Available online at www.sciencedirect.com

SCIENCE DIRECT®

Image and Vision Computing 24 (2006) 837–848

www.elsevier.com/locate/imavis

A partition approach for the restoration of camera images of planar and curled document

Shijian Lu, Ben M. Chen *, C.C. Ko

Electrical and Computer Engineering Department, National University of Singapore, Singapore, Singapore 119260

Received 1 May 2005; received in revised form 28 November 2005; accepted 16 February 2006

Abstract

As camera resolution increases, high-speed non-contact text capture through a digital camera is opening up a new channel for text capture and understanding. Unfortunately, the captured document images are normally coupled with the perspective and geometric distortions that cannot be handled by the existing optical character recognition (OCR) systems. In this paper, we propose a new technique, which is capable of removing the perspective and geometric distortions, and reconstructing the fronto-parallel view of text with a single document image. Different from reported approaches in the literature, the restoration of the distorted camera documents is carried out through the image partition, which divides the documents into multiple small image patches where text can be approximated to lie on a planar surface. The global distortion is thus corrected through the local rectification of the partitioned image patches one by one. Experimental results show that the proposed method is fast and easy for implementation.

Keywords: Document image analysis; Document image rectification; Optical character recognition; Morphological image processing; Fuzzy sets

1. Introduction

Document image restoration, a preprocessing procedure before the OCR operation, mainly focuses on the removal of rotation-induced skew [1–8] introduced during the scanning process by a document scanner. As camera resolution increases in recent years, high-speed non-contact text capture through a digital camera is becoming an alternative choice. Consequently, perspective distortion introduced during the capturing process by a digital camera and the geometric distortion resulting from the non-flat document surface bring up two new problems to the existing OCR systems. Similar to the compensation of rotation-induced skew, perspective and geometric distortion must be removed before the captured camera documents are fed to the generic OCR systems.

A few perspective document rectification techniques [9–13] have been reported in recent years. As a common feature, all these reported techniques assume that documents lie on an ideally planar surface. Unfortunately, most of documents in the real world such as the hand-held newspaper, the paper sheets pasted on cylindrical containers, and even the book pages bound within the thick volumes, lie on a smoothly curved instead of planar surface. Furthermore, unlike the scanning process where documents are physically flattened over the glasses, the capturing process by a digital camera cannot remove the geometric distortion resulting from the non-flat document surfaces.

Some research works [14–21] have also been reported to restore the camera images of non-flat documents. In Brown and Seales [14], presented a framework for restoring the arbitrarily warped documents to their original planar shape. A structured lighting device is set up for 3D measurements and reconstruction. In [15], a laser projector is mounted to capture the geometric structure of the document images that are distorted due to the old bookbinding. In Kawarago et al. [16], propose to use a stereo vision system for 3D measurements. With the reconstructed 3D model [14–16], curved documents are finally flattened through certain 3D mesh manipulation. The limitation of the above three methods [14–16] is that they all require some auxiliary hardware equipments for 3D reconstruction.

Instead of relying on auxiliary hardware, some techniques [17–21] have been reported to exploit certain geometry surfaces to model the smoothly curved documents. For example, Cao et al. [17] use a cylindrical surface to model the shape of the document captured using a digital camera. Apart from cylinder surface assumption, their method requires

* Corresponding author. Tel.: +65 6874 2289; fax: +65 6779 1103.
E-mail address: bmchen@nus.edu.sg (B.M. Chen).

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2006.02.008

that cylinder generatrix should be parallel to the image plane. Zhang [18] also use a cylindrical surface to model the geometric distortion of the bound books that cannot be opened to 180° during scanning process. The proposed algorithm assumes that document text is scanned horizontally and so the flat part of text lines lies on a horizontal straight line. Therefore, it cannot handle the camera images of documents lying over a smoothly curved surface. Later, Tsoi [19] and Liang [20] propose to use a ruled and developable surface to model the document shape. Both methods require that vertical text direction should be roughly parallel to the surface ruling and so they cannot restore documents curled in arbitrary direction.

In [10], we proposed to restore the camera image of planar documents through the exploitation of the vertical stroke boundary (VSB) and the $\boldsymbol{x}$ lines and base lines of text as labeled with (1) and (2) in Fig. 1. In this paper, we extend our earlier work to deal with the restoration of the camera documents where text lies over a planar or smoothly curved surface. We focus on the situations where text line shape can be modeled using a cubic polynomial curve. Compared with the reported methods, our method can handle the camera images in perspective view [9–13] or curled in arbitrary direction [17–20]. It needs no auxiliary hardware [14–16], no camera calibration, and the only thing required is a single document image captured using a common digital camera.

Our method restores camera documents through the image partition, which divides camera images into multiple quadrilateral patches where text can be approximated to lie on a planar surface. In our proposed technique, document partition is implemented through the exploitation of the VSBs and the $\boldsymbol{x}$ lines and base lines of text [10]. For each partitioned patch, a target rectangle is constructed where rectangle height is commonly determined as the average size of the identified VSBs and the width is estimated through a character categorization process that classifies characters into different categories with different aspect ratios. With the partitioned image patches and the constructed target rectangles, global distortion are finally removed through the local rectification of partitioned image patches one by one.

The remaining manuscript is organized into three sections. In Section 2, the details of the proposed document restoration technique is described. We then present experimental results and discussions in Section 3. Finally, we draw some concluding remarks in Section 4.

2. Proposed restoration technique

In this section, we present a detailed procedure of the proposed document restoration technique. In particular, we will divide this section into a few subsections, which deal with the partition of the camera images of documents, the construction of target rectangles, and the final restoration of the camera documents based on the constructed target rectangles and the partitioned quadrilateral patches.

2.1. Document image partition

In this section, we describe the detailed document partition procedure, which includes document binarization, vertical stroke boundary identification, $\boldsymbol{x}$ line and base line fitting, and the final document partition based on the identified VSBs and the fitted $\boldsymbol{x}$ lines and base lines of text.

2.1.1. Document binarization

Camera images must be binarized first before the ensuing VSB and $\boldsymbol{x}$/base line extraction. Compared with scanned documents, camera documents, especially the ones captured in an indoor environment, are normally more susceptible to shading degradation. Therefore, adaptive instead of global thresholding techniques are normally required. A number of document binarization methods [23] have been reported in the literature. We adopt Niblack's method [22] since his method generally outperforms others in terms of speed and segmentation quality [23]. Connected component labeling is then implemented. Binarization noises with small sizes are removed using a size filter. We set the threshold at 10 because nearly all labeled character components contain much more than 10 pixels.

