Camera-Based Document Analysis and Recognition

Proceedings of the

First International Workshop on Camera-Based Document Analysis and Recognition

August 29, 2005

Olympic Parktel, Seoul, Korea

Edited by

Koichi Kise
Osaka Prefecture University, Japan

David S. Doermann
University of Maryland, USA

CBDAR 2005

Table of Contents

Oral Papers

Section I. Image Processing

Section II. Recognition

Section III. Systems

Poster Papers

Keynote Presentation

Welcome from the CBDAR2005 Co-Chairs

It is our great honor to welcome you to the First International Workshop on Camera-Based Document Analysis and Recognition, CBDAR2005.

Pervasive use of camera phones and hand-held digital still and video cameras have led consumers to discover that image-based recording of information by just pressing a button is very convenient. In addition to imaging faces and scenes, people have started capturing documents as an alternative to note taking. Cameras, which are now functioning as personal copiers, will soon produce a significant numbers of imaged documents that will be difficult to handle manually. Although traditional techniques developed in the field of document analysis and recognition provide us with a good starting point, they cannot be directly used on camera captured images. This leads us to a new sub-field of research — camera-based document analysis and recognition (CBDAR). The workshop is indented as a forum for presenting and discussing up-to-date issues and techniques of the CBDAR field.

The submitted papers reflect the diverse nature of the field including image processing such as mosaicing, dewarping, superresolution and tracking; recognition of characters in degraded images as well as data-embedding mechanism for helping recognition; systems for capturing, retrieving and reading documents and text in camera-captured images as well as methods for performance evaluation. We also have excellent poster papers and demos in the above and related fields of research.

It is our hope that this workshop will bring together researchers and developers from various backgrounds and help stimulate new ideas and new research directions in this emerging field of camera-based document analysis and recognition.

We know that the success of the workshop depends on the many people who worked together in planning and organizing it. We would like to thank members of the program committee for their thorough and timely reviewing of the papers and constructive ideas for defining the format of the workshop; all contributing authors for their valuable work; Prof. Jin Hyung Kim (KAIST; the conference chair of ICDAR2005), Prof. Henry S. Baird (Lehigh Univ.; the workshop chair of ICDAR2005), and other ICDAR2005 organizers for their support and generous help; Dr. Hiromichi Fujisawa (Hitachi CRL) for his support in organizing the workshop; Mr. Yang Wang (Univ. of Maryland) and his superb system for handling submitted papers called UMIACS Echelon, Dr. Masakazu Iwamura (Osaka Prefecture Univ.) for his work on both designing the logo of the workshop as well as preparing and maintaining the Web, mailing lists and the proceedings.

We look forward to a successful workshop and the beginning of an intensified research in the area.

CBDAR2005 Co-Chairs
Koichi Kise and David S. Doermann

CBDAR2005 Program Committee

Co-Chairs

Koichi Kise, *Osaka Prefecture University, Japan*

David S. Doermann, *University of Maryland, USA*

Program Committee Members

Thomas Breuel, *DFKI, Germany*

Yeongwoo Choi, *Sookmyung Women's University, Korea*

Daniel DeMenthon, *University of Maryland, USA*

Andreas Dengel, *DFKI, Germany*

Masashi Koga, *Hitachi CRL, Japan*

Simon M. Lucas, *University of Essex, UK*

Majid Mirmehdi, *University of Bristol, UK*

Gregory K. Myers, *SRI International, USA*

Shinichiro Omachi, *Tohoku University, Japan*

Jun Sun, *Fujitsu R&D Center, China*

Chew Lim Tan, *NUS, Singapore*

Seiichi Uchida, *Kyushu University, Japan*

Assistants

Masakazu Iwamura, *Osaka Prefecture University, Japan*

Yang Wang, *University of Maryland, USA*

Section I

Image Processing

Mosaicing-by-recognition for recognizing texts captured in multiple video frames

Seiichi Uchida, Hiromitsu Miyazaki, and Hiroaki Sakoe
Graduate School of Information Science and Electrical Engineering,
Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka-shi, 812-8581 Japan

Abstract

Text recognition in video frames is promising because of its following superiorities over text recognition in a still camera image: (1) it is possible to recognize longer texts by concatenating the frames, and (2) it is also possible to improve the quality of the text image by integrating the frames. In this paper, a mosaicing-by-recognition technique is proposed where video mosaicing and text recognition are simultaneously and collaboratively performed in a one-step manner by a dynamic programming-based optimization algorithm. In this optimization algorithm, rotation, scaling, vertical shift, and speed fluctuation of camera motion are efficiently compensated. The results of experiments to evaluate not only the accuracy of text recognition but also that of video mosaicing indicates that the proposed technique is practical and can provide reasonable results in most cases.

1. Introduction

Text recognition for a single image captured by a camera, i.e., a still image, becomes a practical technique and is often equipped in commercial cellular phones for recognizing e-mail addresses, URLs, single words, and so on. In spite of its practical property, it has several limitations. For example, (1) long texts often cannot be recognized, and (2) it is generally dif cult to improve the quality (e.g., resolution and noise level) of a still image.

Text recognition for multiple video frames (Fig. 1) has been investigated [1] as an alternative to text recognition in a still image, because it has a potential to overcome the above limitations. That is, it is possible to recognize longer texts by mosaicing consecutive frames, i.e., by matching and concatenating the frames. In addition, it is also possible to improve the quality of the text image (e.g., super-resolution, noise removal) by utilizing overlapped ares between consecutive frames.

In this paper, a *mosaicing-by-recognition* technique is proposed. Previous attempts to recognize texts in video sequences generally assumes a two-step manner that video frames are rstly concatenated into one large image by mosaicing techniques (e.g., [2]) and then the texts in the large mosaic image is recognized. In contrast, the proposed technique is organized in a one-step manner that video mosaicing and text recognition are simultaneously and collaboratively performed. Speci cally , multiple frames capturing a long text line are optimally matched and concatenated with a guide of the text recognition framework. The optimization is performed by a dynamic programming (DP)-based algorithm while compensating various distortions of the frames.

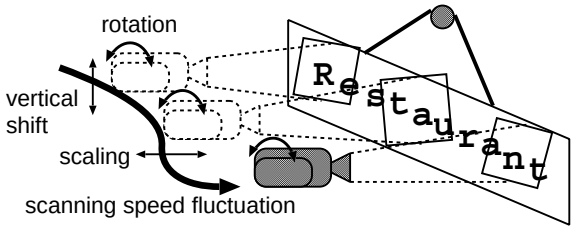


Figure 1. Recognition of the text captured in multiple video frames. When using a hand-held camera, several distortions will appear in the frames due to rotation, vertical shift, scaling, and scanning speed fluctuation.

2. Mosaicing-by-recognition

2.1. Problem formulation

Assume that a long text line is continuously and fragmentarily captured in video frames by a hand-held camera which moves from left to right along the text. Major distortions appeared in the frames are: rotation, scaling, vertical shift, and speed uctuatio n of the camera motion. Our task is the recognition of the captured texts while mosaicing the frames and removing the distortions.

In the remaining part of this section, we will rstly discuss a *simple case* that video frames undergo only speed

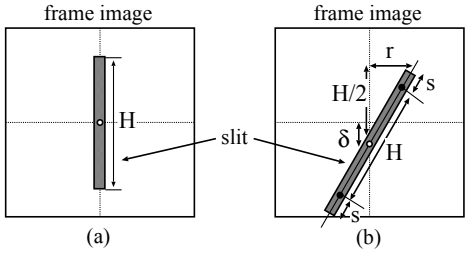


Figure 2. (a) One-pixel slit ($r = s = \delta = 0$) and (b) its controlled version.

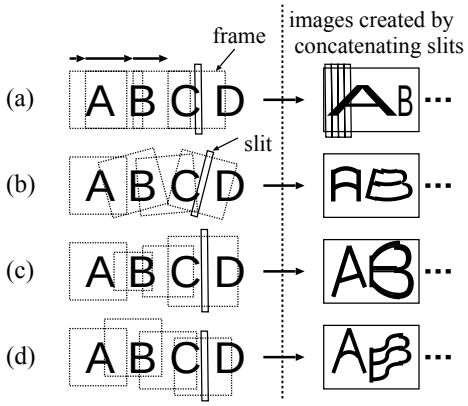


Figure 3. Major distortions in video sequence obtained by a hand-held camera. (a) Scanning speed fluctuation. (b) Rotation. (c) Scaling. (d) Vertical shift. The rightmost images of (b)–(d) indicate the necessity of controlling slit shape.

uctuatio n. This simpli cation is quite useful to grasp the basic principle of the proposed technique. In fact, by this simpli cation our mosaicing-by-recognition problem is reduced to a well-known *segmentation-by-recognition* problem for continuous speeches [3] and texts [4]. Secondly, we will discuss a *general case* that video frames undergo not only speed uctuation but also rotation, scaling, and vertical shift. The mosaicing-by-recognition problem on the general case is derived as an extension of the simple case.

Our discussion is further simpli ed by the use of a *one-pixel slit* (shown in Fig. 2(a)), which is a central part of the frame and has 1 pixel width and H pixel height. Although this simpli cation is useful to understand the principle of the proposed technique, most of information contained in frames is disregarded. Thus, the use of *wider slits* whose width is two or more pixels is discussed in Section 2.4.

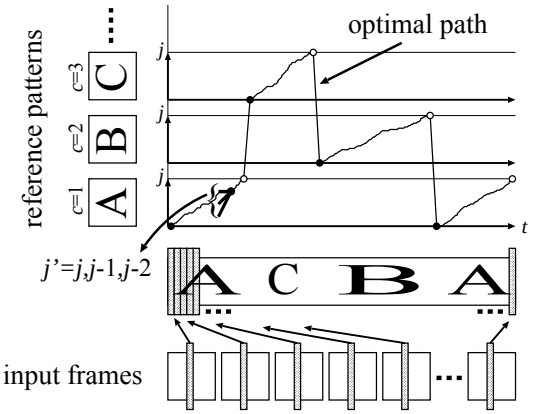


Figure 4. Mosaicing-by-recognition for the simple case that video frames undergo only speed fluctuation.

/* Initialization */
for $c := 1$ **to** C **do begin**
  $g_1(c,1) := d_1(c,1)$
  for $j := 2$ **to** J_c **do**
    $g_1(c,j) := \infty$
end
$D_1 := \infty$
/* DP Recursion */
for $t := 2$ **to** T **do begin**
  for $c := 1$ **to** C **do begin**
    $g_t(c,1) := d_t(c,1) + \min\{g_{t-1}(c,1), D_{t-1}\}$
    for $j := 2$ **to** J_c **do**
      $g_t(c,j) := d_t(c,j) + \min_{j' \in \{j,j-1,j-2\}} g_{t-1}(c,j')$
  end
  $D_t := \min_{c' \in C} g_t(c', J_{c'})$
end

Figure 5. The DP algorithm for mosaicing-by-recognition for the simple case. Several steps for backtracking operation is omitted.

2.2 DP algorithm for simple case

In this section, a mosaicing-by-recognition algorithm for the simple case is provided, where only the uctuation of scanning speed is assumed. The one-pixel slit is also assumed here. Other distortions and wider slits will be considered in later sections.

On the simple case, the problem is reduced to the well-known optimization problem, called segmentation-by-

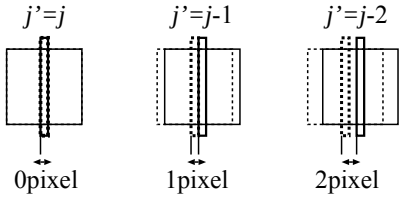


Figure 6. The relation between the selection of j' and scanning speed.

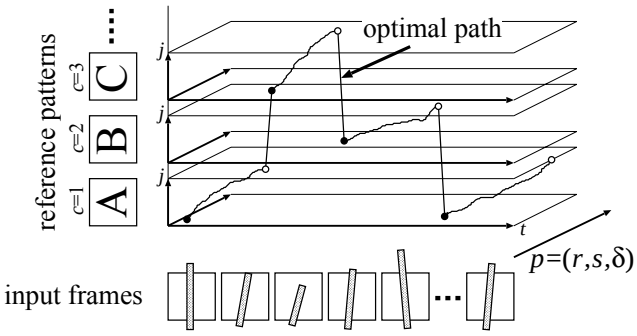


Figure 7. Mosaicing-by-recognition for the general case that video frames undergo not only speed fluctuation but also rotation, scaling, and vertical shift.

recognition problem, of a character sequence. The text contained in the frames can be treated as a deformed character sequence in the image created by concatenating the one-pixel slits of all T frames (shown in the right side of Fig. 3(a)). Thus, the text in the image can be recognized and partitioned into its component characters by solving an optimal path problem on the search space indexed by t and (c, j), where $c \in \{1, \ldots, C\}$ is the character category and $j \in \{1, \ldots, J_c\}$ is the index for the row of the reference pattern image of the category c (Fig. 4).

