A Model-based Book Dewarping Method Using Text Line Detection

Bin Fu, Minghui Wu, Rongfeng Li, Wenxin Li, Zhuoqun Xu, Chunxu Yang

State Key Laboratory on Machine Perception
School of Electronics Engineering and Computer Science
Peking University
Beijing, China
{ fubinpku, wuminghui, rongfeng, lwx, zqxu }@pku.edu.cn

Abstract

In this paper, we propose a book dewarping model based on the assumption that the book surface is warped as a cylinder. This model extends the model proposed by Cao and makes Cao's model a special case of our model. This extension removes the constraint of Cao's model that the camera lens must be strictly parallel to the book surface, which is hard to make in practice, therefore enables a user to take a picture from different point of views conveniently. The main idea of the model is to build up the correspondence between a rectangle region on a flat surface and its curved region on the distorted book image and the dewarping task is to flatten the curved region to its original rectangle shape. The experimental results demonstrate the effectiveness of our proposed book dewarping approach.

1. Introduction

A number of camera document image restoration techniques have been reported in the literature. According to whether auxiliary hardware is required, the proposed techniques can be classified into two different categories. Approaches [1-2] with auxiliary hardware can solve the dewarping problem well; however, the costly equipment makes the "hard" approaches unattractive. Approaches without auxiliary hardware can be further classified into two classes according to the problem it is oriented. Approaches [3-4] focus on the problem of removing the perspective distortion in images of flat documents. Approaches [5 - 10] focus on a more complex situation: Page warping adds a non-linear, non-parametric distortion on the perspective document image. In [5], Ulges propose a line-by-line dewarping approach and In [6], Lu uses a document image segmentation algorithm. Both the two methods are sensitive to the resolution and language in a document image. In [7], Cao models the book surface as a cylinder, and requires the camera lens parallel to the generatrix of the book surface. This constraint makes the approach inconvenient in practice. In [8], Liang makes use of the concept of developable surface to model the warped surface, and use text information to estimate the surface, although this approach can handle complex warping situation, such as page curling at the corner, it is time consuming and needs to get the text lines precisely. In [9], Ezaki uses a group of cubic splines to model the warped text lines and proposes a global optimization algorithm to find the proper cubic splines, their approach is novel and can lessen the curl degree, but they did not show satisfied rectification results in their paper.

In this paper, we propose a transform model to stretch a cylinder surface into a flat one. This model is further applied to solve the problem of restoring document images, which suffer from perspective and page curling distortion. Comparing with Cao's approach [7], our model does not require the camera lens to be parallel to the book page surface's generatrix; therefore it imposes fewer restrictions on the input image, so it is more widely applicable.

The rest of this paper is organized into four sections. Section 2 introduces our dewarping model. Section 3 focuses on the rectification process. We present our experimental results in Section 4, and Section 5 concludes the paper.

2. The Transform Model

Assume we have a rectanglar area on a page surface as shown in Figure 1. The existence of bookbinding often causes the distortion of the book image. The projection of the book image is shown in Figure 2. Figure 3 is the distorted image of the book with its rectangle area.

Before the transformation, we make the following assumptions:

(1) The borders of the rectangle are paralleled to the borders of the page.

(2) The page surface is warped as a cylinder.

(3) A cylinder is a surface generated by a straight line intersecting and moving along a closed planar curve. The straight line is called the generatrix, and the closed planar curve is called the directrix.

(4)The left and right borders of the page are parallel to the generatrix of the cylinder.

(5) Because the left and right borders of the rectangle are parallel to the left and right borders of the page, they are therefore parallel to the generatrix. In this case, the images of the left and right borders of the rectangle are straight line segments.