2.1.2. Vertical stroke boundary identification

VSB refers to the vertical stroke boundary such as the left-side boundary segment of characters 'b' and 'k' and the right-side boundary segment of characters 'd' and 'q'. A VSB identification technique has been reported in our earlier work [10] where oriented stroke boundary segments are firstly extracted using a set of customized morphological operators

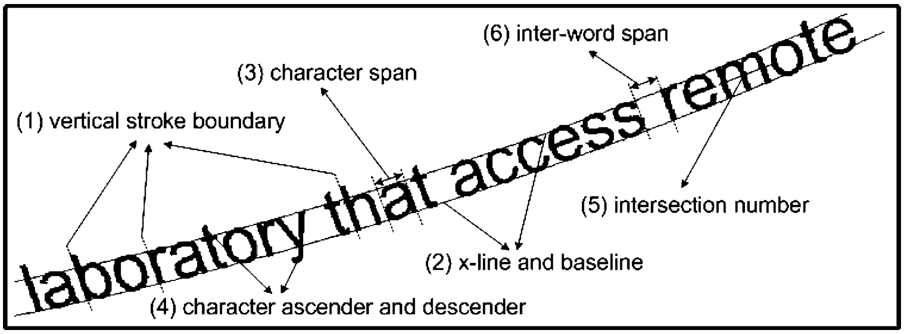

(6) inter-word span
(3) character span
(1) vertical stroke boundary
laboratory that access remote
(5) intersection number
(2) x-line and baseline
(4) character ascender and descender

Fig. 1. Text line definition.

[24]. Three fuzzy sets, namely, size set S, linearity set L, and orientation set P, are then constructed to evaluate whether an extracted stroke boundary segment is the desired VSB. Lastly, the desired VSBs are identified through a fuzzy aggregation operator that combines three constructed sets S, L, and P.

But for camera images of smoothly curved documents, the orientation of the desired VSBs may vary within a much bigger range. Therefore, VSBs in certain image regions may be falsely rejected through a simple aggregation of S, L, and P. In this paper, we instead propose to first determine the VSB candidates based on constructed S and L sets. The desired VSBs are then identified through the exploitation of the orientation of the determined VSB candidates. We illustrate the VSB identification process using the sample word 'laboratory' given in Fig. 3(a), which is cut from a smoothly curved camera document. Similar to the stroke boundary extraction technique in [10], stroke boundary segments as shown in Fig. 3(b) are first extracted using the customized morphological operators. The extracted stroke boundary segments are then filtered using a size filter. We set the size threshold at eight because the desired VSBs normally contain much more than 8 pixels. Fig. 3(c) labels the filtered stroke boundary segments.

The size of stroke boundary is defined as the number of pixels that stroke boundary contains. As the desired VSBs are normally bigger than other stroke boundary segments, a stroke boundary segment with bigger size should be assigned a bigger membership value. We therefore construct the size set L using Zadeh's tion $S(x;\, a,\, b,\, c)$ [25] as shown in Fig. 2(a)

$$S(x;a,b,c) = \begin{cases} 0 & x \le a \\ 2\left(\dfrac{x-a}{c-a}\right)^2 & a < x \le b \\ 1-2\left(\dfrac{x-c}{c-a}\right)^2 & b < x \le c \\ 1 & x > c \end{cases}, \qquad (1)$$

where parameter b represents crossover point and it is set at the median stroke boundary size. Parameter a is defined as the minimum stroke boundary size and parameter c is set at $2b-a$. We set parameter b at the median instead of average boundary size because the stroke boundary segments extracted from non-text components such as graphics may be much bigger than those from characters. For the stroke boundary segments labeled in Fig. 3(c), Table 1 gives the calculated size membership values (***SMVs***).

The linearity property describes the 'straightness' of a stroke boundary. We evaluate the linearity of an extracted stroke boundary segment using the distance defined below

$$dist = \frac{1}{n}\sum_{i=1}^{n} d(p_i, l), \qquad (2)$$

where n is the pixel number of the stroke boundary, l refers to the straight line estimated through the least square fitting of the boundary pixels, and function d calculates the distance between pixel p_i and the fitted straight line l.

As defined in Eq. (2), the stroke boundary is straighter if the calculated distance is smaller. As the desired VSBs are normally straighter than other stroke boundary segments, we construct the linearity set using the complement of Zadeh's S-function $C(x;\, a,\, b,\, c) = 1 - S(x;\, a,\, b,\, c)$ as shown in Fig. 2(b). Parameters a and c are set at the minimum and maximum distances given in Eq. (2) and parameter b represents the crossover point $(a+c)/2$. For the stroke boundary segments labeled in Fig. 3(c), Table 1 gives the calculated linearity membership values (***LMVs***).

VSB candidates can then be determined using the fuzzy aggregation operator [26] that combines the constructed size and linearity sets S and L

$$(S\Theta L)(x) = \gamma(S \cap L) + (1-\gamma)(S \cup L), \qquad (3)$$

where $\cup$ and $\cap$ refers to the union and intersection operations, and γ stands for the compensation factor $\gamma \in [0,1]$, indicating where the actual operator is located between AND and OR. We determine γ as 0.6 to give the size a bit higher weight. For the stroke boundary segments labeled in Fig. 3(c), Table 1 gives the aggregation membership values (***AMVs***) calculated using Eq. (3).

VSB candidates can thus be determined as α cut of constructed aggregation set AMV. For documents printed in Latin-based languages, the ratio between VSBs and characters is roughly 1:2. We therefore set α at $AMV_{n/2+1}$ where AMV represents the constructed aggregation set. Parameter n refers

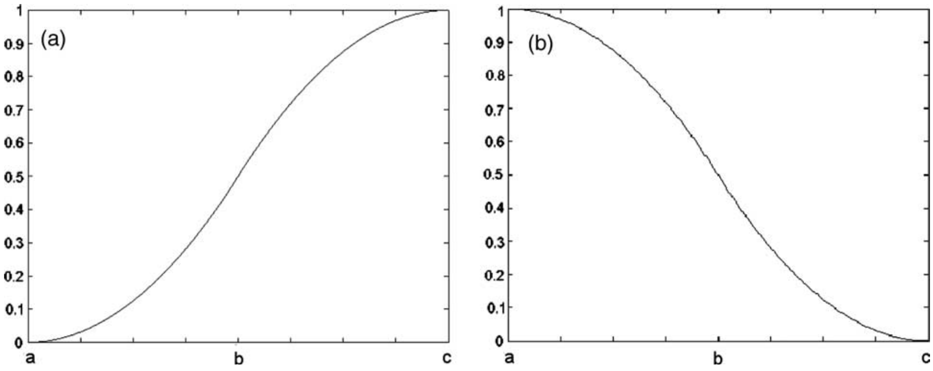


Fig. 2. Membership function: (a) S-function; (b) the complement of S-function.

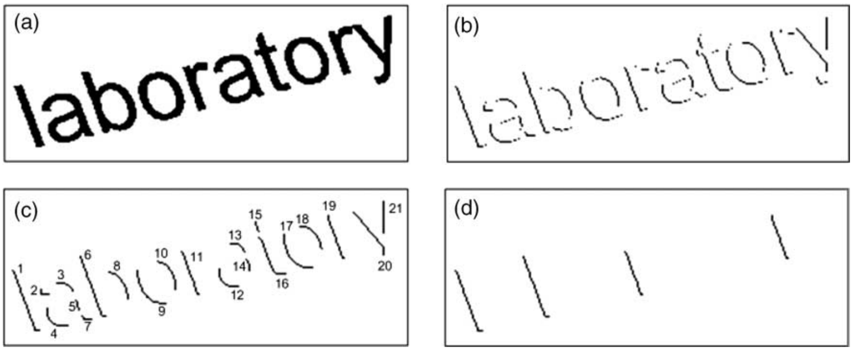


Fig. 3. vertical stroke boundary identification: (a) distorted word; (b) extracted left-side stroke boundaries; (c) labeled stroke boundaries after size filtering; (d) identified vertical stroke boundaries.

to the labeled character number. Therefore, parameter α is determined as $(n/2+1)$th biggest aggregation value and the $n/2$ stroke boundaries with *AMV* bigger than α are determined as VSB candidates. Considering the document degradation where some characters may be broken or merged, parameter α can be relaxed to be $(n/3)$th or even $(n/4)$th biggest aggregation value to ensure that VSBs can be identified correctly. The relaxation does not affect the restoration result because the subsequent document partition does not require so many VSBs.