It is also well-known that this optimal path problem can be solved effectively by DP. Figure 5 shows a DP algorithm for the simple case, where $d_t(c, j)$ is the matching cost between the one-pixel slit of the tth frame and the jth column of the reference pattern of category c. The value $g_t(c, j)$ is the minimum cost accumulated along with the optimal path to the point (so-called the "state") indexed by t, c and j.

The speed uctuation can be compensated by controlling j' in the DP recursion of Step 11. Speci cally , as shown in Fig. 6, $j' = j - 2$ is selected when the scanning speed is 2 pixel/frame and $j' = j$ is selected when it is 0pixel/frame.

The result of character recognition is obtained by backtracking the optimal (c, j)-sequences (illustrated as the optimal path in Fig. 4) after performing the DP algorithm. An optimal mosaic image is also obtained by backtracking as will be shown in the Section 2.5. Thus, the mosaicing of video frames is optimized simultaneously with the text recognition, and therefore we call the above procedure *mosaicing-by-recognition*.

2.3 DP algorithm for general case

In this section, we derive a DP algorithm for the general case, where not only the speed uctuation but also the other distortions are considered. The DP algorithm for the general case is an extension of the foregoing DP algorithm for the simple case. The main idea of the extension is to control (i.e., rotate, scale, and vertical shift) the slit according to the distortions. Figure 2(b) shows a slit controlled by three parameters r, s, and δ which represents rotation, scaling, and vertical shift, respectively. When $r = s = \delta = 0$, the controlled slit is reduced to the original slit of Fig. 2(a) and means that no distortion appears.

The optimal parameters are searched for in the DP framework. Speci cally , as shown in Fig. 7, the problem becomes an optimal path problem in the search space in-

/* Initialization */
1 **for all** $p \in \{(r, s, \delta)\}$ **do begin**
2 **for** $c := 1$ **to** C **do begin**
3 $g_1(p, c, 1) := d_1(p, c, 1)$
4 **for** $j := 2$ **to** J_c **do**
5 $g_1(p, c, j) := \infty$
6 **end**
7 $D_1(p) := \infty$
8 **end**
/* DP Recursion */
9 **for** $t := 2$ **to** T **do begin**
10 **for all** $p \in \{(r, s, \delta)\}$ **do begin**
11 **for** $c := 1$ **to** C **do begin**
12 $g_t(p, c, 1) := d_t(p, c, 1) + \min_{p' \in \mathrm{pre}(p)} \{g_{t-1}(p', c, 1), D_{t-1}(p')\}$
13 **for** $j := 2$ **to** J_c **do**
14 $g_t(p, c, j) := d_t(p, c, j) + \min_{\substack{p' \in \mathrm{pre}(p) \\ j' \in \{j, j-1, j-2\}}} g_{t-1}(p', c, j')$
15 **end**
16 $D_t(p) := \min_{c' \in C} g_t(p, c', J_{c'})$
17 **end**
18 **end**

Figure 8. The DP algorithm for the general case.

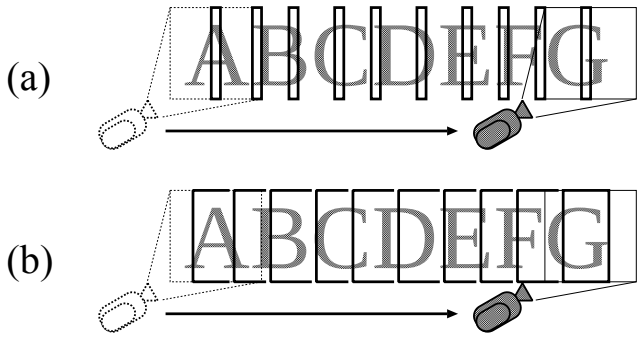


Figure 9. The relation between slit width and scanning speed.

dexed by t and (p, c, j), where $p = (r, s, \delta)$ is a parameter vector. Figure 8 shows a DP algorithm the general case, where $d_t(p, c, j)$ is the matching cost between the one-pixel slit whose shape is controlled by the parameter p and the jth column of the reference pattern of category c. In the DP recursion of Step 14, the smoothness of the distortion is assumed by constraining the parameter vectors of consecutive frames (p for frame t and p' for frame $t-1$) by

$$\mathrm{pre}(p) = \{p' = (r', s', \delta') \mid r - 1 \le r' \le r + 1, \\ s - 1 \le s' \le s + 1,\ \delta - 1 \le \delta' \le \delta + 1\}$$

The computational complexity of the algorithm is $O(TCJRS\Delta)$, where R, S and Δ are the ranges of r, s, and δ, respectively. Similar to the simple case, the result of character recognition is obtained by backtracking the optimal path after performing the DP algorithm.

2.4. Expansion of slit width

In the above discussion, the width of the slit is x ed at one pixel for simplifying the problem. This means, however, that most of information contained in each frame is disregarded.

The modi catio n of the above algorithms for using a wider slit is very straightforward. Speci cally , the modi cation can be done by simply changing the de nition of the matching distance $d_t(c, j)$ to be a distance between the wider slit (a rectangular area on a input frame) and a rectangular area of a reference pattern[1]. Using a wider slit, the recognition accuracy can be improved because "over- tting" and "over-segmentation" can be suppressed as discussed in Section 3.

[1]Strictly speaking, the projective distortion within a wider slit should be considered for recognizing texts captured from a non-frontal video camera.

A wider slit produces another promising effect; a wider slit allows higher scanning speeds. Figure 9 shows the relation between slit width and acceptable scanning speed. As shown in Fig. 9 (a), when the one-pixel slit is used, non-negligible gaps will appear in captured frames as scanning speed becomes higher. Thus, most information will be lost and the accuracy of recognition/mosaicing results will be seriously decreased. On the other hand, as shown in Fig. 9 (b), when a wider slit is used, the gaps will disappear because some overlaps between consecutive frames can be expected. For allowing scanning speeds of K pixel/frame, the DP recursions of the above algorithms (i.e., Step 11 of Fig. 5 and Step 14 of Fig. 8) also should be modi ed so that j' can be chosen not only from $\{j, j-1, j-2\}$ but also from $\{j-3, \ldots, j-K\}$.

2.5 Mosaic image

Although conventional video mosaicing techniques require several corresponding points among consecutive frames, the proposed technique does not. In the simple case, the mosaic image can be obtained by placing the tth frame with a 0$\sim$ K pixel horizontal shift according to the relation between j' and j, which can be obtained by the backtracking operation for the optimal path. (See Fig. 6 for the case $K = 2$.)

Even in the general case, the mosaic image can be obtained by a similar procedure. The only difference is a dewarping operation of the controlled slit of each frame is necessary in advance to placing it with a 0$\sim$ K pixel horizontal shift. The dewarping can be done by using the optimal parameter p at frame t, which can be obtained by the backtracking operation.

On creating a mosaic image by the above procedures, we should manage the overlapped area between two consecutive frames. In the following experiment, a simple averaging was performed to determine a pixel value of the overlapped area. In future, this overlapped area will be utilized to improve the quality of the mosaic image by super-resolution or other image restoration techniques [5, 6, 7].

3. Experimental results

3.1. Data preparation

As test samples for performance evaluation, 20 text lines printed on white A4-sized papers were prepared. Each text line contains about 50 characters (of capital/small English alphabets and digits) and thus about 1000 characters were prepared in total. Each character was printed in the same Times-Roman font. The character height ($\sim H$) in the frame was about 40 pixels.

Each text line was then captured in multiple frames by moving a video camera. A special equipment with a variable speed motor was used for moving the camera horizontally. Thus, we could uctuate the speed of camera movement, while excluding rotation, scaling, and vertical shift. According to this manner, the video frames of the simple case were prepared. Note that naive gray-level was used as the pixel feature for calculating the matching cost $d_t(c, j)$ or $d_t(p, c, j)$

For preparing the video frames of the general case, the above video frames of the simple case were arti cially rotated, scaled and vertically shifted. That is, the video frames for the general case was synthesized from those of the simple case. On the synthesis, the maximum amplitude of distortions were limited so that the distortions can be theoretically compensated by $p = \{(r, s, \delta) \mid |r| \leq k, |s| \leq k, |\delta| \leq k\}$, where k was xed at 1, 2, 3, or 4 (pixels).

3.2. Qualitative analysis

Figure 10 shows a result of the simple case. The one-pixel slits were used here to observe the minimum performance of the algorithm. The camera scanning speed was uctuated between 0$\sim$ 2 pixel/frame. Figure 10 (a) shows several input frames and (b) shows the image created by concatenating the one-pixel slits. This gure (b) indicates that scanning speed became very low around "t" of the word "Character". Figures 10 (c) and (d) show the mosaic image and the recognition result. While most part of the mosaic image was successfully created, several misrecognitions can be observed. The misrecognitions were mainly due to segmentation errors, called *over-segmentation*, such that "m" is misrecognized as "r" and "n". The misrecognitions of this type are often found in the results of segmentation-by-recognition techniques. A well-known remedy for this problem is the use of a word lexicon. The use of a wider slit is also effective to suppress such misrecognitions as will be shown later.

Figure 11 shows a result of the general case where the one-pixel slits were used. As noted in Section 3.1, the video frames were synthesized from the video frames used in the above simple case experiment. (That is, the scanning speed uctuation between 0$\sim$ 2 pixel/frame was appeared together with rotation, scaling, and vertical-shift.) While most part of the mosaic image (c) is well created, the part around misrecognitions shows degradation. For example, the last character "o" is deformed to be close to "v" by abusing the e xibility on controlling slits. Thus, this misrecognition ("o"$\rightarrow$ "v") is caused by so-called *over-fitting*, which often degrades the performance of elastic matching-based character recognition (e.g., [8]). The use of some sophisticated pixel feature (e.g., directional feature, background feature, crossing feature, localized moment feature, etc.), a

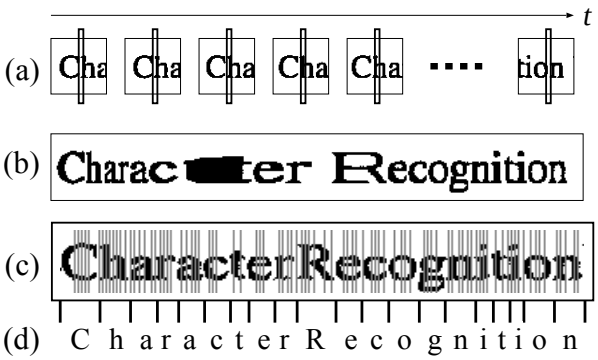


Figure 10. Result of the simple case. The original text is "Character Recognition". (a) Input video frames with one-pixel slits. (b) Image created by simply concatenating their one-pixel slits. (c) Mosaic image and (d) recognition result provided by the simple case algorithm.

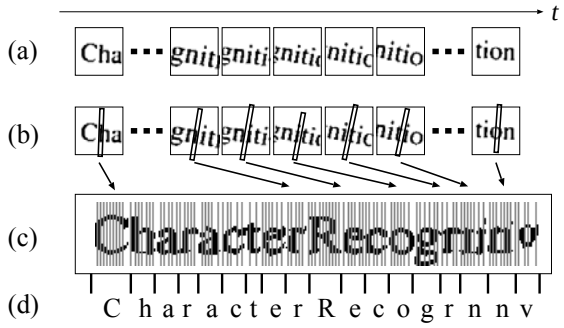


Figure 11. Result of the general case. The original text is "Character Recognition". (a) Input video frames. (b) The optimally controlled one-pixel, (c) mosaic image, and (d) recognition result, provided by the general case algorithm.

word lexicon, and a wider slit will be still helpful to reduce such misrecognitions due to over- tting.