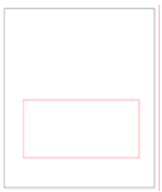


Figure 1. A rectangle area on a page surface

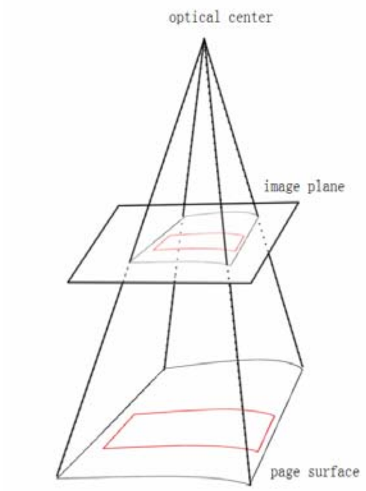


Figure2. The projection of the page surfac**e**

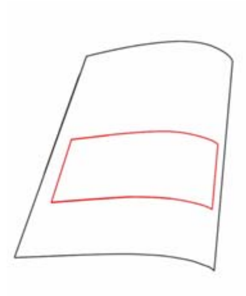


Figure 3. The distorted image of the page surface with its rectangle area

Our goal is to generate a transformation to flatten the image in Figure 3 to its original image in Figure 1. The transformation is a mapping from the curved coordinate system to a Cartesian coordinate system. Once curved coordinate net is set up on the distortion image in Figure 4, the transformation can be done in two steps: First, the curve net is stretched to a straight one (Figure 5) and then adjusted to a well-proportioned square net (Figure 6).

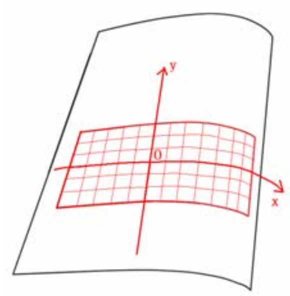


Figure 4. An example of a warp document page with its curved coordinate net

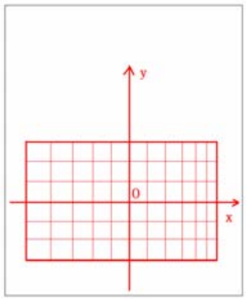


Figure 5. The square net generated by stretching the curved coordinate net in Figure 4.

2.1 Stretched the Curve Coordinate Net

In this step, our goal is to transform a picture like Figure 4 into a picture like Figure 5. We may also view the transform process from the other direction. That is, given the matrix on Figure 5, calculate each

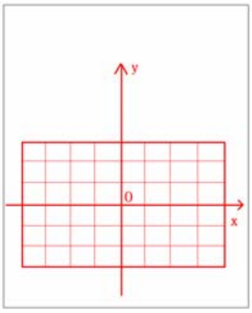


Figure 6. The final result square net generated by adjusting the width in Figure 5 according to the curvature of the page surface.

cell's correspondence point on Figure 4 and fill each cell with the content of its correspondence point. We fill the matrix of Figure 5 in two steps: 1) let the page lower boundary in Figure 4 map with the bottom line in Figure 5; 2) one by one line up the page lower boundary point in Figure 4 and the vanish point (transform center). The line intersects the page upper boundary in Figure 4 at a point and then we get a line segment on Figure 4. Map this line segment with the correspondence column in Figure 5. In step 2, we cannot evenly distribute the points on the line segment in Figure 4 to match with the column in Figure 5 due to their different distances to the lens. In the following section, we explain how to match the line segment in Figure 4 to a column in Figure 5.

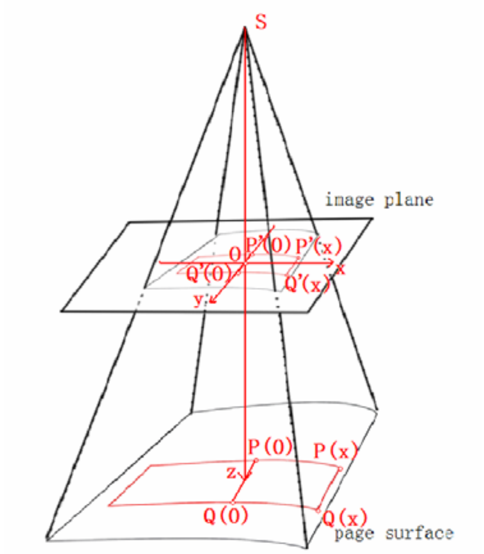


Figure 7. The details of the projection between an arbitrary line segment P'(x)Q'(x) which across the transform center and its inversimage P(x)Q(x)