For sample word given in Fig. 3(a), the number of the VSB candidates is 5 because 10 characters are captured. Therefore, parameter α is determined as 0.7259, which is the sixth biggest aggregation value corresponding to that of the stroke boundary 21. Accordingly, the stroke boundaries 1, 6, 11, 19, and 20 are determined as the VSB candidates. The aggregation values of the determined VSB candidates are labeled with bold numbers in '***AMV***' columns in Table 1.

Based on the size and linearity properties, the stroke boundaries of some characters such as 'v', 'w', and 'y' are frequently classified as the VSB candidates as well. For sample word in Fig. 3(a), stroke boundary 20 as given in Fig. 3(c) is also determined as a VSB candidate because it is big and straight. The undesired VSB candidates can be further rejected based on their orientations, which is evaluated as the slope of the straight line that fit to the determined VSB candidates

Table 1
Membership and orientation property

	SMV	***LMV***	***AMV***	***PV***
BS 1	0.9999	0.8967	**0.9586**	**2.5894**
BS 2	0.0064	0.6359	0.3841	0.3333
BS 3	0.2715	0.0028	0.1640	0.7011
BS 4	0.0402	0.6504	0.4063	0.4464
BS 5	0	0.9472	0.5683	2.8000
BS 6	0.9973	0.9089	**0.9620**	**2.6378**
BS 7	0.0016	0.8496	0.5104	0.2302
BS 8	0.3148	0.1909	0.2653	1.3378
BS 9	0.7575	0	0.4545	1.0873
BS 10	0.3614	0.1057	0.2591	1.4570
BS 11	0.7165	0.9384	**0.8496**	**2.7006**
BS 12	0.2313	0.0159	0.1452	0.7707
BS 13	0.0257	0.5909	0.3648	0.5204
BS 14	0.0016	0.9012	0.5413	2.7500
BS 15	0	0.9506	0.5704	2.6250
BS 16	0.7575	0.0494	0.4743	1.5881
BS 17	0.7575	0	0.4545	1.0601
BS 18	0.2715	0.1630	0.2281	0.9958
BS 19	0.7165	0.9364	**0.8484**	**2.7100**
BS 20	0.6722	0.8021	**0.7501**	**1.2861**
BS 21	0.3148	1.0000	0.7259	∞

BS, stroke boundary; ***SMV***, size membership value, ***LMV***: linearity membership value; ***PV***, pose value; ***AMV***, aggregation membership value.

$$orient = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2} \tag{4}$$

where x_i and y_i refer to coordinates of ith boundary pixels. $\bar{x}$ and $\bar{y}$ denote the expectations of x and y coordinates of boundary pixels. For the stroke boundaries labeled in Fig. 3(c), the column ***PV*** in Table 1 gives the calculated orientation values.

Though the orientation of the desired VSBs varies within a big range within the whole image, the orientation variation within a local image region is normally much smaller. We therefore identify the desired VSBs based on the orientation expectation estimated based on the local VSB candidate neighbors. For each VSB candidate, the orientation expectation is determined as

$$VT_t = \frac{1}{N} \sum_{i=1}^{N} orient_i \tag{5}$$

where $orient_i$ refers to the orientation of ith determined VSB candidate. Parameter N is the number of the VSB candidates nearest to the studied one. We set it at 4–8 so that Eq. (5) gives the local estimation. For the stroke boundary 20 given in Fig. 3(c), its orientation is far from its expectation calculated using the other four VSB candidate neighbors. Therefore, it is not the desired VSB and should be rejected. For sample word in Fig. 3(a and d) gives the identified VSBs.

2.1.3. X line and base line fitting

We exploit the ***x*** lines and base lines of text for document shape estimation. In [10], we propose a point tracing technique to classify character tip points to different text lines within a camera image of planar document. The ***x*** lines and base lines of text are then estimated through the least square fitting of the classified characters tip points. In this paper, we adapt this technique to construct the $\boldsymbol{x}$ lines and base lines within the smoothly curved camera documents.

Instead of using character tip points, we exploit the character extremum points for $\boldsymbol{x}$ line and base line fitting. Character extremum points refer to the highest and lowest character pixels in the direction of nearest VSB. Therefore, each character with no ascender or descender has two extremum points lying around the $\boldsymbol{x}$ line and base line positions, respectively. Fig. 4 illustrates the tracing process where the white, light gray and darker gray points represent the extremum points lying around the $\boldsymbol{x}$ line position of three adjacent text lines. The tracing process is controlled by two distance constraints, namely, point to point constraint and point to line constraint. T_{pl} defines the point to line distance constraint. As Fig. 4 shows, all extremum points satisfying this constraint lie within the band defined by two parallel lines L_b and L_t, which are parallel and with the same distance, T_{pl}, to the tracing line L.

For camera images of planar document, $\boldsymbol{x}$ line and base line of text can be fitted using the straight line model. Therefore, we fit the tracing line L using all classified character extremum points C_i ($i=1\ldots n$) in [10]. But for camera image of smoothly curved documents, character extremum points fit well to a cubic polynomial curve in most cases. We therefore fit the tracing line L using the last 4–8 classified extremum points because L fitted using the last 4–8 classified extremum points is able to predict the position of the next character extremum point accurately. If there is less than four extremum points classified, the tracing line L can be directly determined as the straight line that passes through the last classified extremum point C_l with orientation estimated based on the relative orientation of adjacent character pairs nearest to C_l.

T_{pp} defines the point to point distance constraint and it stops the tracing process of short text lines at the text line end positions. As Fig. 4 shows, all extremum point satisfying this constraint lie within the circle with center at C_l and radius equal to T_{pp}. We determine these two distance thresholds based on the identified VSBs

$$T_{pl} = k_l VBS_{avg}, \qquad T_{pp} = k_p VBS_{avg} \tag{6}$$

where VBS_{avg} refers to the average size of the identified VSBs, which normally reflect the height of the labeled characters. Parameter k_l and k_p adjust two distance thresholds. We set them at 1 and 4, respectively, to control the tracing process within the studied text line.

As Fig. 4 shows, the light gray points {1}–{8} refer to the classified character extremum points where point {8} is the last classified one C_l. To search for the next extremum point that belong to the same text line, the tracing line L is firstly fitted using the last 4 classified extremum points (light gray points {5}–{8}). The extremum points that satisfy this constraint are accordingly determined as the white points {13}–{16} and light gray points {9}–{12}. At the same time, the extremum points satisfying the point to point distance constraint include the white points {5}–{10}, light gray points {9}–{11}, and dark gray points {6}–{9}. Therefore, the next extremum point that should be classified to light gray points {1}–{8} group is finally determined as light gray point {9}, which satisfies two distance constraints and at the same time, it is closest to the last classified extremum point C_l (point {8} in this example). The remaining extremum points can be classified to the related text lines in a similar way.