3.3. Quantitative analysis

Figure 12 shows the recognition rates for the general case. A wider slit with 20 pixel width was used here. The camera scanning speed was uctuated between 0$\sim$ 2 pixel/frame. This result shows that the proposed technique could provide recognition rates over 95% even when the video frames undergo scaling and vertical shift. Considering that we only use a naive gray-level feature to obtain

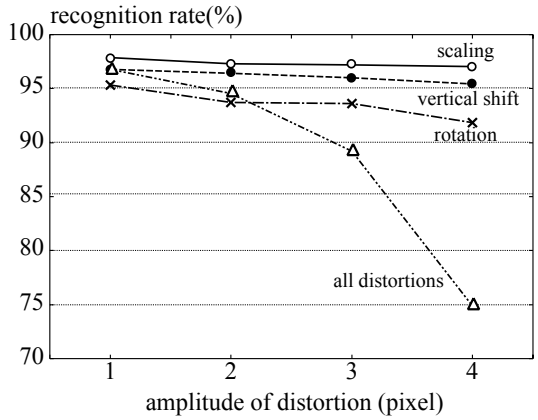


Figure 12. Recognition rate for the general case. The horizontal axis represents the amplitude of distortions, k (pixels). The slit width W was fixed at 20.

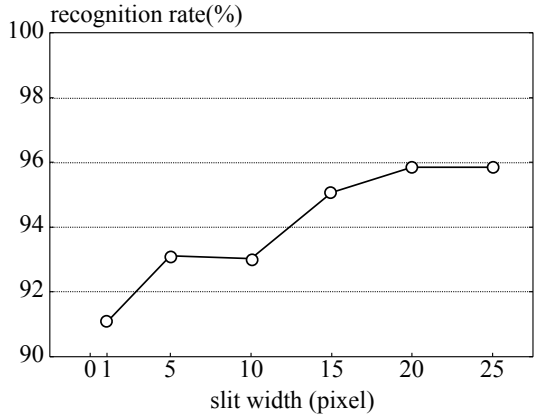


Figure 13. Recognition rates as a function of slit widths W. Here, the simple case with scanning speed fluctuation between 0$\sim$ 2 pixel/frame was assumed.

matching score $d_t(c, j)$, those rates are acceptable one. The recognition rates were degraded by rotation. The reason of this degradation was quantization errors on dewarping to compensate the rotation. Thus, this degradation can be minimized by using blurring operation, local perturbation matching, invariant feature, and so on.

Figure 13 shows the effect of slit width W on recognition accuracy. This result was of the simple case; that is, only camera scanning speed uctuation (0$\sim$ 2 pixel/frame) was imposed. The constant K which de nes the acceptable scanning speed was xed at 2. The result shows that recognition accuracy is improved by increasing W to 20 pixels, i.e., about half of average character width. When

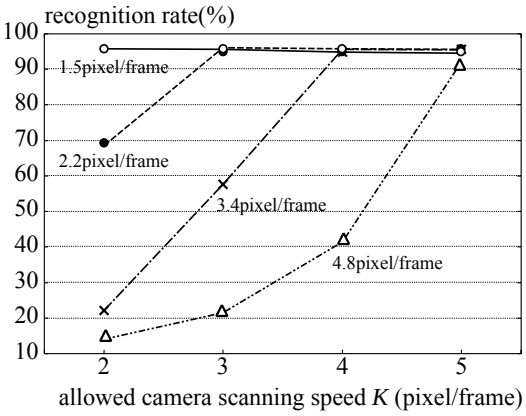


Figure 14. Recognition rates at higher camera scanning speeds. ($W = 20$)

using wider slits, the misrecognitions between similar characters (e.g., "I" to "1") and the misrecognitions due to oversegmentation (e.g., "m" to "r"+"n") were successfully reduced.

Figure 14 shows the recognition rates when higher camera scanning speeds were allowed by using larger K. In this experiment, the video frames whose scanning speed was x ed at about 1.5, 2.2, 3.4, or 4.8 pixel/frame were used. No geometric distortion (rotation, scaling, and vertical-shift) was imposed on those frames. The result of Fig. 14 clari es that K should be x ed at larger values for compensating higher scanning speeds. For example, when the scanning speed is 4.8 pixel/frame, K should be set at 5 or more. Conversely, when K was smaller than the scanning speed, the recognition rate was seriously degraded.

4. Conclusion and future work

A mosaicing-by-recognition technique was proposed for recognizing texts in multiple video frames and mosaicing those frames. Those two procedures, i.e., recognition and mosaicing, are simultaneously and collaboratively performed in a one-step manner by a DP-based optimization algorithm. Experimental results showed that the proposed technique can attain about 90% character recognition rate even when rotation, scaling, vertical shift, and speed uctuation appear in the frames.

Future work will focus on the following points:

- *Lexicon*: The proposed technique often produces misrecognitions by over-segmentations (e.g., "m"$\rightarrow$ "r" and "n") and over- tting (e.g., "o" $\rightarrow$ "v"). Like the other text recognizer based on segmentation-by-recognition framework, the use of lexicon will be help-

ful to exclude such misrecognitions.

- *Sophisticated pixel feature*: In the experiment conducted in this paper, only naive pixel feature, i.e., gray-level feature, was used. Since this feature is very weak to geometrical distortions, sophisticated pixel features, such as directional feature, background feature, crossing feature, and localized moment feature, should be used for improving the matching between a slit and a reference pattern.

- *Reduction of computational complexity*: Beam search techniques (cost-based pruning and lexicon-based pruning) will be effective to reduce the computational complexity.

Acknowledgment: This work was supported in part by the Research Grant of The Okawa Foundation and the Research Grant (No.17700198) of The Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

[1] D. Doermann, J. Liang and H. Li: "Progress in Camera-Based Document Image Analysis," Proc. ICDAR, pp. 606–616, 2003.

[2] A. Zappala, A. Gee, M. Taylor: "Document mosaicing," Image and Vision Computing, vol. 17, no. 8, pp. 585–595, 1999.

[3] H. Sakoe, H. Fujii, K. Yoshida, and M. Watari: "A high-speed DP-matching algorithm based on frame synchronization, beam search and vector quantization," Systems and Computers in Japan, vol. 20, no. 11, pp. 33-45, 1989.

[4] R. Plamondon and S. N. Srihari: "On-Line and Off-Line Handwriting Recognition : A Comprehensive Survey," IEEE Trans. Pat. Anal. Mach. Intell., vol. 22, no. 1, pp. 63–84, Jan. 2000.

[5] T. Sato, S. Ikeda, M. Kanbara, A. Iketani, N. Nakajima, N. Yokoya, and K. Yamada: "High-resolution video mosaicing for documents and photos by estimating camera motion," Proc. SPIE Electronic Imaging, vol. 5299, 2004.

[6] H. Li and D. Doermann: "Text Enhancement in Digital Video Using Multiple Frame Integration," Proc. ACM Multimedia, pp. 19–22, 1999.

[7] J. Kosai, K. Kato, and K. Yamamoto: "Recognition of low resolution character by a moving camera," Proc. 5th Int. Conf. Quality Control by Arti cial Vision (QACV'99), pp. 203-208, 1999.

[8] S. Uchida and H. Sakoe : "Eigen-deformations for elastic matching based handwritten character recognition," Pattern Recognition, vol. 36, no. 9, pp. 2031–2040, 2003.

Super-resolution Text using the Teager Filter

Céline Mancas-Thillou
Faculté Polytechnique de Mons
Avenue Copernic, 1
7000, Mons, Belgium
celine.thillou@tcts.fpms.ac.be

Majid Mirmehdi
University of Bristol
Department of Computer Science
Bristol BS8 1UB, UK
M.Mirmehdi@cs.bris.ac.uk

Abstract

We propose a super-resolution technique specifically aimed at enhancing low-resolution text images from handheld devices. The Teager filter, a quadratic unsharp masking filter, is used to highlight high frequencies which are then combined with the warped and interpolated image sequence following motion estimation using Taylor series decomposition. Comparative performance evaluation is presented in the form of OCR results of the super-resolution output.

1. Introduction

Recent advances in hardware and sensor technologies have led to handheld camera-enabled devices such as PDAs or smartphones which in turn have become extremely popular. This has given rise to new potential applications many of which remain impractical due to some of the relative drawbacks when using these devices, e.g. low-resolution, sensor noise, uneven illumination, and complexity of natural scene images. The drawback dealt with in this paper is the problem of low resolution images; we present our experimental approach to reconstruct a higher resolution image by way of a super-resolution (SR) technique which responds better to standard off-the-shelf OCR software.

SR methods can be found in a multifarious range of imaging applications, such as medical imaging, astronomical and space imaging, surveillance imaging and many more. Park et al. provide a comprehensive review of general SR image reconstruction in [1]. For text and document analysis and recognition, super-resolution methods are becoming more important and necessary as the application areas extend to lower resolution camera enabled devices. A typical application scenario may be the use of a mobile phone camera to capture one or more lines of text on an advertising poster while on a metro train. The result will be a shaky low-resolution image sequence. This could possibly be sent to a server for transformation into text or be done on the fly on the phone if (one day) enabled. Other applications which may require SR text preprocessing include a tourist translation assistant or text-to-speech transformation for the visually impaired.

Multi-Input Single-Output (MISO) super-resolution techniques recover high frequencies from multiple low-resolution (LR) frames into a SR image. The motion present between LR frames of the same scene enables the recovery of high frequencies after registration and warping. The former step can be achieved by employing any one of a variety of *motion estimation* techniques, depending on the model required for the complexity of motion involved. The latter step is performed by interpolating LR registered frames into a single higher resolution one using techniques generally referred to as *reconstruction* methods. Finally, due to aliasing effects, errors during the motion estimation step, and/or initial blur present in the original LR frames, an additional *deblurring and denoising* step can be applied to smooth the SR image.

In this paper, the extraction of high frequencies is made easier by using an unsharp masking filter inside the SR process. In order to be more robust against impulsive noise, the quadratic 2D Teager filter [2] is used instead of linear unsharp filters. Quadratic non-linear filters have proven their efficiency to enhance character edges properly, as detailed in [3].

Initially, we apply Taylor series based motion estimation using a simple affine model followed by an outlier removal stage. The frames are then warped and interpolated to obtain an initial SR image. Then the Teager filter is applied to the LR image sequence and the frames are warped using the motion parameters obtained from the original unfiltered sequence. After also interpolating these frames, the result is fused with the initial SR image to obtain a final SR result. For data, short video sequences of text documents (e.g. advertisements, newspapers, book covers) were captured with a camera-enabled PDA at 320×240 resolution. The scene motion was induced by simply holding the de-

vice over the document (with a quivering hand) for a short period of around $5-7$ seconds at approximatively 5 fps, resulting in 25 to 35 frames per sequence. The scenes were mainly composed of nearly uniform backgrounds and the images were processed in grayscale. No a priori knowledge of parameters such as camera sensor noise, PSF etc was used. Hence, the proposed method is independent of camera models.

Next, we review previous work in super-resolution applied to text images. Section 3 outlines our SR approach combining motion corrected frames with Teager filtered frames. Comparative results are presented in Section 4. The paper is concluded in Section 5 with a discussion of the merits and shortcomings of the proposed method.

2. Previous Work on SR Text

Several past works on general SR have illustrated their results on images containing text as well as other scenes e.g. [4, 5], but very few have addressed SR specifically aimed at text analysis, and even fewer have carried out proper assessment and evaluation of the results using OCR recognition. Here, we consider the text-related works only for brevity. A more comprehensive review of SR text can be found in [6].

Applying text "enhancement" to overlaid texts in TV video sequences, Li and Doermann [7] assumed a pure translational model between frames. This was particularly suitable for their application since overlaid text, such as programme credits, usually have rigid and linear horizontal or vertical motion only. The motion estimation was performed using spatial-domain pairwise correlation minimizing sum of square differences between interpolated text blocks. In a driver assistance system [8], Fletcher and Zelinski used feature-based registration for the recognition of road signs, e.g speed limits. First, signs were detected as the dominant circles in a sequence using the Fast Symmetry Transform. Then, the circles were the features to register and normalized cross-correlation was performed on them to compute the translational motion vectors. A running integration of multiple image inputs was used to achieve super-resolved images for better recognition. Donaldson and Myers [9] also assumed a pure translational model and motion estimation was carried out by pairwise correlation. Then, a Bayesian framework with a MAP estimator was used for reconstruction of SR text which allowed the inclusion of a priori information to constrain errors: a bimodality prior assuming that text is bimodal and a Gibbs prior with a Huber gradient penalty function assuming that text images are locally smooth. Chiang and Boult [10] considered the same motion estimation algorithm as in [7] and applied local blur estimation for the reconstruction phase. To build illumination-invariancy, edge and blur models of all their frames were warped followed by a median fusion of the frames to a reference image with standard illumination. Then classical interpolation was applied to increase the resolution. Only visually enhanced results were shown.