In Figure 7. P'(x)Q'(x) is a line segment (its x coordinate is x) which along the y coordinate in the image plane and P(x)Q(x) is its inverse image on the page surface. S is the lens. O is the intersection of the optical axis and the image plane. Here the image plane may not be parallel to the page surface due to the variation of direction of the lens. Figure 8 shows the yOz plane of Figure 7. The z axis meets the page surface at R(x). I(x) and H(x) are the foot points of Q(x) and P(x) on the z axis.

By geometry, we get the following equations:

$$\frac{SO}{SH(x)} = \frac{P'(x)O}{P(x)H(x)}, \quad \frac{SO}{SI(x)} = \frac{Q'(x)O}{Q(x)I(x)},$$

$$P'(x)O = \frac{P(x)H(x)\cdot SO}{SH(x)} = \frac{P(x)R(x)\sin\angle P(x)R(x)H(x)\cdot SO}{SR(x)+R(x)H(x)} \quad (1)$$

$$= \frac{SO\cdot P(x)R(x)\sin\angle P(x)R(x)H(x)}{SR(x)+\cos\angle P(x)R(x)H(x)\cdot P(x)R(x)}$$

Let $\alpha = \angle P(x)R(x)H(x), f = SO, d(x) = SR(x),$, we get

$$y'_P(x) = P'(x)O, y_P(x) = P(x)R(x)$$

$$y'_P(x) = \frac{f y_P(x)\sin\alpha}{d(x)+y_P(x)\cos\alpha} \quad (2)$$

Let $a = \frac{d(x)}{f\sin\alpha}, b = \frac{\cos\alpha}{f\sin\alpha}$,

$y_P = y_P(x), y'_P(x) = y'_P$, then

$$y'_P = \frac{y_P}{a+by_P} \quad (3)$$

Analogously, let $y'_Q = Q'(x)O, y_Q = Q(x)R(x)$, we have the equation:

$$y'_Q = \frac{y_Q}{a+by_Q} \quad (4)$$

Now we consider $P_0, P_1, \ldots, P_n$ as a series of equidistant points on P(x)Q(x), which have the y coordinates: $y_i = \frac{n-i}{n}y_P + \frac{i}{n}y_Q$, we can get the y coordinates of the image points $P_0', P_1', \ldots, P_n'$:

$$y'_i = \frac{y_i}{a+by_i} \quad (5)$$

y'_P, y'_Q are the y coordinates of the points P(x) and Q(x) which are on the edge of the page. They can be detected by page extraction in the next section. In practice, we let $y_P = y_Q = k$, where k is a constant value obtained by experience. Thus, with the knowledge of y'_P, y'_Q and y_P, y_Q, we can calculate the parameters *a* and *b* with the equations (3) and (4).

Using (5), the line segment P'Q' is divided into n pieces. Dividing all the line segment along the y coordinate, the coordinate net in Figure2 is set up. Thus, the coordinate net in Figure3 is generated by a one-to-one mapping.

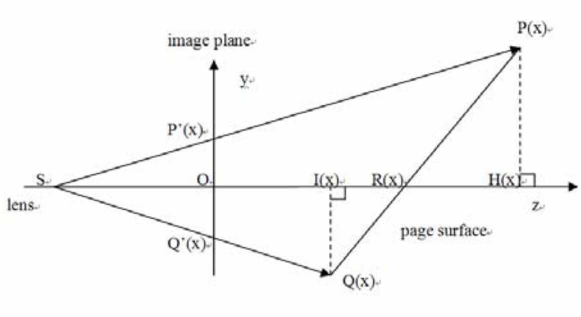


Figure 8. The yOz plane of figure 5

2.2 Adjust the Width of the Square Net

Consider the surface of the page as a cylinder whose curve equation is *b(x)* (Figure 9). In order to adjust the *x* coordinate to become well-proportioned, for each point we adjust its x coordinate x_{adjust} into its curve distance *j(x)* from the book spine.