For documents lying over a smoothly curved surface, the shape of text lines can be modeled using one or more pieces of cubic polynomial. In most cases, $\boldsymbol{x}$ lines and base lines of text can be modeled using a single cubic polynomial

$$C = p_0 + p_1 x + p_2 x^2 + p_3 x^3 \tag{7}$$

where coefficients p_0, p_1, p_2, p_3 are determined through the least square fitting of the classified character extremum points

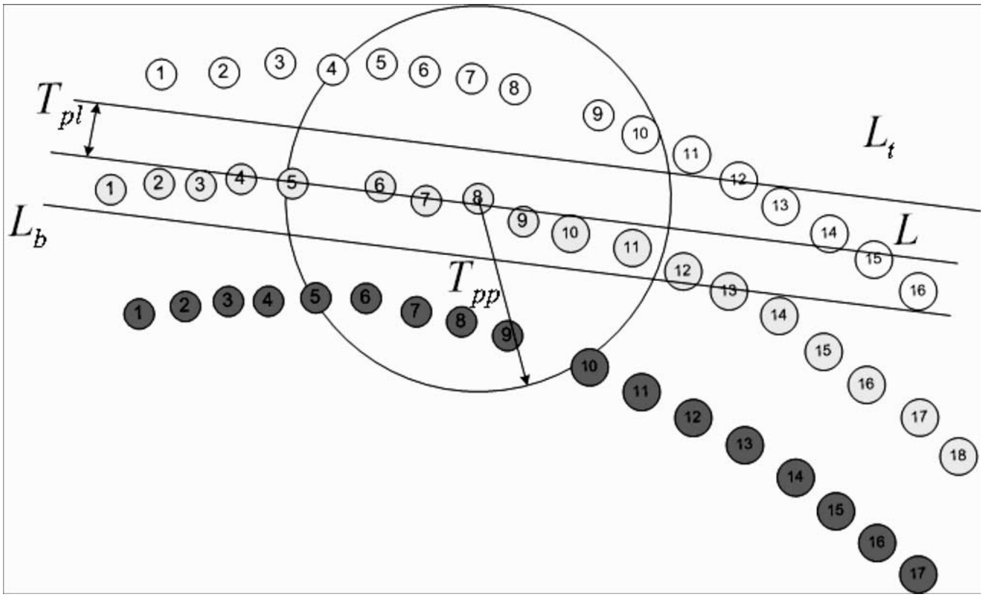


Fig. 4. Character extremum point classification process.

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i & \cdots & \sum_{i=1}^{n} x_i^k \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \cdots & \sum_{i=1}^{n} x_i^{k+1} \\ \cdots & \cdots & \cdots & \cdots \\ \sum_{i=1}^{n} x_i^k & \sum_{i=1}^{n} x_i^{k+1} & \cdots & \sum_{i=1}^{n} x_i^{2k} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \cdots \\ p_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \cdots \\ \sum_{i=1}^{n} x_i^k y_i \end{bmatrix} \quad (8)$$

where x_i and y_i ($i=1\ldots n$) correspond to the x and y coordinates of the extremum points that are classified to one specific text line. Parameter k is 3 for cubic polynomial.

2.1.4. Document image partition

In this section, we present the proposed document partition process, which divides camera images of document into multiple quadrilateral patches through the exploitation of the identified VSBs and the fitted ***x*** lines and base lines.

For sample word 'laboratory' given in Fig. 3(a), Fig. 5(a) shows the fitted ***x*** line and base line and the identified VSBs. The identified VSBs must be processed further to partition the text lines completely. Firstly, some VSBs must be deleted in order to control the width of the partitioned document patches. The deletion is carried out based on the distance between the adjacent VSBs within the same text line, which is defined as the Euclidean distance between their centroids

$$c_x = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad c_y = \frac{1}{n}\sum_{i=1}^{n} y_i \quad (9)$$

where x_i and y_i ($i=1,\ldots, n$) denote the ***x*** and y coordinates of the VSB pixels. Parameter n refers to the number of pixels within the studied VSB. In our proposed technique, the distance threshold is determined as

$$D_{thre} = k_d VBS_{avg} \quad (10)$$

where VBS_{avg} represents the average size of the identified VSBs. To restore the partitioned document patches more efficiently, we set k_d at 3–6 so that each partitioned document patch roughly encloses 3–6 characters. Fig. 5(b) shows the processed VSBs where the second VSB in Fig. 5(a) is deleted.

In addition, for text lines that have no VSBs detected at their left or right end position, one must be estimated there to enclose all characters that belong to the studied text line. The orientation of the VSBs at text line end positions can be estimated through the linear interpolation

$$pv = pv'' + \frac{(x-x'')(pv'-pv'')}{(x'-x'')} \quad (11)$$

where x is x coordinates of the leftmost or rightmost character pixel. x' and x' denote the x coordinates of centroids of two VSBs nearest to x. Parameters pv' and pv'' refer to the orientation of two nearest VSBs.

For sample word given in Fig. 3(a), two VSBs nearest to right end position correspond to the stroke boundary 11 and 19 labeled in Fig. 3(c). Based on the ***PV*** values of ***BS 11*** and ***BS 19*** in Table 1 and their centroid positions given in Eq. (9), the orientation of the VSB at the right end of sample word 'laboratory' can accordingly be determined as 2.7210 using Eq. (11). For sample word in Fig. 3(a), Fig. 5(c) shows the estimated VSB at right end position. For each processed VSB, a straight line can be determined through the least square fitting of the VSB pixels. The sample word 'laboratory' is finally partitioned into three quadrilateral patches in Fig. 5(d).

2.2. Target rectangle construction

We propose to construct the target rectangles based on the number of the characters enclosed within the partitioned source quadrilateral and the specific character aspect ratios. Character aspect ratios are determined through a character categorization process, which classifies characters and other text symbols to several categories with different aspect ratios.

We focus on documents printed in Latin-based languages and classify characters and other text symbols into six categories with six specific aspect ratios. The classification is carried out based on character shape features including character span, character ascender and descender, and the intersection number as labeled with (3–5) in Fig. 1. Character span refers to the span of a character in the direction orthogonal to the nearest VSB. It can be determined as the distance between two parallel character tangent lines $L_l(a, b, c)$ and $L_r(a, b, d)$ as shown in Fig. 1:

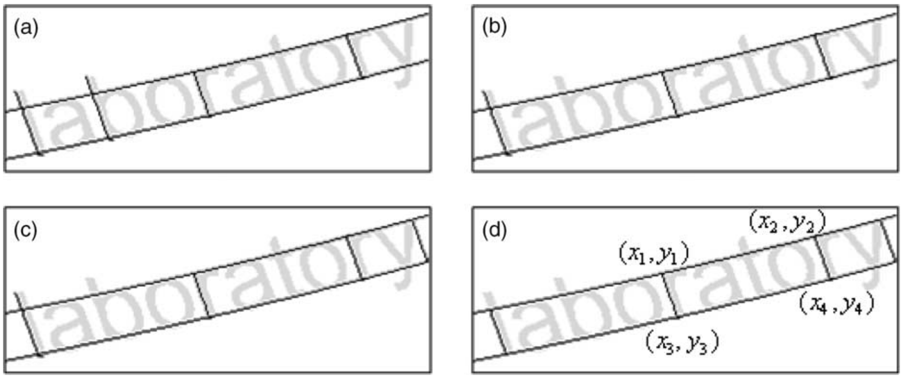


Fig. 5. Document image partition: (a) identified VSBs and the ***x*** and base line; (b) VSBs after deletion operation; (c) VSB estimated at text line end position; (d) partition results.

$$SP = \frac{|c-d|}{\sqrt{a^2+b^2}} \tag{12}$$

Character ascender and descender can be detected based on relative positions between character extremum points and the ***x*** lines and base lines, as extremum points from character ascender and descender are all far above or below the ***x*** lines and base lines of the related text line. The last feature refer to intersection number and it is equal to the number of intersection between character strokes and the straight line passing through character centroid with orientation orthogonal to the nearest VSB. For example, for characters ‘l’, ‘n’, and ‘m’, the horizontal intersection numbers are 1–3, respectively.