A pure translational model is a common assumption in most papers due to its simplicity and ease of implementation. Nevertheless, with real-scene data, it can lead to misregistration and require a more elaborate reconstruction step. Capel and Zisserman [11] used a projective transform motion model for SR text specifically for image sequences in which the point-to-point image transformation was of enough complexity to demand such consideration. Two methods, a MAP estimator based on a Huber (edge penalty function) prior and an estimator regularized by using the Total Variation norm were proposed and compared for SR text. Again, only visually enhanced results were reported.

Interestingly, no affine models have been tried on text image sequences. For applications of a camera-enabled device, held at a sensible distance from a text scene, we suggest that a simple 3-parameter affine model of motion is a good representation and compromise between accuracy and overall complexity of a solution.

3. Proposed Method

We propose a method in which motion estimation is applied on the LR frames using Taylor decomposition, followed by a simple RANSAC-based step to discard obvious outlier frames. The frames are then warped and bilinearly interpolated to obtain a preliminary SR result. The original frames (except the outliers) are then put through the Teager filter to generate a high pass set of frames which are also warped and interpolated for a secondary SR result. The two resulting SR images are then fused and median denoising is applied to smooth artefacts due to the reconstruction process to obtain the final SR image. This process is illustrated in Figure 1 and detailed next.

3.1 Super-resolution Text

To avoid a propagation of errors it is important to estimate the motion parameters as accurately as possible. We apply Taylor series decomposition, as suggested in [12] who applied it to register frames to correct atmospheric blur in images obtained by satellite. This approach fits very well to text capture with a quivering hand since a shaking hand can produce slight random motions and the approximation computed by Taylor series decomposition can be suitable due to the small motion amplitudes involved. Initially a pure translational model was used but this led to too many (small) misregistration errors to adequately and reasonably correct afterwards. We noticed a significant improvement when stepping up to a 3-parameter affine motion model

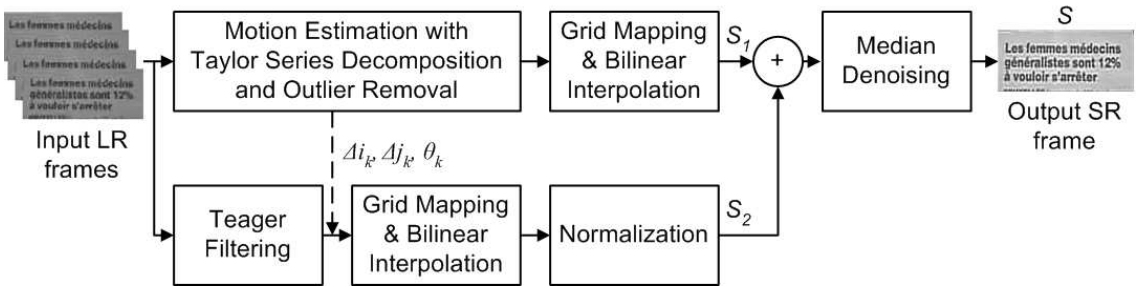


Figure 1. Schema of the proposed SR method.

(Δi_k, Δj_k, for horizontal and vertical translation, and θ_k for rotation). Given K frames with $k = 1, .., K$, the motion between a frame y_k and the first frame y_1 can be written as:

$$y_k(i,j) = \quad (1)$$
$$y_1(i\cos\theta_k - j\sin\theta_k + \Delta i_k, j\cos\theta_k + i\sin\theta_k + \Delta j_k)$$

Then, if the $\sin$ and $\cos$ terms are replaced by their 1st-order Taylor series expansion:

$$y_k(i,j) \approx y_1(i+\Delta i_k - j\theta_k - i\frac{\theta_k^2}{2}, j+\Delta j_k + i\theta_k - j\frac{\theta_k^2}{2}) \quad (2)$$

This can be approximated using its own 1st-order Taylor series expansion:

$$y_k(i,j) \approx y_1(i,j) + (\Delta i_k - j\theta_k - i\frac{\theta_k^2}{2})\frac{\partial y_1}{\partial i} \quad (3)$$
$$+(\Delta j_k + i\theta_k - j\frac{\theta_k^2}{2})\frac{\partial y_1}{\partial j}$$

The optimum motion parameter set $\mathbf{m}_k = (\Delta i_k, \Delta j_k, \theta_k)$ can then be estimated by solving this least-squares problem:

$$\mathbf{m}_k = \min_{\Delta i_k, \Delta j_k, \theta_k} \sum_{i,j} [y_1(i,j) + (\Delta i_k - j\theta_k - i\frac{\theta_k^2}{2})\frac{\partial y_1}{\partial i} \quad (4)$$
$$+(\Delta j_k + i\theta_k - j\frac{\theta_k^2}{2})\frac{\partial y_1}{\partial j} - y_k(i,j)]^2$$

After this motion estimation stage, we perform outlier frame removal (see Section 3.2 for details), followed by warping and bilinear interpolation by a factor of 4 on the remaining N low-resolution images to obtain the first stage initial SR image S_1 as:

$$S_1 = \mathcal{I}(\sum_{k=1}^{N} W_{\mathbf{m}_k} y_k) \quad (5)$$

where $W_{\mathbf{m}_k}$ is the warp matrix for each LR frame y_k using motion estimation parameter set $\mathbf{m}_k$, and $\mathcal{I}$ is the interpolation function.

To recover high frequencies easily and efficiently for MISO super-resolution, we need to enhance them in the LR images with appropriate filters. Relevant high frequencies such as character/background borders should be highlighted but impulsive perturbations must not. Non-linear quadratic unsharp masking filters using local properties of the image can satisfy these requirements. For example, the 2D Teager filter which is a class of quadratic Volterra filters [2] can be used to perform mean-weighted high pass filtering with relatively few operations. Its response is stronger in regions of high average intensity than in regions of low average intensity satisfying Weber's law [13]. Hence, using the local statistics of the image, the readability by a human user or the recognition by an OCR software is improved. Comparison to a linear unsharp masking filter such as the most classical one based on the negative Laplacian high-pass filter is detailed in Section 4. Using the same N corresponding original frames, we perform Teager filtering to obtain $y_k^\tau, (k = 1, ..., N)$ as the set of filtered images (see the lower row in Figure 1). For example, for any image y:

$$y^\tau(i,j) = \quad (6)$$
$$3y^2(i,j) - \frac{1}{2}y(i+1,j+1)y(i-1,j-1)$$
$$-\frac{1}{2}y(i+1,j-1)y(i-1,j+1)$$
$$-y(i+1,j)y(i-1,j) - y(i,j+1)y(i,j-1)$$

This filter enables us to highlight character edges and suppress noise. The shape of the Teager filter is shown in Figure 2 and an example image with its Teager filtered output in Figure 3. Next, we warp the frames using the same corresponding motion parameters $\mathbf{m}_k$ to reconstruct a secondary SR image S_τ:

$$S_\tau = \mathcal{I}(\sum_{k=1}^{N} W_{\mathbf{m}_k} y_k^\tau) \quad (7)$$

This is then normalized to provide:

$$S_2(i,j) = \frac{S_\tau(i,j) - min(S_\tau(i,j))}{max(S_\tau(i,j)) - min(S_\tau(i,j))} \quad (8)$$

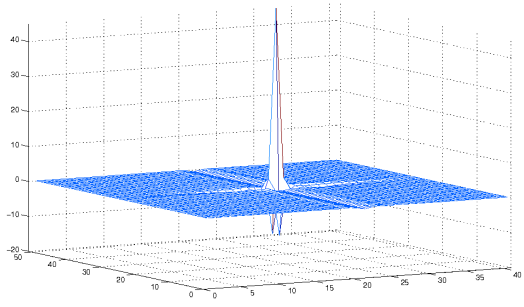


Figure 2. Visualization of the 2D Teager filter

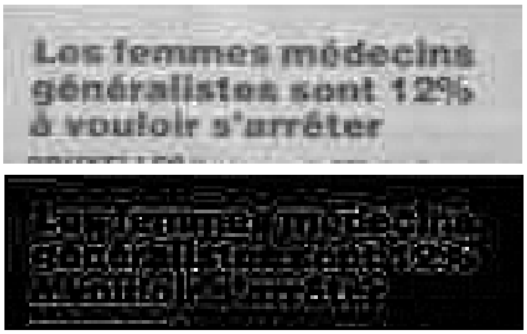


Figure 3. Top: initial LR image, bottom: Teager-filtered output.

The final SR output image S of our proposed method can then be expressed as:

$$S = med(S_1 + S_2) \tag{9}$$

where med is median denoising applied after fusion of the motion corrected representation with the motion corrected high frequency content.

3.2 A Closer Look

We now consider several important aspects of the method.

During motion estimation between frames errors occur if a text line is incorrectly registered with a neighboring one. A frame corresponding to incorrectly estimated parameters in $\mathbf{m}_k$ should therefore be dropped from further analysis. In our experiments we found that Δi_k or θ_k rarely caused any errors, whereas misregistrations frequently occurred on the vertical translations Δj_k leading to results such as that shown in Figure 4. The left example in Figure 5 shows a plot of Δj_k points in which an outlier value can be rejected after linear regression. However, there may be consecutive sets of outlier frames, hence we detect outliers by fitting a

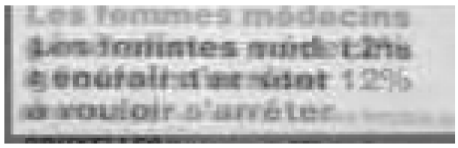


Figure 4. Fusion of two misregistered frames.

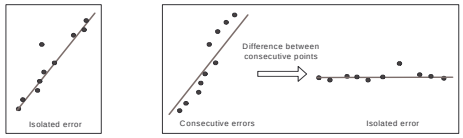


Figure 5. Left: an isolated Δj_k error, right: consecutive Δj_k errors result in wrong estimation, so Δj_k differences must be examined.

RANSAC-based least squares solution to the *differences* between vertical translations (illustrated in the right of Figure 5). Outlier frame rejection not only reduces the number of frames processed, but most importantly removes the need to apply regularization techniques during or after the reconstruction process. Note, this can easily be performed on all parameters in $\mathbf{m}_k$.

In Figure 6 we present a zoomed in view of a text document to emphasize the importance and effect of (a) Teager filtering and (b) the median denoising stages. The left image on the second row shows a pure interpolation of the original frame. The right image shows the interpolation result of all the frames in the sequence and hence is the result of $med(S_1)$ only. The left image in the last row is the result of $(S_1 + S_2)$ illustrating significant improvement when the Teager processing pipeline shown in Figure 1 is employed. Median denoising becomes necessary as the reconstruction result $(S_1 + S_2)$ alone is not smooth enough with errors arising from all the earlier stages of motion registration, warping, and interpolation. The resulting artefacts are objectionable to the human eye and would affect OCR. We applied a 3×3 neighborhood median for all our text images. The right image in the last row in Figure 6 shows the final result obtained from (9).

4. Experiments and Results

The impact of **unsharp masking filtering** can be further emphasized as follows. The top image in Figure 7 shows the results of a classical MISO approach (the same as just the top row of the schema in Figure 1, i.e. $med(S_1)$ only). In comparison, the bottom image shows the result of the proposed method displaying better sharpness and readability.

In Figure 8 we compare our method to the one presented

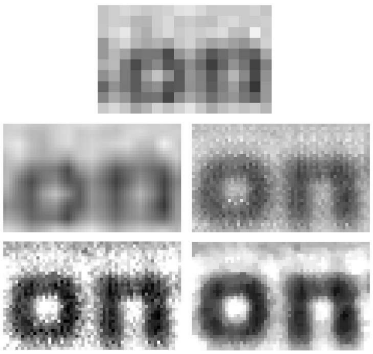


Figure 6. First row: original LR frame. Second row: bilinear interpolation applied on one LR frame, SR output without using Teager-filtered frames (S_1). Third row: proposed method without denoising ($S_1 + S_2$), full proposed method.

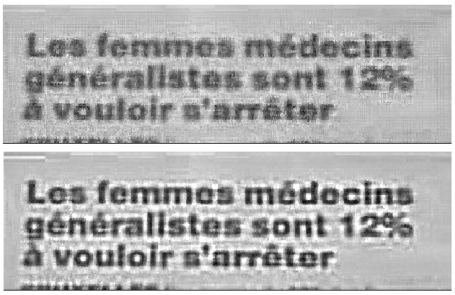


Figure 7. Top: classical approach ($med(S_1)$), bottom: the proposed method.

Figure 8. Top: SR image obtained with the algorithm in [7], bottom: our method.