$$x_{adjust} = j(x) = \int_0^x \sqrt{(db(x))^2 + (dx)^2} = \int_0^x \sqrt{1 + b'(x)^2}\,dx \qquad (6)$$

In Figure 7, P(0)Q(0) is the book spine, their x coordinate is 0. P'(0), Q'(0) is the image of P(0),Q(0). We can get b(x) by the y coordinate of P'(x),P'(0) (that is $y'_P(x), y'_P(0)$). Figure 10 shows the yOz plane of Figure 7.

In Figure 10,

$$SR(x) = SR(0) - R(0)R(x) = SR(0) - \frac{b(x)}{\sin\angle P(0)R(0)R(x)} \qquad (7)$$

Thus,

$$d(x) = d(0) - \frac{b(x)}{\sin\alpha} \qquad (8)$$

In equation (2), ignoring the variety of $y_P(x)$, let $y_P = y_P(x)$,

$$y'_P(x) = \frac{f y_P \sin\alpha}{d(0) - \frac{b(x)}{\sin\alpha} + y_P\cos\alpha} \qquad (9)$$

$$y'_P(x) - y'_P(0) = \frac{f y_P \sin\alpha}{d(0) + y_P\cos\alpha - \frac{b(x)}{\sin\alpha}} - \frac{f y_P \sin\alpha}{d(0) + y_P\cos\alpha} \qquad (10)$$

$$= \frac{f y_P b(x)}{(d(0) + y_P\cos\alpha)(d(0) + y_P\cos\alpha - \frac{b(x)}{\sin\alpha})}$$

Assuming that b(x) << $d(0) + y_P\cos\alpha$,

$$y'_P(x) - y'_P(0) = \frac{f y_P b(x)}{(d(0) + y_P\cos\alpha)^2} = Cb(x) \qquad (11)$$

where *C* is a constant and in practice we used an experience number for it.

Using (11) we get b(x). Assuming b'(x) is the derivative of b(x), the square net can be adjusted to be a well-proportioned one (Figure 6) with the following equation:

$$x_{adjust} = \int_0^x \sqrt{1 + b'(x)^2}\,dt \qquad (12)$$

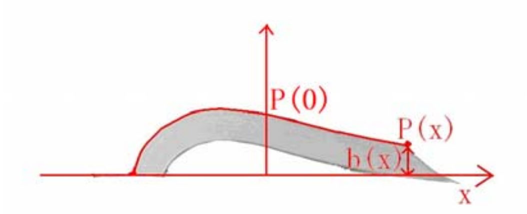


Figure 9. The curve b(x) of the page surface

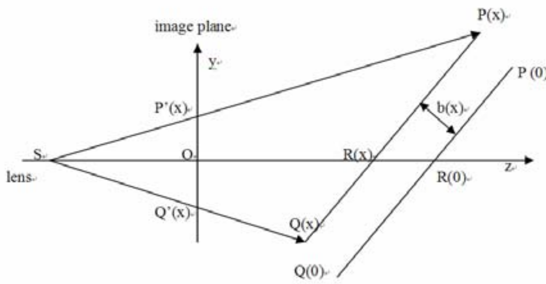
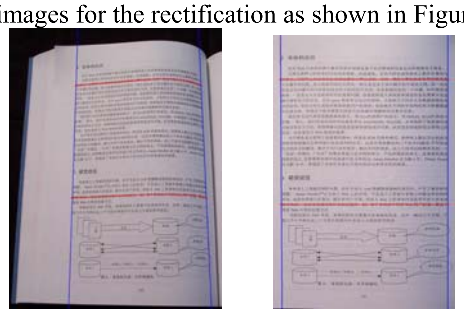


Figure 10. The yOz plane of Figure 5.