The character classification algorithm is as follows:

Inputs: Calculated character spans *CSpan*; Ascendant and descendant information *ADInfo*; intersection numbers *Inter*

```
Procedure CC (CSpan, ADInfo, Inter)
(1) Initialize $i=1$
(2) Calculate the average character span $CSpan\_mdn$ based on $CSpan$.
    (3) If $Inter(i) \geq 3$ and $ADInfor=1$ (with ascendant), character is classified as M, W, @…
    (4) Else if $Inter(i) \geq 3$ and $ADInfor=0$ (no ascendant), character is classified as m, w…
    (5) Else if $ADInfor=1$ (with ascendant) and $CSpan(i) > k_u \cdot CSpan\_mdn$, character is classified as A–H, J–L, N–V, or X–Z…
    (6) Else if $CSpan(i) > k_l \cdot CSpan\_mdn$ and $CSpan(i) < k_u \cdot CSpan\_mdn$, character is classified as a–e, g–h, k, n–q, s, u–v, x–z, 0, 2–9, #, â, ë…
    (7) Else if $CSpan(i) < k_s \cdot CSpan\_mdn$, character is classified as i, j, l, I, 1, (,), !, [, |, {…
    (8) Else, character is classified as t, f, r, …
    (9) $i=i+1$
```

Table 2 shows the six categories and the related aspect ratios. We note that texts printed in different styles may hold different aspect ratios. We fix the aspect ratios with six categories because the aspect ratio of text in different styles normally varies within a small range and the generic OCR systems are tolerant of small aspect ratio variation as well. To classify characters into six categories, the median character span *CSpan_mdn* in Step (2) is first calculated. Parameter k_u, k_l, and k_s in Steps (6–8) are three key parameters for character classification. We determine them as 1.2, 0.7, and 0.3, respectively, based on the observation of relative aspect ratios of characters in different categories.

We test classification performance using 30 camera images of curved document text printed in different fonts and languages. The classification rate defined by the ratio between the number of correctly classified characters and that of all labeled characters reaches over 96%. We note the small classification errors will not lead to the obvious restoration errors. This is partially due to the fact that partitioned document patches normally enclose 3–6 characters as determined by Eq. (10). On the other hand, though document texts printed in different styles possess different aspect ratios, the relative aspect ratios of different characters printed in different styles normally vary within a small range. For example, though characters ‘a’ and ‘r’ in Time New Roman and ‘a’, ‘r’ in Verdana have different aspect ratios, the relative aspect ratio between ‘a’ and ‘r’ in Time New Roman and Verdana is nearly the same.

Similar to character span, inter-word span as labeled with (6) in Fig. 1 must be calculated as well to restore the width of target rectangles. In our proposed technique, the inter-word blanks are detected based on the centroid distance between two adjacent characters. We set the distance threshold at three VBS_{avg} because the distance between adjacent characters within the same word is generally much smaller than three VBS_{avg}. Similar to character span, inter-word span can be determined as the distance between two parallel blank tangent lines as shown in Fig. 1. The aspect ratio of inter-word blanks and target rectangle height can accordingly be determined based on the relation between calculated inter-word spans and the average character span *CSpan_mdn*:

$$R = \frac{WB}{CSpan\_mdn} \tag{13}$$

The width of target rectangles can finally be determined as

$$T_w = \sum_{i=1}^{n} R_i VBS_{avg} \tag{14}$$

where VBS_{avg} represent the average size of identified VSBs and parameter n represents the number of characters and inter-word spans enclosed. Parameter R_i refers to aspect ratios of characters and inter-word blanks within the partitioned image patches as given in Table 2 and Eq. (14). For the partitioned image patches given in Fig. 5(d), Fig. 6(a) shows the constructed target rectangles.

Table 2
Character classification and related width–height ratio

Classified characters	Character width–height ratios (***R***)
M, W, @…	1.6:1
m, w	1.4:1
A–H, J–L, N–V, X–Z…	1.2:1
a–e, g–h, k, n–q, s, u–v, x–z, 0, 2–9, #, â, ë…	0.8:1
t, f, r, …	0.5:1
i, j, l, I, 1, (,), !, [, |, {…	0.2:1