Figure 9. A face example: classical approach on the left and our method on the right.

in Li and Doermann [7] in which a simple translational model was used for text enhancement. Bearing in mind that their method was developed for text primarily moving in vertical and horizontal directions, nevertheless this comparison shows that the use of an affine model is minimally necessary in our type of applications. The registration errors in the top image of Figure 8 make it very difficult for interpretation by OCR analysis.

As a matter of interest we also applied our method to several non-text examples. Figure 9 demonstrates the results of traditional SR (here the top steps in our schema in Figure 1, i.e. $med(S_1)$ only) on the left and the proposed method on the right for a face video sequence. In this example we did not apply the frame outlier removal step and used a larger 9×9 neighborhood for median denoising.

Figures 10 and 11 present more text images with and without the Teager stage to highlight the usefulness of this filter. In the zoomed examples in Figure 11, while OCR of all the SR images will recognize the characters in both methods, however note the difference in quality after Otsu binarization where the proposed method produces a much sharper and better defined set of characters with Teager filtering than without.

Finally, percentage recognition rates based on several natural scene text video sequences are shown in Table 1 for comparison of the classical approach in general super-resolution (C) to the proposed method using either a linear Laplacian-based unsharp masking filter (L) or the quadratic non-linear Teager filter (S). Our results demonstrate much better performance at 87.8% accuracy on average, computed on the number of correctly recognized characters, showing that the proposed method is clearly better equipped in handling noisier data.

5. Discussion and Future Work

A SR text application was presented using low-resolution camera-based video sequences with the motion induced while holding a camera-enabled PDA device. In

Table 1. Comparative OCR accuracy rates (%)

Test	C	L	S
1	48.1	78.8	78.8
2	75.2	94.3	92.9
3	65.2	56.5	78.3
4	77.7	84.4	86.0
5	95.1	100.0	100.0
6	66.6	83.3	91.6
7	75.0	79.4	86.4
8	79.3	79.3	79.3
9	72.7	81.8	90.9
10	72.5	88.8	93.8
Avg.	72.7	82.7	87.8

order to recover the high frequencies in the LR images and interpolate the data into a SR image, we enhanced the classical SR approach with the Teager filter. The final results show sharper characters with more contrast against their background. This is particularly important in increasing OCR efficiency. We obtained very good comparative OCR results on a small set of sequences.

An important drawback in SR text is the presence of thin characters. Motion estimation has to be very accurate in order not to lose them. The proposed method is not immune to this drawback.

The Teager filter is very good as a quadratic, unsharp masking filter. Other similar filters such as the Ramponi filter [3] may also be capable of achieving similar results.

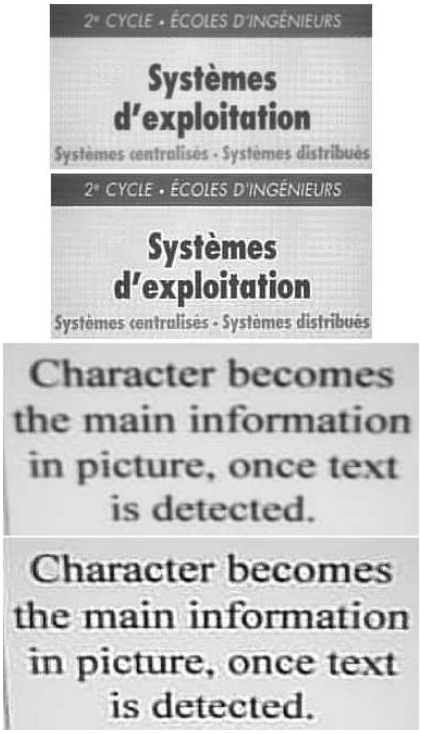


Figure 10. SR using the classical approach (top) and the proposed method (bottom).

Acknowledgements

The first author was partly funded by Ministère de la Region wallonne in Belgium and by a mobility grant from Ministère de la Communauté Française to work at the University of Bristol.

References

[1] S.C.Park, M.K.Park, G.K.Moon. Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, 20(3):21-36, 2003.

[2] S.Mitra, G.Sicuranza. *Nonlinear Image Processing*, Academic Press, 2000.

[3] G.Ramponi, P.Fontanot. Enhancing document images with a quadratic filter. *Signal Processing*, 33:23-34, 1993.

[4] P.Vandewalle, S.Süsstrunk, M.Vetterli. A frequency domain approach to registration of aliased images with application to super-resolution. To appear in *EURASIP Journal on Applied Signal Processing*, 2005.

[5] S.Farsiu, M.D.Robinson, M.Elad, P.Milanfar. Fast and robust multiframe super-resolution. *IEEE Trans. Image Processing*, 13(10):1327-1344, 2004.

[6] C.Mancas-Thillou, M.Mirmehdi. An introduction to super-resolution text. *To appear in Recent Advances in Digital Document Processing*, Springer-Verlag, 2005.

[7] H.Li, D.Doermann. Text enhancement in digital video using multiple frame integration. *Proc. of the ACM Int. Conf. on Multimedia*, 19-22, 1999.

[8] L.Fletcher, A.Zelinsky. Super-resolving signs for classification. *Proc. of Australasian Conf. on Robotics and Automation, Canberra, Australia*, 2004.

[9] K.Donaldson, G.K.Myers. Bayesian super-resolution of text in video with a text-specific bimodal prior. To appear in *Int. Journal of Document Analysis and Recognition*, 2005.

Figure 11. Two zoomed in SR results comparing the classical approach and the proposed method and their binarized images.

[10] M-C.Chiang, T.E.Boult. Local blur estimation and super-resolution. *Proc. of IEEE Computer Vision and Pattern Recognition*, 821-826, 1997.

[11] D.Capel, A.Zisserman. Super-resolution enhancement of text image sequences. *Proc. of Int. Conf. on Pattern Recognition*, 600-605, 2000.

[12] D.Keren, S.Peleg, R.Brada. Image sequence enhancement using sub-pixel displacements. *IEEE Proc. on Computer Vision and Pattern Recognition*, 742-746, 1988.

[13] A.K.Jain. *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

Camera Document Restoration for OCR

SHIJIAN LU, CHEW LIM TAN, SENIOR MEMBER, IEEE
School of Computing, National University of Singapore
Singapore 117574

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 skew, perspective, and geometric distortions coupled within the captured images make it hard to recognize the document text using the generic OCR systems. In this paper, we propose a document restoration technique, which is capable of removing the three types of distortions, and reconstructing the fronto-parallel view of the document text using a single document image captured through a digital camera. Different from the reported techniques, the proposed restoration technique is carried out based on the vertical stroke boundary and the top line and base line of text lines. Experimental results show the proposed technique is fast, accurate, and robust.

1. Introduction

As sensor resolution increases in recent years, high-speed non-contact text capture through a digital camera is becoming an alternative choice. Unfortunately, the document images captured through a digital camera are often coupled with the distortions including rotation-induced skew, perspective, and geometric distortions. These three types of distortions must be removed before the camera documents are fed to the generic OCR system.

As Figure 1(a) shows, the rotation-induced skew normally occurs as the image plane *R* of the digital camera is parallel to the document plane *D*. While the camera image plane *R* is not parallel to the document plane *D*, the perspective distortion as illustrated in Figure 1(b) is inevitably introduced. In addition, as most of scene documents such as the hand-held newspaper, the paper sheets pasted on cylindrical containers and even the pages bound within the thick books generally lie on a smoothly curved instead of planar surface, the camera documents are often coupled with the geometric distortion as illustrated in Figure 1(c) as well.

A large number of document restoration methods [1-6] have been reported in the literature. Traditionally, document distortion generally refers to the rotation-induced skew and the main problems of the reported skew detection methods lie with the restriction on the detectable skew angle range [1] and the heavy computation load [2]. In recent years, a few perspective restoration techniques have been reported, but most of the reported techniques rely heavily on the image features such as the high contrasted document boundary (HDB) [3] and the paragraph formatting (PF) information such as paragraph margins [4]. A few geometric restoration techniques have been reported as well in recent years. Most of reported methods [5-6] approach the restoration problem through the 3D reconstruction, but auxiliary hardware is normally required for 3D measurements.

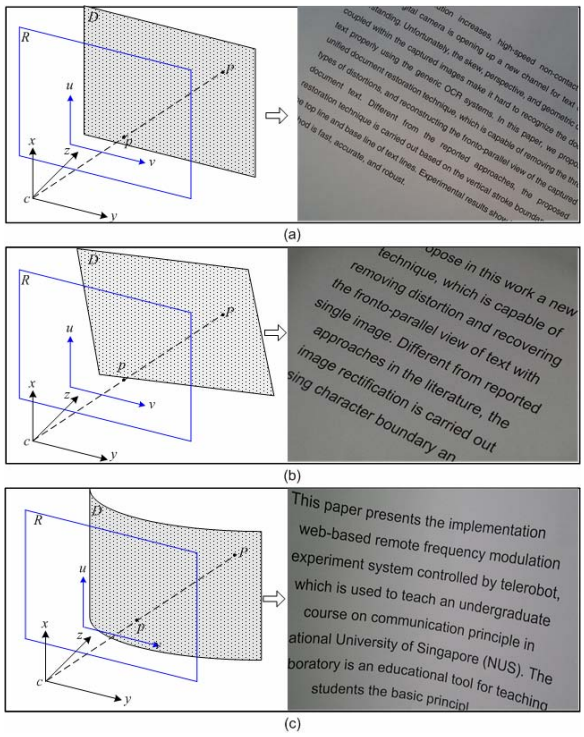


Figure 1: (a) camera document with skew; (b) camera document with perspective distortion; (c) camera document with geometric distortion

In [7], we propose to remove the perspective distortion through the detection of the vertical stroke boundary (VSB) and the top line and base line of text lines as labeled with (1) and (2) in Figure 2. VSBs are firstly identified based on three fuzzy sets that characterizes the size, linearity, and orientation of the extracted stroke boundaries. The top line and base line of text lines are then fitted using character tip points that are classified based on the structure of the typeset document text. For the sample document given in Figure 1(b), Figure 3 shows the identified VSB where text is printed in a light gray color to highlight its relative position to the identified VSBs.

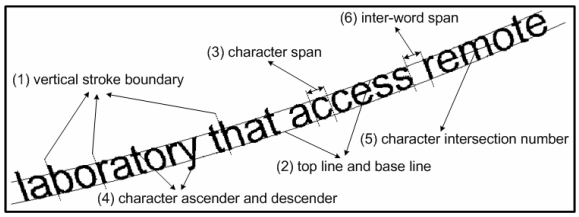


Figure 2: Text line definition

In this paper, we extend our work reported in [7] to restore the camera documents with three types of document distortions. The proposed technique has multiple advantages. Firstly, it is able to estimate the skew angle ranging from 0˚ to 360˚ and the estimation speed is totally independent of the skew angle. Secondly, it is able to rectify the perspectively distorted camera documents that have no HDB or PF features and may contain only one text line or even just a few words. Thirdly, it is able to restore the camera documents with geometric distortion with just a single document image captured through a digital camera. Lastly, the proposed technique needs no camera calibration and it requires only a camera image of document with good resolution.

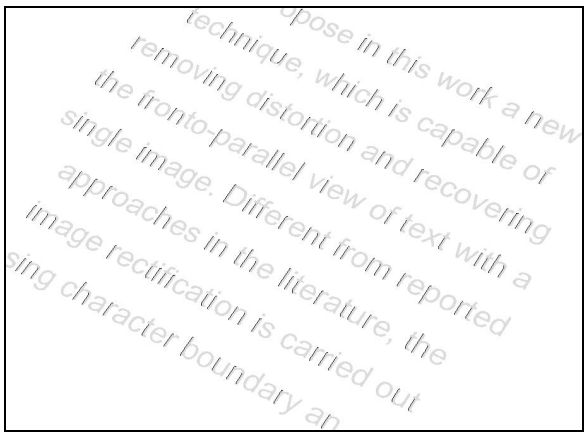


Figure 3: Identified VSB from document image given in Figure 1(b)

In our proposed approach, the skew angle can be simply estimated based on the orientation of the top line and base line. For the camera documents with perspective distortion, the restoration is carried out with the homography estimated based on the top line and base line of text lines and the identified VSB. For camera documents with geometric distortion, we propose to remove the distortion through the image segmentation, which partitions the camera documents into multiple small image patches where text can be approximated to lie on a planar surface. The global geometric distortion is then removed through the local rectification of the partitioned image patches one by one.

2. Proposed Restoration Technique

We present in this section the outline of the proposed document restoration technique. In particular, we will divide this section into a few subsections, which deal with the identification of document distortions and the restoration of camera documents with rotation-induced skew, perspective, and geometric distortions respectively.