3. The Proposed Approach

According to the transform model described in Section 2, we need two line segments and two curves to dewarp a cylinder image, so our task is to find the left and right boundaries and top and bottom curves in book images for the rectification as shown in Figure 11:



(a) (b)

Figure 11. The illustration of document image before and after rectification (a) Original book image; (b) Dewarped book image.

The rectification process involves three steps: 1) the text line detection, 2) left and right boundary estimation and top and bottom curves extraction, and 3) document rectification. The flowchart of our rectification process is illustrated in Figure 12:

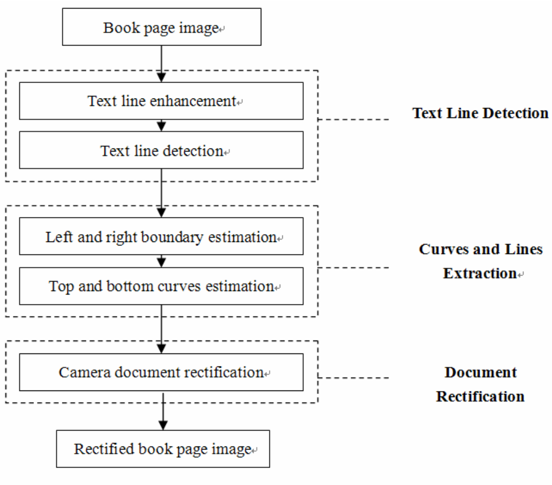
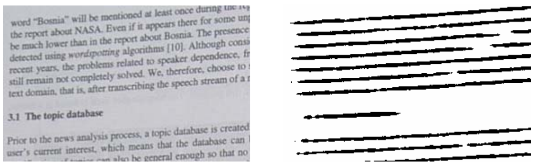


Figure 12. The flowchart of the proposed approach.

3.1 Text Line Detection

The text line detection process includes two steps. The first step makes characters on one text line connected to each other and the second step classifies the pixels on different text lines into different collection for further curve fitting.

3.1.1 Text line enhancement. S This step uses a method similar to the method proposed by Ching-Tang Hsieh in [10]. Figure 13 shows the character connection results.



(a) (b)

Figure 13. Character connection result (a) Original book image (b) Result image.

3.1.2 Pixel classification. In this step, we first detect the mid points of all text lines, and then with each mid point, we trace all the pixels belong to the same text line as it. Algorithm 1 gives the details of the pixel classification process.

Algorithm 1: Text line point collection with automatic line direction adjustment:

Input: Thinned image: $T_{m \times n}$ ' , Text line mid point set: $Set_{midpoint}$

Output: Point set of each text line $SetP_i$

For each Point (x_i, y_i) in $Set_{midpoint}$, the algorithm detects points in both left and right directions.

Step 1: detect points on the left of Point(x_i, y_i) at the same text line, as is illustrated in Figure 14. We use a step of L pixels and a detection height of H pixels (In our experiment, L = 10, while H = 6), the detection skew is set to K_t.

Let (x_{t-1}, y_{t-1}) be the point detected in the previous step, and (x_{t-2}, y_{t-2}) be the point detected before (x_{t-1}, y_{t-1}).

The skew of line direction can be calculated using the following formula (K_0 is set to be 0) :

$$K_t = K_{t-1} + (y_{t-1} - y_{t-2}) / (x_{t-1} - x_{t-2}) / 10 \quad (15)$$

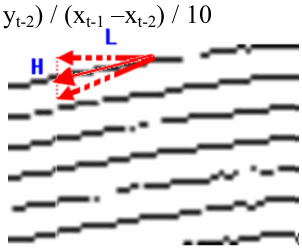
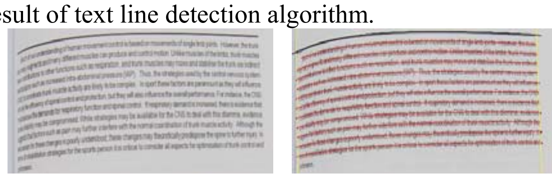


Figure 14. Text line collection step.