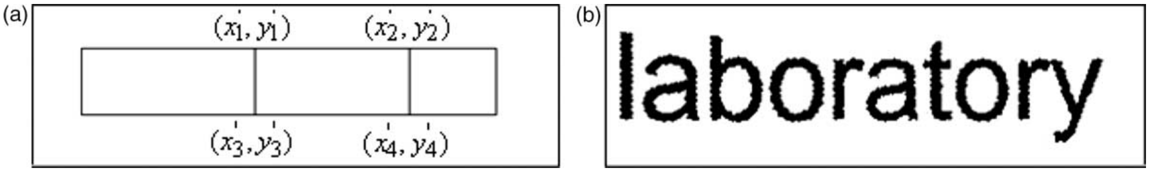
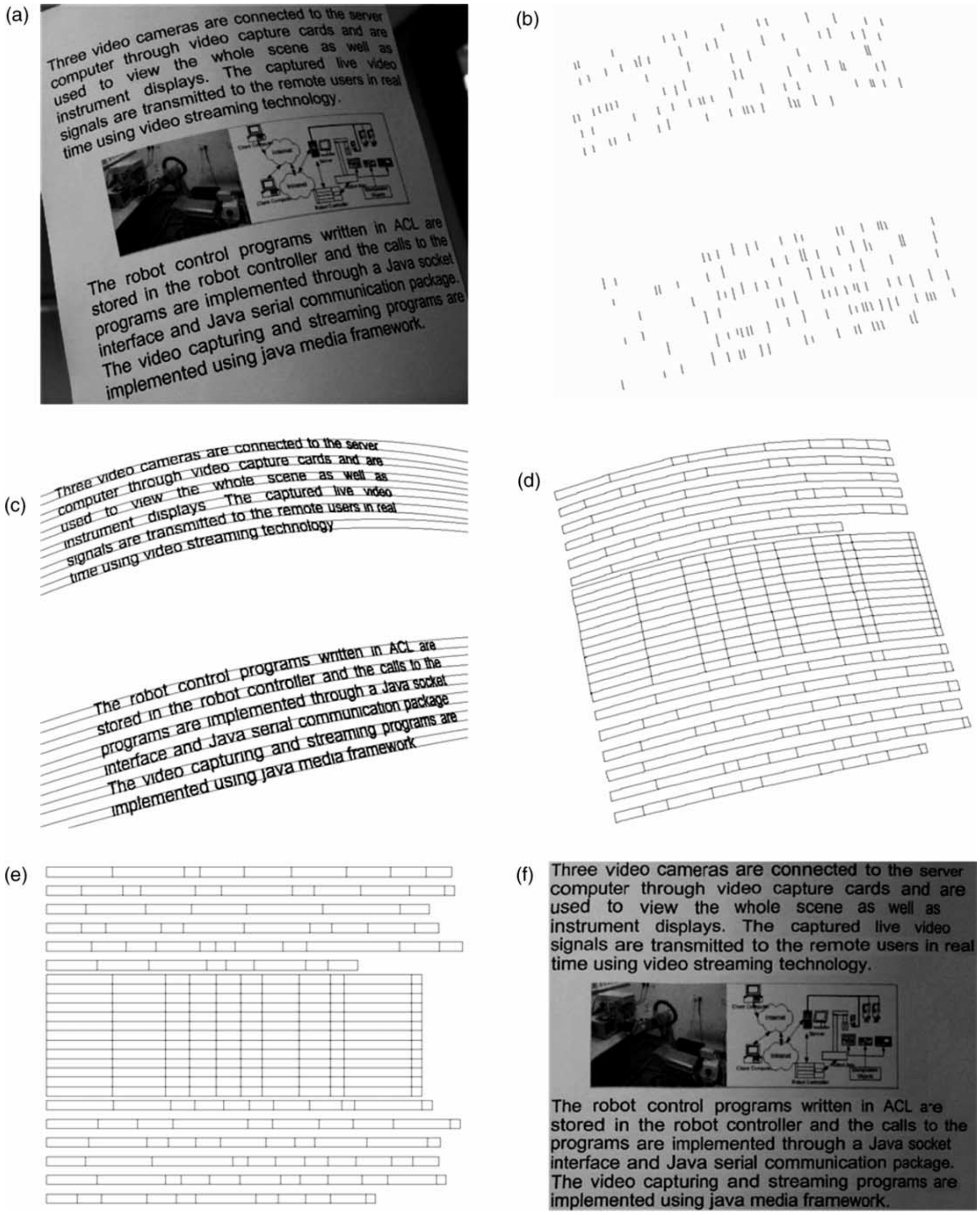


Fig. 6. Document image restoration: (a) constructed target rectangles; (b) rectified document text.

2.3. Document restoration

With the correspondence between the partitioned quadrilateral patches and the constructed target rectangles, the camera documents with perspective and geometric distortions can be restored using the four point mapping algorithm [27]. The rectification homography between the perspective and front-parallel view of each partitioned document patch can be



Three video cameras are connected to the server computer through video capture cards and are used to view the whole scene as well as instrument displays. The captured live video signals are transmitted to the remote users in real time using video streaming technology.

The robot control programs written in ACL are stored in the robot controller and the calls to the programs are implemented through a Java socket interface and Java serial communication package. The video capturing and streaming programs are implemented using java media framework.

Fig. 7. Perspective and geometric distortion rectification: (a) distorted camera document; (b) identified vertical stroke boundaries; (c) fitted ***x*** lines and base lines; (d) partitioned document patches; (e) constructed target rectangles; (f) rectified document image.

estimated as

$$H = A^{-1} \bullet R \tag{15}$$

where H is the homography matrix and matrixes A, R are constructed using four point correspondences. The three matrixes take the following forms

$$A = \begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1' \bullet x_1 & x_1' \bullet y_1 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & y_1' \bullet x_1 & y_1' \bullet y_1 \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2' \bullet x_2 & x_2' \bullet y_2 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & y_2' \bullet x_2 & y_2' \bullet y_2 \\ -x_3 & -y_3 & -1 & 0 & 0 & 0 & x_3' \bullet x_3 & x_3' \bullet y_3 \\ 0 & 0 & 0 & -x_3 & -y_3 & -1 & y_3' \bullet x_3 & y_3' \bullet y_3 \\ -x_4 & -y_4 & -1 & 0 & 0 & 0 & x_4' \bullet x_4 & x_4' \bullet y_4 \\ 0 & 0 & 0 & -x_4 & -y_4 & -1 & y_4' \bullet x_4 & y_4' \bullet y_4 \end{bmatrix},$$

$$H = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix}, \quad R = \begin{bmatrix} x'_1 \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \\ x_4' \\ y_4' \end{bmatrix}, \tag{16}$$

where 3×3 homography matrix is expressed in vector form and h_{33} is equal to 1 under homogeneous frame. We determine four point correspondences $\{(x_i, y_i), (x_i', y_i')\}$, $i=1,\ldots,4$, as the four vertices of the partitioned quadrilateral patch and constructed target rectangle. For sample word 'laboratory' given in Fig. 3(a), Figs. 5(d) and 6(a) illustrate one example of quadrilateral correspondence and the related vertices. With the estimated homography, the sample word can be finally restored in Fig. 6(b).

Fig. 7 illustrates the restoration process where Fig. 7(a) gives a camera image of smoothly curved document. Based on the proposed VSB identification technique, VSBs are firstly identified in Fig. 7(b). The $\boldsymbol{x}$ line and base line of text are then fitted using the classified character extremum points in Fig. 7(c). With the identified VSBs and the fitted $\boldsymbol{x}$ lines and base lines, distorted camera document is finally partitioned into multiple quadrilateral patches in Fig. 7(d). Target rectangles are then constructed based on the enclosed characters and the specific aspect ratios. Target rectangles are arranged horizontally line by line where the bottom edge of all rectangles of the same text line lies on a horizontal line and the adjacent rectangles share a common vertical edge in Fig. 7(e). Global distortion is lastly removed through the local rectification of the partitioned document patch one by one in Fig. 7(f).

It should be noted that character ascender and descender are actually above or below of partitioned image patches given in Fig. 7(d). Therefore, these character ascender and descender will disappear within the restored document image if the transformation is conducted within target rectangles. To rectify character ascender and descender correctly, the transformation must be extended above and below a bit. In our proposed method, the extension range is defined as half of the average size of identified VSBs so that transformation extension is able to cover character ascender and descender areas and at the same time, it will not reach adjacent text lines.

Non-text document components such graphics and mathematical equations can be partitioned as well. The position and orientation of cubic polynomials can be estimated through interpolation of $\boldsymbol{x}$ lines and base lines of two nearest text lines. The position and orientation of vertical partition lines can be directly determined based on the VSBs of the nearest text line. As Fig. 7(d) shows, the vertical partition lines are imported from the VSBs of the first text line below the graphic component. The aspect ratios of the target rectangles can be determined the same as that of the partitioned document patches of that text line as well. The middle parts in Fig. 7(d) and (e) show the partitioned document patches of graphic region and the constructed target rectangles.