2.1. Distortion Identification

For document images with rotation-induced skew, the restoration can be simply implemented through an image rotation operation. But for document images with geometric distortion, the restoration process is much more complex because it involves the VSB identification, the top line and base line fitting, and the image segmentation. Therefore, it is better to identify the distortion type first before the actual restoration operation. We propose to identify the distortion type based on the pattern of the classified character centroids, which normally fit well to a set of parallel straight lines, unparallel straight lines and smooth curves respectively for document images with the three types of distortions.

The document images with skew or perspective distortions can be firstly differentiated from the ones with geometric distortion based on the linear fitting error, which can be evaluated using the distance:

$$D = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m} Dist(C_j, L_i) \qquad (1)$$

where parameters n and m refer to the number of the fitted middle lines and the number of characters centroids within the *ith* classified character centroid category. Parameter L_i refers to the straight middle line fitted with the character centroids within the *ith* category. Function $Dist(C_j, L_i)$ calculates the distance be-

tween the *jth* character centroid C_j within the *ith* character centroid category and the *ith* fitted straight middle line L_i.

Based on the distance defined in Equation (1), the distance threshold can be determined as:

$$TD = k_d \cdot VSB_{avg} \tag{2}$$

Parameter VSB_{avg} is the average size of the identified VSBs, which normally indicates the size of the captured document text. Parameter k_d [0.1 0.5] is designed to adjust the distance threshold and we set it as 0.3 in the implemented system. Therefore, geometric distortion is detected if the distance D determined using Equation (1) is bigger than the distance threshold TD given in Equation (2). Otherwise, document images are determined to contain rotation-induced skew or perspective distortion.

Skew and perspective distortions can be further differentiated from each other based on relative orientation of the fitted middle lines. For document images with skew or perspective distortion, the fitted middle lines normally correspond to a set of parallel or unparallel straight lines respectively. The relative orientation of the fitted middle lines can thus be evaluated as:

$$RO = \frac{1}{n}\sum_{i=1}^{n}\left(\varphi_i - \frac{1}{n}\sum_{i=1}^{n}\varphi_i\right)^2 \tag{3}$$

where parameter n refers to the number of the fitted middle lines. Parameter φ_i refers to the orientation angle of the *ith* fitted middle line. For the document images with rotation-induced skew, the relative orientation RO determined in Equation (3) is quite close to zero. But for the document images with perspective distortion, the relative orientation RO is normally much bigger. Skew and perspective distortions can thus be differentiated based on the relative text line orientation RO given in Equation (3)

Table 1 Distortion identification performance

	NO of sample images	No of correctly identified images	Identification rate
Camera documents with skew distortion	30	31	96.67%
Camera documents with perspective distortion	30	29	96.67%
Camera documents with geometric distortion	30	30	100%

The proposed distortion identification technique is able to differentiate the three types of document distortions in most cases. We test the identification performance using 90 distorted camera images of documents as given in Table 1 where the distortion types of the 88 images are correctly identified. The identification error normally occurs while the two related distortions are quite close to each other. For example, perspective distortion may be falsely identified as skew distortion as the angle between the optical axis of digital camera and document plane is close to 90˚ and the captured text lines are roughly parallel to each other.

2.2. Document Restoration

This section presents the restoration of the camera documents with the three types of distortions including rotation-induced skew, perspective and geometric distortions respectively.

2.2.1. Skew Detection and Correction

For the camera documents with rotation-induced skew, the top line and base line of text lines are actually a set of approximately parallel straight lines. The skew angle can thus be simply estimated based on the orientation of the top line and base line. The skew angle can be determined as:

$$\tan(\beta) = \frac{1}{n}\sum_{i=1}^{n} slp_i \tag{4}$$

where parameter n is the number of fitted top line and base line and slp_i refers to the slope of the *ith* top line or base line.

Based on the skew angle determined using the Equation (4), the camera documents with skew distortion can be restored through a simple image rotation operation. For the skewed camera document given in Figure 1(a), Figure 4 shows the restored document image.

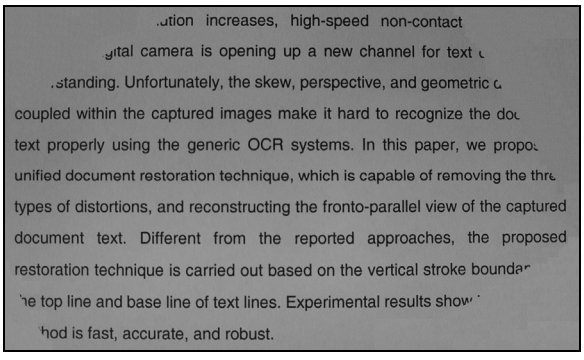


Figure 4: Skew restoration result

2.2.2 Perspective distortion detection and correction

For camera documents with perspective distortion, the top line and base line generally correspond to a set of unparallel straight lines. In this paper, we propose to remove the perspective distortion through the exploitation of the quadrilateral correspondences where the source quadrilaterals are constructed using the identified VSB and the top line and base line of text lines. For each determined source quadrilateral, a target quadrilateral is constructed based on the number of the characters enclosed within the source quadrilateral and the specific character width height ratios. With the constructed source and target quadrilateral correspondences, optimal rectification homography is estimated and perspective distortion is finally removed.

For each straight line fitted based on an identified VSB, there will be multiple intersections between it and the top line and base line pairs. The identified VSB must be classified to the text line from which they are extracted to construct the desired source quadrilaterals and this can be achieved based on the distances between the centroids of the identified VSB and the fitted top line and base line pairs. Thus, source quadrilaterals can be constructed using the identified VSB and the related top line and base line pairs.

As the camera capturing process impairs the geometric relation between the straight lines, we propose to construct the target quadrilateral based on the number of the characters enclosed within the source quadrilaterals and the approximation that the width height ratio of characters is 1:1. It should be clarified that the 1:1 width height ratio is only an average approximation, as the width height ratios of different characters such as "m" and "i" may differentiate quite a lot. To make the approximation more close to the fact, the constructed source quadrilaterals must be wide enough to enclose more characters. In our proposed technique, the distance threshold between two VSB is defined based on the average length of the captured text lines:

$$L = k_r \cdot \frac{1}{n}\sum_{i=1}^{n} leg_i \tag{5}$$

where n is the number of detected text lines. Parameter k_r is used to adjust distance threshold and we take it as 0.4 in our system. Symbol leg_i represent the length of *ith* text line, which is calculated as the distance between the leftmost and rightmost pixels of characters that belong to the same text line.

Characters within the constructed source quadrilaterals can thus be determined based on relative position between the character centroids and the four source quadrilateral edges. The inter-word blank can be detected as well based on the distance between the centroids of the adjacent characters and it takes the width same to a character. With the approximated character height width ratio, the relation between the width and height of target quadrilaterals can be restored as

$$l_q = n \cdot h_q \tag{6}$$

where parameters l_q and h_q are the length and height of the target quadrilaterals. Parameter n is the number of character enclosed within the source quadrilateral, including the detected inter-word blanks. The height of the target quadrilateral h_q can be commonly determined as the average size of the identified VSB.

With multiple pairs of source and target quadrilaterals, multiple rectification homographies can be determined using the four point algorithm [8]. The homography between the distorted and restored document images can be estimated as:

$$H = A^{-1} \cdot R \tag{7}$$

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

$$H = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix}, \quad A = \begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1' \cdot x_1 & x_1' \cdot y_1 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & y_1' \cdot x_1 & y_1' \cdot y_1 \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2' \cdot x_2 & x_2' \cdot y_2 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & y_2' \cdot x_2 & y_2' \cdot y_2 \\ -x_3 & -y_3 & -1 & 0 & 0 & 0 & x_3' \cdot x_3 & x_3' \cdot y_3 \\ 0 & 0 & 0 & -x_3 & -y_3 & -1 & y_3' \cdot x_3 & y_3' \cdot y_3 \\ -x_4 & -y_4 & -1 & 0 & 0 & 0 & x_4' \cdot x_4 & x_4' \cdot y_4 \\ 0 & 0 & 0 & -x_4 & -y_4 & -1 & y_4' \cdot x_4 & y_4' \cdot y_4 \end{bmatrix}, \quad R = \begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \\ x_4' \\ y_4' \end{bmatrix} \tag{8}$$

where the 3×3 homography matrix is expressed in vector form and h_{33} is equal to 1 under homogeneous frame. Four point correspondences $\langle (x_i, y_i), (x_i', y_i') \rangle$, i = 1, …, 4, are taken as the four vertices of the constructed source and target quadrilateral pairs.

The vertex position of the constructed source quadrilateral normally contains errors. As a small error in source quadrilateral vertices may introduce a big error to the restored document images, a criterion must be set to choose the homography that optimize the restoration performance. Based on the facts that the top line and base line should be restored to multiple horizontal lines and the identified VSB should lie on multiple vertical lines within the restored document image, we define the objection function as:

$$J = \frac{1}{m}\sum_{i=1}^{m} abs\left(\frac{S_{l_i}}{S_{avg}}\right) + \frac{1}{n}\sum_{j=1}^{n} abs\left(\frac{ptx_j - pbx_j}{Dist_{avg}}\right) \tag{9}$$

where m is the number of detected text lines and n is the number of the identified VSB. S_{li} is the orientation of *i-th* restored text line and S_{avg} is the orientation average. ptx_j and pbx_j represent two horizontal coordinates of vertices of *j-th* restored VSB and the component $abs((ptx_j - pbx_j)/ Dist_{avg})$ is the normalized distance

$abs((ptx_j - pbx_j)/ Dist_{avg})$ is the normalized distance in horizontal direction between the vertices of that vertical stroke boundary. The first part on the right side of Equation (9) represents the sum of normalized orientation of the restored text lines, which should be zero ideally, and the second part refers to the sum of normalized vertex distance of the restored VSB in horizontal direction, which ideally should be zero as well. The optimal homography can accordingly be determined as the one that minimizes the objection function defined in Equation (9).

The camera document with perspective distortion as given in 1(b) can be finally restored based on the estimated optimal homography. Figure 5 shows the restored document image.

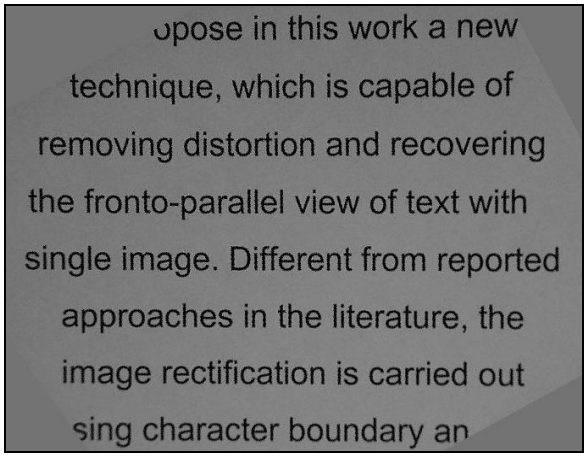


Figure 5 Perspective restoration result

2.2.3. Geometric distortion detection and correction

Based on the top line and base line pairs and the identified VSB, we propose to remove the geometric distortion through the image segmentation, which partitions the distorted camera documents into multiple small image patches where text can be approximated to lie on a planar surface. For each partitioned image patch, a target rectangle is then constructed based on the characters within that partitioned image patch and the specific character width height ratios. Lastly, the global geometric distortion is corrected through the local rectification of the partitioned image patches one by one.

Before camera document segmentation, the identified VSB must be processed further to facilitate the later restoration and make sure that the partitioned image patches enclose all captured text. Firstly, some VSB must be deleted if they are too close to their left adjacent neighbor. VSB deletion operation is designed to control the size of partitioned image patches, but the identified VSB cannot be deleted arbitrarily. In our proposed technique, we delete the VSB based on its distance to the left adjacent VSB and the distance threshold is determined as:

$$D_{thre} = k_d \cdot VBS_{avg} \tag{10}$$

where parameter VBS_{avg} represents the average size of the identified VSB. Parameter k_d is designed to adjust the distance threshold and it is determined as a number between 3 and 5 so that each partitioned image patch enclose 3-5 characters.

In addition, for the text lines that have no VSB identified at their left or right end, a VSB must be constructed there so that the partitioned image patches are able to enclose all characters that belong to the studied text line. The orientation of the VSB at the text line end positions can be estimated through linear interpolation:

$$slp = slp'' + \frac{(x - x'') \cdot (slp' - slp'')}{(x' - x'')} \tag{11}$$

where x is x coordinates of the leftmost or rightmost text pixel and x', x'' are x coordinates of centroids of two VSB that are nearest to the related leftmost or rightmost text pixel. Parameters slp' and slp'' are slopes of the straight lines fitted based on the two nearest VSB neighbors.