Step 2: a similar trace process as step 1 is used to collect all points on the right side of the mid point and the direction is to the right of the mid point of text line.

Step3: combine point sets got in step1 and step2 into $SetP_i$, this point set is then the point set of text $line_i$.

After Text line collection, all points belong to a same text line are collected. Using a polynomial curve fitting algorithm, we can get a set of polynomial curves which represent the text lines. Figure 15 illustrates the result of text line detection algorithm.



(a) (b)

Figure 15. Example of (a) Part of original book image, (b) Text line curve extracted using our proposed method - the curves across the characters are extracted text lines.

3.2 Feather curves and lines extracting

Once we get all the text lines, we estimate the left and right boundaries of the text lines and choose two text lines as the basis for transformation.

3.2.1 Left and right boundary estimation. The algorithm for estimating the left and right boundaries can be formalized as follows:

Algorithm 2: left and right boundary estimation

Input: End point set (x_i, y_i) of text lines

Output: Boundaries of columns.

Step 1: For all the end points on the left sides of all text lines, least square estimate (LSE) method is used to get a straight line L.
Step 2: Calculate the distance between (x_i,y_i) and L as D(i). Eliminate (x_i,y_i) if D(i) > T_a and (x_i,y_i) is at the right of L. (T_a is a threshold for eliminate the end points of text lines which are indent text lines or short text lines)
Step 3: do step 1 and step 2, until each D(i) < T_b. (T_b is a threshold to make sure L be close to each end point of text lines)
Then L is considered the left boundary of this page.
A similarly iterative process is used for right boundary estimation.
The result of boundary estimation is illustrated in Figure 16:

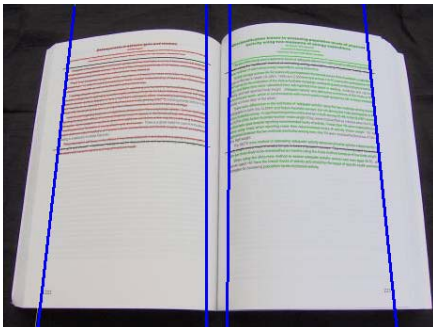


Figure 16. Left and right boundaries.

3.2.2 Top and bottom curves selection algorithm. The algorithm for selecting top and bottom curves can be formalized as follows:

Algorithm 3: Top and bottom curve selection algorithm

Let D_{li} represents the distance between the left end point of text line i and the left boundary of text lines. D_{ri} represents the distance between the right end point of text line i and the right boundary. N_{line} be the amount of text lines.

Top text line is the line with the smallest i which following the formula $(D_l + D_r) * i / N_{line} < Td$, and bottom text line is the line with the largest i which following the formula $(D_l + D_r) * (N_{line} - i) / N_{line} < Td$.

These formulas guarantee that the two lines selected are not too close to each other, and their end points are not too far from the left and right boundaries.

3.3 Document Rectification Algorithm

After estimating the left and right boundary and top and bottom curves, Rectification algorithm is applied to dewarp book image. Let curve1 denotes the curve segment generated by the top curve cut by the left and right boundaries, and curve2 denotes the bottom curve cut by the left and right boundaries.

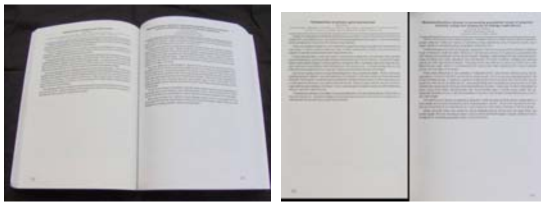
Algorithm 4: Document Rectification

Input: Original Image – $I_{m \times n}$, Curve1, Curve2

Output: Output Image - $O_{m \times n}$

Step 1: Calculate b(x) from Curve1,Curve2 according to equation (11)
Step 2: For each point P on Curve1, Q on Curve2, PQ is paralleled to the directrix.
1. Calculate a, b according to equation (3) and (4).
2. For each point on line segment PQ,
    a. Calculate its new y coordinate with (5) and x coordinate with (12)
    b. Get its color from $I_{m \times n}$ and set it to $O_{m \times n}$ on its new coordinate

After algorithm 4, the original image is rectified. The rectification result is illustrated in Figure 17.



