply a transformation model which maps the projection of a curved surface into a 2D rectangular area in order to achieve a coarse dewarping of the document image. At a second step, fine dewarping is achieved based on text line and word segmentation. Our experimental results show that the proposed approach can dewarp the document images well and improve the method proposed in [16], as the recognition performance of the restored images is highly improved and the average recognition rate reaches over 98%.

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 215064 (project IMPACT).

5. References

- [1] J. Liang, D. Doermann, and H. Li. "Camera-based analysis of text and documents: a survey" *International Journal on Document Analysis and Recognition*, 7(2-3), 2005, pp. 84–104.
- [2] F. Shafait, T. M. Breuel, "Document Image Dewarping Contest", *In 2nd Int. Workshop on Camera-Based Document Analysis and Recognition*, Curitiba, Brazil, 2007, pp. 181-188.
- [3] A. Yamashita, A. Kawarago, T. Kaneko, K. T. Miura, "Shape Reconstruction and Image Restoration for Non-Flat Surfaces of Documents with a Stereo Vision System", *International Conference on Pattern Recognition*, vol. 1, Cambridge, UK, 2004, pp. 482-485.
- [4] M.S. Brown & W.B. Seales, "Image Restoration of Arbitrarily Warped Documents", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(10), 2004, pp. 1295-1306.
- [5] C.L. Tan, L. Zhang, Z. Zhang & T. Xia, "Restoring Warped Document Images through 3D Shape Modeling", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(2), 2006, pp. 195-208.
- [6] A. Ulges, C. H. Lampert, and T. Breuel, "Document capture using stereo vision" *ACM Symposium on Document Engineering*, Milwaukee, Wisconsin, USA, October 28-30, 2004, pp. 198–200.
- [7] N. Gumerov, A. Zandifar, R. Duraiswarni, and L. S. Davis, "Structure of applicable surfaces from single views" *European Conference on Computer Vision*, Prague, 2004, pp. 482–496.
- [8] H. Cao, X. Ding and C. Liu, "A cylindrical surface model to rectify the bound document image" *International Conference on Computer Vision*, vol. 1, Nice, France, 2003, pp. 228.
- [9] A. Masalovitch and L. Mestetskiy, "Usage of continuous skeletal image representation for document images dewarping" *In 2nd Int. Workshop on Camera-Based Document Analysis and Recognition*, Curitiba, Brazil, 2007, pp. 45-53.
- [10] O. Lavaille, X. Molines, F. Angella and P. Baylou, "Active Contours Network to Straighten Distorted Text Lines" *International Conference on Image Processing*, Thessaloniki, Greece, 2001, pp. 1074-1077.
- [11] Z. Zhang & C. L. Tan, "Correcting document image warping based on regression of curved text lines" *International Conference on Document Analysis and Recognition*, Edinburgh, Scotland, 2003, pp. 589–593.
- [12] H. Ezaki, S. Uchida, A. Asano & H. Sakoe, "Dewarping of document image by global optimization" *International Conference on Document Analysis and Recognition*, Seoul, Korea, 2005, pp. 500-506.
- [13] C. Wu & G. Agam, "Document image De-warping for text/graphics recognition" *Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, Windsor, Canada, 2002, pp. 348-357.
- [14] A. Ulges, C.H. Lampert & T.M. Breuel, "Document image dewarping using robust estimation of curled text lines" *International Conference on Document Analysis and Recognition*, Seoul, Korea, 2005, pp. 1001- 1005.
- [15] Shijian Lu, Chew Lim Tan, "The Restoration of Camera Documents through Image Segmentation", *Workshop on Document Analysis Systems VIII*, Nelson, New Zealand, 2006, pp. 484-495.
- [16] B. Gatos, I. Pratikakis and I. Ntirogiannis, "Segmentation Based Recovery of Arbitrarily Warped Document Images" *International Conference on Document Analysis and Recognition*, Curitiba, Brazil, 2007, pp. 989-993.
- [17] J. Liang, D. DeMenthon, and D. Doermann, "Flattening curved documents in images" *Conference on Computer Vision and Pattern Recognition*, San Diego, USA, 2005, pp. 338–345.
- [18] Y. C. Tsoi and M. S. Brown, "Geometric and shading correction for images of printed materials—A unified approach using boundary" *Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2004, pp. 240-246
- [19] Z. Zhang & C. L. Tan, "Warped image restoration with applications to digital libraries", *International Conference on Document Analysis and Recognition*, Seoul, Korea, 2005, pp. 192-196
- [20] M. Wu, R. Li, B. Fu, W. Li and Z. Xu, "A Page Content Independent Book Dewarping Method to Handle 2D Images Captured by a Digital Camera", *International Conference on Image Analysis and Recognition*, Montreal, Canada, 2007, pp. 1242-1253.
- [21] B. Gatos, I. Pratikakis & S.J. Perantonis, "Adaptive Degraded Document Image Binarization", *Pattern Recognition*, 39, 2006, pp. 317-327.
- [22] N. Stamatopoulos, B. Gatos, and A. Kesidis, "Automatic Borders Detection of Camera Document Images" *In 2nd Int. Workshop on Camera-Based Document Analysis and Recognition*, Curitiba, Brazil, 2007, pp. 71-78.
- [23] U. V. Marti and H. Bunke, "Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system", *International journal of Pattern Recognition and Artificial intelligence*, 15(1), 2001, pp. 65–90.
- [24] <http://quito.informatik.uni-kl.de/dewarp/dewarp.php>
- [25] ABBYY FineReader OCR: <http://finereader.abbyy.com/>