3. Experiment results and discussion

We implement the proposed document restoration technique based on the methods described above. The programs are written in C++ and run on a personal computer equipped with Window XP and Pentium 4 CPU. The system was evaluated with an image database that contains 90 camera images of documents printed in English. Sample documents are collected from book pages, proceedings, and printed paper sheets and texts are printed in different fonts including Time New Roman, Arial, and Verdana. Book pages within thick bound volumes originally lie over a smoothly curved surface and they are captured directly using a digital camera. Printed proceedings and paper sheets are manually put over certain smoothly curved surfaces and captured again. Experimental results show that our restoration method is fast and the restored document text is friendly to the generic OCR systems.

All sample documents are captured using a digital camera of seven mega pixels. Currently, it takes around 18 s on average for document restoration. Through code optimization, there is still a big room for speed improvement. The execution time can be roughly divided into three parts, namely, adaptive document thresholding, connected component labeling, and our proposed document partition and restoration. Among the three, the first two occupy most of execution time. Considering the big image size (around 3000×2000) and the time-consuming adaptive binarization process, our method is fairly fast and has the potential for real time applications.

We use character recognition rate to evaluate the restoration performance because OCR is one of the main applications and the subsequent document indexing and retrieval depend heavily on OCR results as well. Recognition experiments are conducted using the OCR software FineReader. Ninety distorted camera images are first input to recognition system one by one and the recognition rates are recorded. Distorted camera documents are then restored using our proposed method and the restored document images are fed the FineReader again for OCR test. Fig. 8 gives the recognition rates of 90 camera documents before and after the restoration operation.

The dotted line labeled with square in Fig. 8 give the recognition rates of 90 camera documents before the restoration. The lower recognition rates (<30% on average) can be expected since generic OCR systems can not deal with camera documents with perspective and geometric distortions. More specifically, most of text regions are falsely recognized as graphic due to the large skew angle. The solid curve labeled with diamond in Fig. 8 gives the recognition rates of document images restored using our method. As figure shows, the average recognition rate reaches over 90%. The recognition rate is so high because generic OCR systems are tolerant of slight variation of aspect ratio of character images. Therefore, restored document text can be recognized correctly even though the aspect ratios of some target rectangle are not accurately estimated.

Our method works well on most frequently used text fonts including Time New Roman, Arial, and Verdana because text font does not affect VSB identification and ***x*** line and base line fitting in most cases. At the same time, most binarization noises of small sizes are removed using the size filter described in Section 2.1.1. However, our method depends heavily on the resolution of camera images of document. The reason is that the orientation of vertical partitioning line cannot be estimated accurately if characters are too small. Furthermore, the binarization of camera images with poor resolution may introduce a large number of broken and touching character components, which affect the VSB identification and subsequent target rectangle construction. Fortunately, this problem can be alleviated through the image interpolation, which enlarges camera images if the character size is too small.

Compared with the reported techniques, our method requires no auxiliary hardware [14–16], no unreliable document patterns such as highly-contrasted document boundary or paragraph formatting [12,19], and the only thing required is a document image captured using a common digital camera. The three restoration examples in Fig. 9 illustrate the advantages of our proposed restoration method. Unlike the reported methods that can only handle camera document lying over a planar [9–13] or smoothly curved surface [17–19], our method can rectify camera image of planar documents automatically though it targets the flattening of smoothly curved documents. For camera images of planar documents, VSBs and ***x***/base line can be determined using the same techniques. Quadrilateral partition and the subsequent restoration can be implemented in a similar way. For the planar document patch given in Fig. 9(a) and (b) show the document image restored using our method.

Our method can handle camera documents with different curvatures [17,18]. Fig. 9(c) gives a document patch lying over a concave surface. For camera documents of this type, the shape of text lines can be modeled using a cubic polynomial as well. Fig. 9(d) shows the document image restored using our method. Furthermore, our method can be adapted to handle certain documents with more complex curvature where the shape of text lines cannot be modeled using a single unimodal cubic polynomial. For camera documents of this type, piecewise polynomial or B-spline can be exploited for ***x***/base line fitting and the subsequent document partitioning. For the document patch given in Fig. 9(e), two pieces of cubic polynomials are exploited for ***x*** line and base line fitting. Fig. 9(f) gives the restored document image based on our proposed method.

Currently, our method can only work on the documents printed in Latin-based languages such as English, French, and German. For documents printed in other scripts such as Arabic, VSBs may not be identified properly. With the same reason, our method may not be able to restore the handwritten documents where VSBs are not available. Some new restoration approaches will be investigated for the restoration of these two types of camera documents.

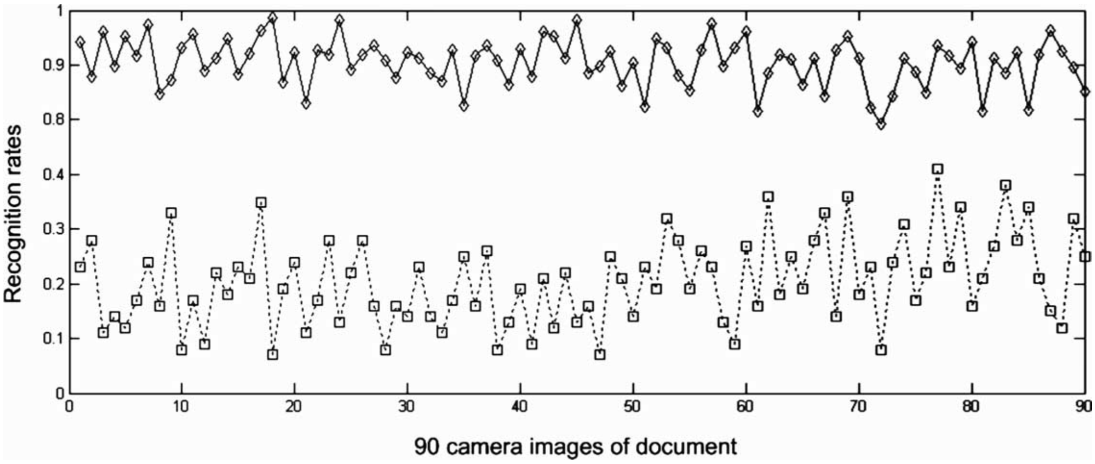


Fig. 8. Recognition rate before and after restoration.