(a) (b)

Figure 17. The transform result of a book image. (a) is the original image, (b) is the transformed result.

4. Experimental results and analysis

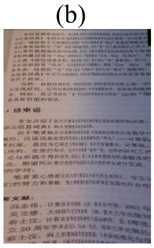
We implemented the algorithms in C# and run it on a personal computer equipped with Windows XP, and AMD 3600+ AM2 CPU, 1G Memory. An image database of 100 distorted book images that contain different languages is used for the testing. We captured the 100 book images according to the following input requirements:

1) The distance between the camera lens and page surface is about 50 - 100cm.
2) The angle between the camera lens and the book surface is less than 30 degree.
3) Book image should contain enough long text lines.
4) Book image need not contain the whole page but the text area of interest.

A Kodak DX7630 digital camera with a capture resolution of 2856 x 2142 - 6 Mega pixel is used, and images are taken under day light with dark background. Six books with different contents are involved, one book is in Chinese, four are books in English, and one is bilingual. Among the 100 images, ten percent contain pictures.

We evaluate the performance of the proposed method based on whether the text lines of a book image are straight or not and the sizes of characters on the same text line are uniform. Results are labeled as “A+”, “A”, “B”, or “C”. “A+” means the text lines in the dewarped result do not have any wave and looks exactly like generated by a scanner. “A” means the text lines have little waves, but still comfortable for reading. “B” means the dewarping result have waves and affect the feeling. “C” means the dewarping result has large waves or is hard for reading. Figure 18 illustrates the classifications.


(a) (b)

(c) (d)

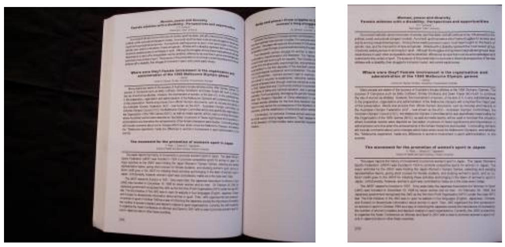
Figure 18. Sample images from each category (a) “A+” (b) “A” (c) “B” (d) “C”.

The experimental results are illustrated in Table 1.

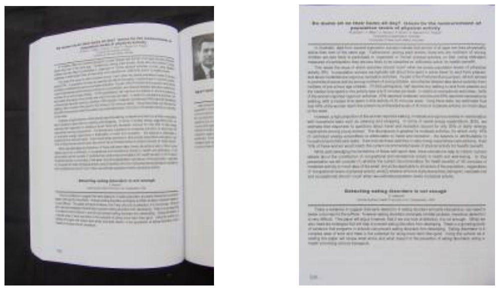
Table 1. Test result

Classification	Percentage
A+	55%
A	35%
B	9%
C	1%

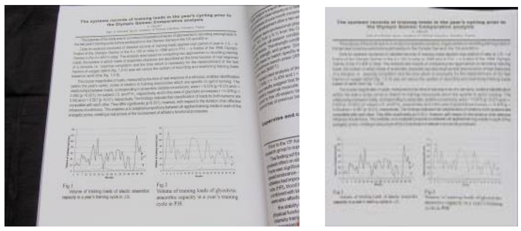
Some results are illustrated in Figure 19. Figure 20 provides a closer view of 19(a):



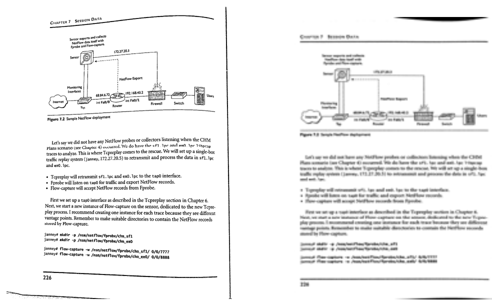
(a)



(b)

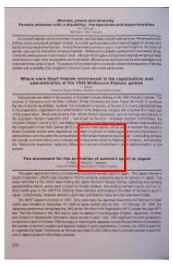


(c)



(d)

Figure 19. Transformation results.