Accordingly, the VSB at the leftmost or rightmost end can be estimated as a straight line that passes through the leftmost or rightmost character pixel with orientation same to the one estimated in Equation (11). For the distorted word given in Figure 6(a), Figure 6(b) shows the top line and base line and the identified VSB. Figure 6(c) gives the processed VSB after the deletion and addition operations where the second VSB from the left is deleted and the VSB at the rightmost end of text line is estimated. For each processed VSB, a straight line can thus be fitted using the least square method. With the top line and base line of text lines and the straight lines fitted based on the processed VSB, distorted text as given in Figure 6(a) is finally segmented into three small patches as given in Figure 6(d).

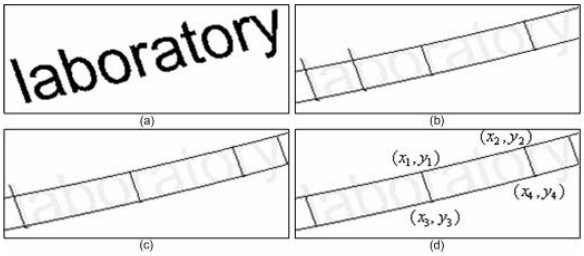


Figure 6: (a) Original sample word; (b) identified VSB and the top and base line; (c) VSB after the deletion and addition operation; (d) three segmented document patches

For each partitioned image patch, a target rectangle correspondence must be constructed within the target image to rectify that partitioned image patch. The height of target rectangles, which is same to the height of rectified characters with no ascender and descender, can be commonly determined as the average size of identified VSB. We thus propose to restore the width of target rectangles based on the characters enclosed within the partitioned image patches and specific character width height ratios.

While restoring the camera documents with only perspective distortion, we approximated the character width height ratio as 1:1. The approximated ratio works well because the width of the source quadrilateral determined using Equation (5) is large enough to contain a large number of characters. But for camera documents with geometric distortion, the partitioned image patches enclose only 3-5 characters and so the 1:1 approximation cannot be used here. We thus propose a rough character classification process to classify characters into different categories with different width height ratios.

We propose to classify characters into six categories with different character width height ratios. The classification is carried out based on multiple image features extracted from character strokes including character span, character ascender and descender, character intersection numbers, and inter-word spans as labeled with (3), (4), (5), and (6) in Figure 2. Character span is defined as the distance between two parallel straight lines tangent to the left and right sides of the studied character with the orientation same to that of the straight line fitted based on the nearest VSB. Inter-word span can be determined in the similar way as character span. The intersection numbers are equal to the number of intersection between character strokes and the straight lines that pass through the character centroid with orientation orthogonal to that of the straight line fitted based on the nearest VSB. Character ascender and descender can be determined based on the distance between the highest and lowest character pixel and the top line and base line.

With the determined text line features, the character classification algorithm is as follows:

Inputs: Binarized document image *BDI*; Calculated character spans *CSpan*; Ascender & descender information *ADInfo*; intersection numbers *Inter*

Procedure: *CC* (*BDI*, *CSpan, ADInfo, Inter*)

1) Initialize $i = 1$
2) Calculate the average character span *CSpan_avg* based on *CSpan.*
3) Repeat:
4) If *Inter(i)* ≥ 3 and *ADInfor(i)* = 1(with ascender), character is classified as 'M' or 'W'.
5) Else if *Inter(i)* ≥ 3 and *ADInfor(i)* = 0(no ascender), character is classified as 'm' or 'w'.
6) Else if *ADInfor(i)* = 1(with ascender) and *CSpan(i)* > $k_u \cdot$*CSpan_avg*, character is classified as A-H, J-L, N-V, or X-Z.
7) Else if *CSpan(i)* > $k_l \cdot$*CSpan_avg* and *CSpan(i)* < $k_u \cdot$*CSpan_avg* , character is classified as, a-e, g-h, k, n-q, s, or u-v, or x-z
8) Else if *CSpan(i)* < $k_s \cdot$*CSpan_avg*, character is classified as 'i', 'l', 'I' or 'j'.
9) Else, character is classified as t, f, or r.
10) $i = i + 1$
11) Until i is equal to the number of characters within *BDI*

Table 5.1 shows the proposed six categories and the related character width-height ratios.

Table 1: 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
Inter-word span	1:1
a-e, g-h, k, n-q, s, u-v, x-z	0.8:1
t, f, r	0.5:1
i, j, l, I,	0.2:1

The average character span *CSpan_avg* in Step 2) is firstly determined before the classification. Parameter k_u, k_l, and k_s in Steps 6), 7) and 8) are three key parameters for character classification. In our implemented system, the three parameters are determined as 1.2, 0.7, and 0.3 respectively based on the relative width of characters in different categories.

We evaluate the proposed character categorization technique using the same 90 sample images as used for distortion identification. Experiment results show that the correct classification rate can reach over 96%. The small classification error will not affect the recognition performance of the restored document images seriously because each partitioned image patch normally contain 3-5 characters.

The width of target rectangles can thus be determined as:

$$T_w = \sum_{i=1}^{n} R_i \cdot VBS_{avg} \tag{12}$$

where VBS_{avg} represent the average size of identified VSB and parameter n represents the number of charac-

ters and inter-word blanks enclosed within the partitioned image patches. Parameter R_i refers to width height ratios of characters and inter-word blanks within the partitioned image patches as given in Table 1.

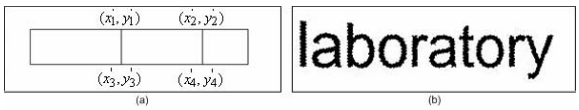


Figure 7 (a) constructed target quadrilateral; (b) restored document image

For the segmented image patches given in Figure 6(d), Figure 7(a) shows the constructed target rectangles. For each quadrilateral correspondence, a homography can be determined using Equation (8). Document text with geometric distortion given in Figure 6(a) can thus be restored through the rectification of three partitioned image patches. Figure 7(b) gives the restoration result.

Figure 8 illustrates the geometric restoration process where Figure 8(a) gives a camera document with geometric distortion. Based on the top line and base line and the identified VSB, the camera document is segmented into multiple image patches as shown in Figure 8(b). With the partitioned image patches, target rectangles are then constructed based on the enclosed characters and the specific character width height ratios as given in Figure 8(c). Finally, the camera document is restored based on the partitioned image patches and the constructed target rectangles. Figure 8(d) shows the restored image.

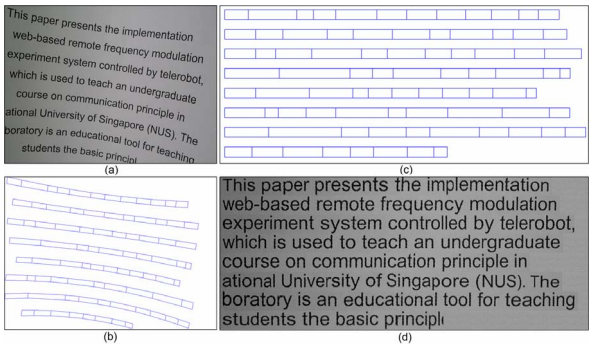


Figure 8(a) camera document with geometric distortion; (b) segmented image patches; (c) constructed target patches; (d) restored document image

3. Experimental Results

We implement the proposed technique based on the methods described above. The system is implemented in C++ and runs on a personal computer equipped with Window XP and Pentium 4 CPU. We evaluate the proposed technique with an image database that contains 90 camera documents with each 30 coupling with the skew, perspective and geometric distortions respectively. Experimental results show the proposed technique is able to restore the camera documents with three types of distortions efficiently.

We evaluate the proposed restoration technique based on the recognition rates of the document image after our proposed restoration operation. The OCR performance is tested using the software Omnipage Pro 14.0 [9]. The average recognition rates of 90 camera documents before the restoration operation are less than 10%. This result can be expected since the generic OCR systems can not deal with the perspective and geometric distortions well. At the same time, most of OCR systems perform poorly while the skew angle is bigger than 20˚. Figure 9 gives the experimental results where the recognition rates of three groups of images are illustrated with three types of curves labeled with pentagram, star, and diamond symbols respectively.

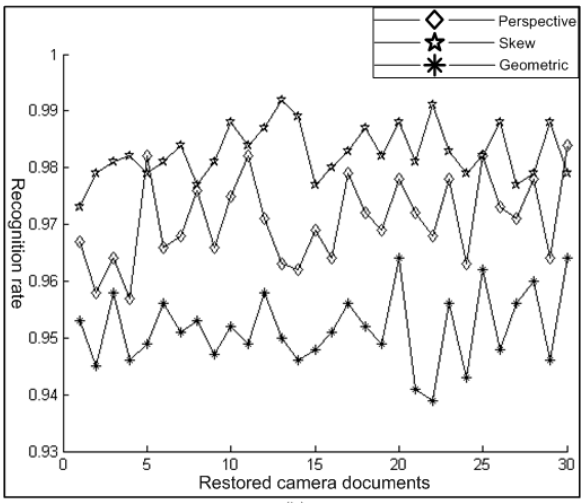


Figure 9: Recognition rate after the proposed restoration operation

As Figure 9 shows, the average recognition rate of the restored document images can reach over 95%. Such recognition rate shows that the proposed technique has the potential to be applied for the recognition of the distorted camera documents in practice. As the figure shows, the recognition rate of the camera document with skew and perspective distortion is normally a bit higher than that of the ones with geometric distortion. Such difference can be explained by the bigger errors introduced during the image segmentation process, which is required for the restoration of the geometric distortion.

Though the proposed technique is able to handle most of camera documents, some problems still exist.

Firstly, the proposed technique depends heavily on the resolution of the captured camera documents, as the required VSB component may not be identified properly from the camera documents with poor resolution. With the same reason, the proposed technique cannot handle the camera documents with arbitrary geometric distortion such as the crumpled paper sheets and the ones printed in handwritten text. Some new approaches will be investigated to solve these problems next.

4. Conclusion

In this paper, a unified document restoration technique is proposed to correct the document images with skew, perspective, and geometric distortions captured through a digital camera. The restoration of the camera documents is implemented through the exploitation of the vertical stroke boundary and the top line and base line of text lines. Different from the reported document restoration techniques that depend heavily on HDB, PF, and the auxiliary hardware equipments, the proposed technique needs only a single document image captured through a digital camera. Experimental results show that the proposed document restoration technique is fast, accurate, and robust.

References

[1] T. Akiyama and N. Hagita, "Automated Entry System for Printed Documents," *Pattern Recognition*, vol. 23, no.11, pp. 1141-1154, 1990

[2] D. S. Le, G. R. Thoma and H. Wechsler, "Automated Page Orientation and Skew angle Detection for Binary Document Image," *Pattern Recognition,* vol. 27, no. 10, pp. 1325-1344, 1994.

[3] P. Clark, M. Mirmehdi, "Recognizing text in real scenes," *International Journal of Document Analysis and Recognition*, vol. 4 no. 4, pp. 243–257, 2002

[4] C. R. Dance, "Perspective estimation for document images," *SPIE Conference on Document Recognition and Retrieval IX,* San Jose, USA, pp. 244–254, 2002.

[5] M. S. Brown and W. B. Seales, "Document restoration using 3D shape: a general deskewing algorithm for arbitrarily warped documents," *International Conference on Computer Vision,* Vancouver, Canada, pp. 117-124, 2001.

[6] A. Doncescu, A. Bouju and V. Quillet, "Former books digital processing: image warping," *Proceedings of Workshop on Document Image Analysis*, San Juan, Puerto Rico, pp. 5-9, 1997.

[7] S. J. Lu, B. M. Chen and C. C. Ko, "Perspective rectification of document images using fuzzy set and morphological operations," *Image and Vision Computing,* vol. 23, pp. 541-553, 2005.

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

[9] http://www.scansoft.com/omnipage/

Unwarping Images of Curved Documents Using Global Shape Optimization

Jian Liang, Daniel DeMenthon, David Doermann
Language And Media Processing Laboratory
University of Maryland, College Park, MD, 20770
{lj,daniel,doermann}@cfar.umd.edu

Abstract

The unwarping of curved document images is a crucial problem for camera-based document analysis since most of current OCR techniques can not handle distortion due to perspective and warping. In previous work we have shown how to recover the page shape from a single image using an iterative procedure without camera calibration, and using the shape information to restore a frontal view of a flat document. In this paper we report our recent progress using a global optimization method to do shape estimation. Experimental results show a clear improvement over our previous method.