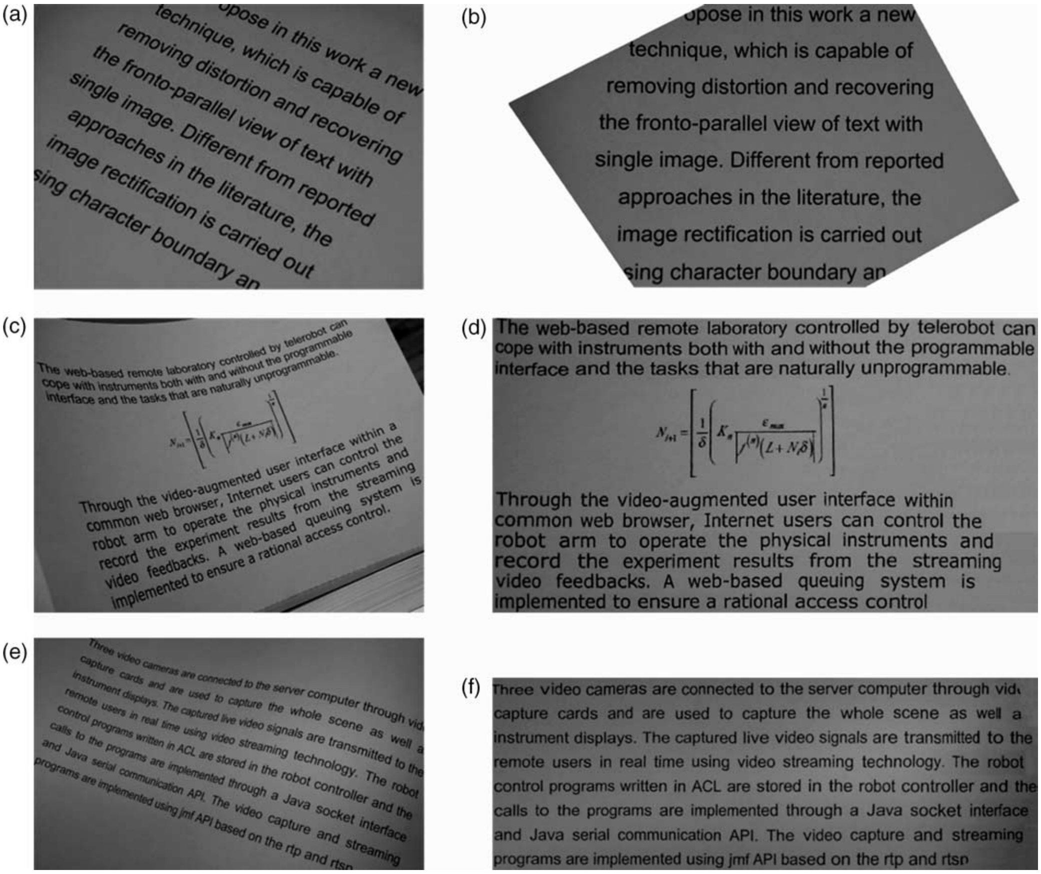


Fig. 9. Document restoration: (a) camera document lying over a planar surface; (b) restored document image; (c) camera document lying a concave surface; (d) restored document image; (e) camera document with more complex curvature; (f) restored document image.

4. Conclusion

In this paper, a computationally efficient technique is proposed to restore the camera images of document lying over a planar or smoothly curved surface. The restoration is carried out through image partition, which divides camera documents into multiple quadrilateral patches through the exploitation of the identified VSBs and the fitted $\boldsymbol{x}$ lines and base lines of text. Compared with the reported document restoration techniques, our method needs no auxiliary hardware, no camera calibration, no complicated 3D reconstruction and the only thing required is a single document image captured by a common digital camera. Experimental results show that proposed restoration process is fast and easy for implementation. With a digital camera, the proposed document restoration technique may open a new channel for document capture and understanding. Furthermore, it may be applied to some other portable devices such as the mobile phone and the personal digital assistant (PDA) with the increase of sensor resolution. As a result, these camera-sensor-embedded devices may need only to store and transmit recognized ASCII text instead of the document images with a huge size.

References

[1] H.K. Kwag, S.H. Kim, S.H. Jeong, G.S. Lee, Efficient skew estimation and correction algorithm for document images, Image and Vision Computing 20 (2002) 25–35.

[2] B. Yu, A.K. Jain, A robust and fast skew detection algorithm for generic documents, Pattern Recognition 29 (1996) 1599–1629.

[3] E. Kavallieratou, N. Fakotakis, G. Kokkinakis, Skew angle estimation for printed and handwritten documents using the Wigner–Ville distribution, Image and Vision Computing 20 (2002) 813–824.

[4] L. O'Gorman, The document spectrum for page layout analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (1993) 1162–1173.

[5] P.Y. Yin, Skew detection and block classification of printed documents, Image and Vision Computing 19 (2001) 567–579.

[6] O. Okun, M. Pietikainen, J. Sauvola, Document skew estimation without angle range restriction, International Journal on Document Analysis and Recognition 2 (1999) 132–144.

[7] U. Pal, B.B. Chaudhuri, Multi-oriented text lines detection and their skew estimation, Third Indian Conference on Computer Vision, Graphics and Image Processing, Multi-Oriented Text Lines Detection, Their Skew Estimation, India, 2002, pp. 270–275.

[8] M.B.H. Ali, An object/segment oriented skew-correction technique for document images, IEEE Conference on Document Analysis and Recognition, Ulm, Germany, 1997, pp. 671–674.

[9] M. Pilu, Extraction of illusory linear clues in perspectively skewed documents, IEEE Computer Vision and Pattern Recognition Conference, Kauai, Hawaii, 2001, pp. 363–368.

[10] S.J. Lu, B.M. Chen, C.C. Ko, Perspective rectification of document images using fuzzy set and morphological operations, Image and Vision Computing 23 (2005) 541–553.

[11] C.R. Dance, Perspective estimation for document images, Proceedings of the SPIE Conference on Document Recognition and Retrieval IX, San Jose CA, 2001, pp. 244–254.

[12] P. Clark, M. Mirmhedi, Rectifying perspective views of text in 3D scenes using vanishing points, Pattern Recognition 36 (2003) 2673–2686.

[13] G. Myers, R. Bolles, Q.T. Luong, J. Herson, Recognition of text in 3-D scenes, Proceedings of the Fourth Symposium on Document Image Understanding Technology, Columbia, MD, 2001, pp. 85–99.

[14] M.S. Brown, W.B. Seales, Document restoration using 3D shape: a general deskewing algorithm for arbitrarily warped documents, IEEE Conference on Computer Vision, Vancouver, 2001, pp. 117–124.

[15] M. Pilu, Undoing paper curl distortion using applicable surfaces, IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2001, pp. 67–72.

[16] Y.A. Kawarago, A. Kaneko, T. Miura, K.T. Shizuoka, Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system, IEEE Conference on Pattern Recognition. 2004, pp. 23–26

[17] H. Cao, X. Ding, C. Liu, A cylindrical surface model to rectify the bound document image, IEEE International Conference on Computer Vision, Nice, France 1, Vol. 1, 2003, pp. 228–233

[18] Z. Zhang, C.L. Tan, Correcting document image warping based on regression of curved text lines, IEEE Conference on Document Analysis and Recognition Vol. 1, Edinburgh, Scotland, 2003, pp. 3–6.

[19] Y.C. Tsoi, M.S. Brown, Geometric and shading correction for images of printed materials a unified approach using boundary, IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 240–246.

[20] J. Liang, D. DeMenthon, D. Doermann, Flattening curved documents in images, IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 338–345.

[21] D. Hearn, M.P. Baker, Computer Graphics, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[22] W. Niblack, An Introduction to Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1986. pp. 115–116.

[23] Ø.D. Trier, T. Taxt, Evaluation of binarization methods for document images, IEEE Transactions on Pattern Analysis and Machine Intellegence 17 (1995) 312–315.

[24] P. Soille, Morphological Image Analysis: Principles and Applications, second ed., Springer, New York, 2003.

[25] L.A. Zadeh, Calculus of Fuzzy Restrictions, Fuzzy Sets and Their Application to Cognitive and Decision Making Processes, Academic Press, San Diego, 1975.

[26] H.J. Zimmermann, P. Zysno, Latent connectives in human decision making, Fuzzy Sets and Systems 4 (1980) 37–51.

[27] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, New York, NY, 2000.