(a) (b)

Figure 20. (a) is Figure 21(a) with a rectangle on it; (b) is a close view of (a) .

We also use the OCR software Tsinghua TH-OCR 9.0 to make a comparison of OCR qualities between before and after rectification of 20 book images. 10 books are in English and 10 books in Chinese. There are totally 9212 English words on English books. And 7102 Chinese characters on Chinese books. 5923 English words are correctly recognized before rectification, while 8788 words recognized after rectification. 305 Chinese characters are correctly recognized before rectification, while 6726 characters recognized after rectification.

Table 2 illustrates the OCR rates before and after rectification.

Table 2. OCR Test result

Book Language	Before Rectification (OCR rate)	After Rectification (OCRrate)
English	64.3%	95.4%
Chinese	4.3%	94.7%

Our experimental results reveal that the coordinate transform model and document rectification approach proposed in this paper can rectify both perspective distortion and warping well. It ensures effective layout analysis and word segmentation, hence bringing about higher reading performance and less recognition errors.

5. Conclusion

In this paper, we propose a model for dewarping a cylinder image into a flat one. This model is applied to the problem of rectifying document image taken by a digital camera, which suffers from perspective distortion and document surface warping. Our experimental results show that our approach can dewarp the curved surface well, and it is efficient on warped document images of different languages.

Further improvement in the processing speed and de-blurring of the shade area will be explored. As digital camera is developed rapidly, the proposed document restoration technique may open a new channel for document capture and digitalization.

Acknowledgements

This work is financial supported by Laserfiche. The authors would also like to thank and Kurt.Rapelje, Ryan.Wong and Ed Heaney from Laserfiche for their valuable comments on this work.

Reference

[1] M. S. Brown and W. B. Seales, “Image restoration of arbitrarily warped documents”. IEEE Transactions on Pattern Analysis and Machine Intellegence, 26(10):1295–1306, October 2004.

[2] 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, August 2004, Cambridge, UK, page 482 - 485.

[3] M. Pilu, “Extraction of illusory linear clues in perspectively skewed documents”. In Proceedings of the Conference on Computer Vision and Pattern Recognition,
volume 1, pages 363–368, 2001.

[4] P. Clark, M. Mirmhedi, “Rectifying perspective views of text in 3Dscenes using vanishing points”, Pattern Recognition, vol. 36, pp. 2673-2686, 2003

[5] A. Ulges, C. H. Lampert, and T. M. Breuel, “Document image dewarping using robust estimation of curled text lines”, in Proceedings of the Internatioanl Conference on Document Analysis and Recognition, 2005, pp. 1001–1005

[6] Shijian Lu, Chew Lim Tan, “The Restoration of Camera Documents through Image Segmentation”, Analysis Systems, 2006, pp. 484-495

[7] H. Cao, X. Ding, and C. Liu, “Rectifying the bound document image captured by the camera: A model based approach”. In Seventh International Conference on Document Analysis and
Recognition - ICDAR2003, pages 71–75, 2003.

[8] Jian Liang, Daniel DeMenthon, David Deormann “Flattening Curved Documents in Images”, International Conference on Computer Vision and Pattern Recognition, 2005, pp. 338-345.

[9] Hironori Ezaki and Seiichi Uchida, “Dewarping of document image by global optimization”, in Proc. 8th International Conference on Document Analysis and Recognition, 2005, pp. 302-306.

[10] Ching-Tang Hsieh, Eugene Lai, You-Chuang Wang “An effective algorithm for fingerprint image enhancement based on wavelet transform” Pattern Recognition 36 (2003) 303 – 312”.