1 Introduction

Digital cameras have become more and more popular not only among consumers but also business and technical professionals. For the OCR community, they provide a potential alternative to scanners as document imaging devices. Current OCR techniques are, however, designed with digital scans of flat documents in mind, and cannot handle general camera-captured documents due to both perspective and warping.

One way of removing the added 3D distortion is to use special 3D scanning equipments such as structured light. A mesh can be built to represent the 3D surface and directly flattened [1] or transformed to a developable mesh [8]. Alternatively, the shape can be estimated from the image. The problem of removing only the perspective from images of planar documents is addressed in [3, 8, 4]. For warped documents, there are parametric approaches [2, 5, 12] that estimate the 3D shape and non-parametric ones [10, 11] that bypass shape estimation. Among them, [11, 12] are designed only for scans of bound books; [2, 10] require a straight frontal view of page with cylinder shape; [5] is proposed for general images, but needs camera calibration and a prior knowledge of a closed contour (e.g., page boundaries) on the page which may be difficult in practice. Overall, current methods have various restrictions that keep them from being applied to general images.

Our goal is to handle general warped documents with fewer restrictions. Our method falls in the parametric category. It is based on two key observations: 1) curved document pages form developable surfaces which can be approximated by planar strips, and 2) the projected image of printed textual content on the page constrains the underlying surface shape by the parallelism, geodesic, and equidistant properties of text lines (see [6] for a discussion on geodesic texture flow and developable surface under perspective projection). Compared to other's work, our method does not require special equipment or camera calibration, can be applied to general warped documents, and can work on partially occluded documents.

In [7] we have discussed the details of image processing, shown that page shape can be estimated, and obtained much higher OCR rates from unwarped images. However, shape information is not explicitly expressed in [7], which makes it difficult for evaluation, nor is the estimation process globally optimal with regard to developable property and text property. In this paper we present our recent progress using global shape optimization which gives significant improvement over [7].

Section 2 and Section 3 briefly recalls the work in [7]. In Section 4 we describe the initialization and optimization of shape estimation. Section 5 discusses experiment results and finally Section 6 concludes the paper.

2 Problem Modeling

The shape of a smoothly rolled document page can be modeled by a developable surface. In [7] we show that a developable surface can be approximated by planar strips that come from the family of its tangent planes (see Fig 1), which can be fully described by a set of reference points $\{P_i\}$, and surface normals $\{\mathbf{N}_i\}$.

For documents that are covered by printed text we can define two texture flows on the surface, both in 3D space and in 2D projected images. One corresponds to the text line direction, which we call *3D(2D) major texture flow* and denote by **T**(or **t**); the other corresponds to the vertical char-

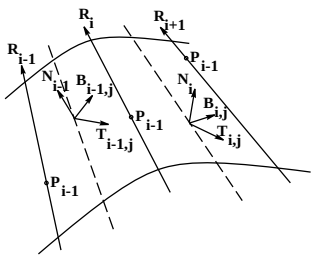
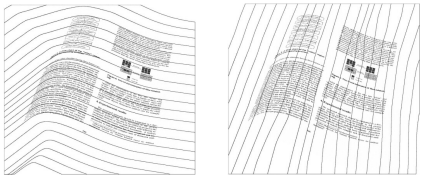


Figure 1. A developable surface can be approximated by planar strips (for variable definitions see Section 4)



(a) (b)

Figure 2. Texture flow detection: (a) major texture flow (b) minor texture flow

acter stroke direction, which we call *3D(2D) minor texture flow*, denoted by **B**(or **b**). For a 3D ruling **R** we also define a *2D projected ruling* **r** in the image.

3 Image Processing

Based on the developable surface model, our approach is to segment the surface by a group of rulings, approximate the pieces in between the rulings by planar strips, and unroll the page strip by strip. In this section we will very briefly go over the image processing step that support the shape estimation in next section (for more details see [7]).

The first step is to find the printed text area because we will be using the properties of printed text and any non-text element may cause unexpected results. An adaptive thresholding [9] inside the text area gives us the binary text image, and all following computation is based on the binary text image. Secondly we extract the two 2D texture flows. We divide the image into small blocks, and use projection profile analysis to compute the local texture orientations. A relaxation process resolves any conflict among neighboring blocks, and results in two texture flow fields (Fig. 2).

Projected ruling detection is based on the property that texture flow patterns along a projected ruling is more consistent than that along an arbitrary line. For each ruling, its vanishing point is estimated by the fact that printed text lines are usually equally spaced and the invariance of cross ratio under perspective projection.

4 Page Shape Estimation

In [7] we described how to iteratively optimize the shape of a document under developable property and text property constraints. However, there are two problems. First, we do not get an explicit surface normal for each strip; instead we compute *horizontal* and *vertical* vanishing points. Second, we do not have an explicit objective function and therefore the iterative process does not have an explicit measurement of the progress. In this paper, we address these two problems by formally introducing several constraints defined on surface normals as well as focal length of the camera, and an objective function based on the constraints. By optimizing the objective function we obtain explicit surface normals and focal length.

4.1 Constraints

It is difficult to estimate the normal of each planar strip only using local features. Fortunately there are strong global constraints imposed by the developable property and text properties. First we will define the variables (see Fig. 1), and then introduce the constraints.

- Wanted unknowns:
  - 3D normals: $\{\mathbf{N}_i\}_{i=1}^{L}$, where L is the number of strips
  - 3D reference points: $\{P_i\}_{i=1}^{L+1}$
  - Focal length: f_0.
- Preprocessing results and known variables:
  - Projected rulings: $\{\mathbf{r}_i\}_{i=1}^{L+1}$
  - Projected reference points: $\{p_i\}_{i=1}^{L+1}$
  - Projected texture flow: **t** and **b** all over the image
- Other related variables:
  - 3D rulings: $\{\mathbf{R}_i\}_{i=1}^{L+1}$
  - 3D texture flow: For the i-th strip, we select a group of J_i sample points inside the strip, and define $\mathbf{T}_{ij}$ as the 3D major texture flow vector at the j-th point, and $\mathbf{B}_{ij}$ as the minor texture flow vector.
  - 3D viewing direction vector: For the j-th sample point in the i-th strip, we define $\mathbf{V}_{ij}$ as its viewing direction vector with respect to the camera's optical center.

All the vectors are of unit length.

Suppose $\eta(\cdot)$ represents the normalization operator where $\eta(\mathbf{v}) = \mathbf{v}/|\mathbf{v}|$, then the 3D vectors are related to their projections in the image by the following equations:

$$\begin{aligned} \mathbf{R}_i &= \eta((\mathbf{r}_i \times \mathbf{V}_i) \times (\mathbf{N}_i + \mathbf{N}_{i-1})/2) \\ \mathbf{T}_{ij} &= \eta((\mathbf{t}_{ij} \times \mathbf{V}_{ij}) \times \mathbf{N}_i) \\ \mathbf{B}_{ij} &= \eta((\mathbf{b}_{ij} \times \mathbf{V}_{ij}) \times \mathbf{N}_i) \end{aligned}$$

Note that in the equation relating **R** and **r** we use $\mathbf{N}_i + \mathbf{N}_{i+1}$ to approximate the surface normal along $\mathbf{R}_i$.

Without loss of generality we can assume that P_i are on the rulings. By the continuity property of the planar strips it

is easy to see that once we have obtained surface normals, focal length, and the depth of any particular P_{i_0}, the rest P_i are fully determined.

There are four constraints that we can derive from the developable property of the page and the property of text documents:

• Orthogonality between surface normals and rulings: Ideally, we would want $\mathbf{N}_{i-1} \cdot \mathbf{R}_i = \mathbf{N}_i \cdot \mathbf{R}_i = 0$. Since we have fixed $\mathbf{R}_i$ to be orthogonal to $\mathbf{N}_{i-1}+\mathbf{N}_i$, we only need to check $\mathbf{R}_i \cdot (\mathbf{N}_i - \mathbf{N}_{i-1})$. We define $\mu_1 = \sum_{i=1}^{L-1} (\Delta \mathbf{N}_i \cdot \mathbf{R}_i)^2$ where $\Delta \mathbf{N}_i = \mathbf{N}_i - \mathbf{N}_{i-1}$, and ideally $\mu_1 = 0$.

• Parallelism of text lines inside each strip: Text line directions are represented by $\mathbf{T}_{ij}$. We use $\mu_2 = \sum_i \sum_j |\mathbf{T}_{ij} - \overline{\mathbf{T}}_i|$, where $\overline{\mathbf{T}}_i$ is the average of all $\mathbf{T}_{ij}$ within the i-th strip, to measure their parallelism. Ideally $\mu_2 = 0$.

• Geodesic property of text lines crossing two neighboring strips: The text lines on two neighboring strips form two different angles with the 3D ruling that separates the strips. After unwarping, the angles do not change. If the sum of the two angles is π, it means the text line is straight in the unwarped image. We use $\mu_3 = \sum_i ((\overline{\mathbf{T}}_{i+1} - \overline{\mathbf{T}}_i) \cdot \mathbf{R}_i)^2$ to measure the straightness, which ideally is zero.

• Orthogonality between text line direction and vertical stroke direction: The orthogonality can be measured by $\mu_4 = \sum_i \sum_j |T_{ij}^{\tau} B_{ij}|$, which in the idea case should be zero.

In our experiment we embedded two additional constraints:

• Smoothness: We use $\mu_5 = \sum_i |\Delta \mathbf{N}_i|$ to measure the smoothness of the surface. A large value indicates abrupt changes in normals of neighboring strips and therefore should be avoided.

• Unit length: Each normal should be of unit length. We measure this by $\mu_6 = \sum_i (1 - |\mathbf{N}_i|)^2$.

The overall optimization objective function is the weighted sum of all constraint measurements,

$$F(\mathbf{X}) = \sum_{i=1}^{6} \alpha_i \mu_i$$

where $\mathbf{X}$ represents all normals and the focal length, and α_i are weights.

Overall, given $\{\mathbf{r}_i\}$, $\{\mathbf{t}_{ij}\}$ and $\{\mathbf{b}_{ij}\}$, the objective function is fully determined by the unknown $\{\mathbf{N}_i\}$ and f_0. The optimal set of $\{\mathbf{N}_i^*\}$ and f_0^* should minimize F.

4.2 Shape Initialization and Optimization

A good initial value of $\mathbf{X}$ is essential for optimizing the highly non-linear objective function. Such initial values can be obtained using the estimated vanishing points of rulings. These vanishing points, when focal length is given, determine the direction of 3D rulings. Since surface normals are orthogonal to 3D rulings, this eliminates one degree of freedom from the unknown normals. The remaining degree of freedom allow a normal to rotate in the plane orthogonal the the ruling. So the objective function is decided by a set of rotation angles. Furthermore, the computation of the objective function involves either each individual normal (μ_2, μ_4, μ_6), or two neighboring normals (μ_1, μ_3, μ_5). Therefore, we can use dynamic programming (DP) search to find the best set of rotation angles that gives the minimum objection function output.

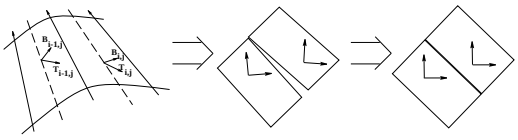


Figure 3. After the surface is developed, post-processing ensures that strips fit each other seamlessly and major texture flow direction is horizontal

The focal length is not covered by the DP search, however. It is independent of the surface normals, and we have to perform an exhaustive search for the initial focal length. More specifically, we select a set of possible focal lengths that are constrained by the physical lens specification, and for each value we find the "best" surface normals, and compute the objective function. We fit a 3rd order polynomial curve to the objective function values vs. the focal lengths, and use the curve to find the "best" focal length. Then we compute the "best" normals for this focal length, and take them as the initial values for non-linear optimization process.

Our non-linear optimization module is based on the optimization toolbox in MATLAB, which is fairly fast and produces good result as long as the initial point is reasonably close to the true solution.

After we have estimated the surface normals and focal length, we can arbitrarily select the depth of any one reference point, which determines the depth of the other reference points, and thus fully determine the 3D position of planar strips. The planar strips can then be mapped to a flat plane, placed side by side to form the flat document. Due to the errors in shape estimation and the fact that the document page in real world may be not perfectly developable, some postprocessing is required to make sure that the strips fit each other and that restored text lines are horizontal and continuous across the whole unwarped image (see Fig. 3).

5 Experiment Results

We have applied our method to both synthetic and real images. The synthetic images are generated by warping a flat document image around a predefined developable surface and projecting it onto the image plane of a pinhole

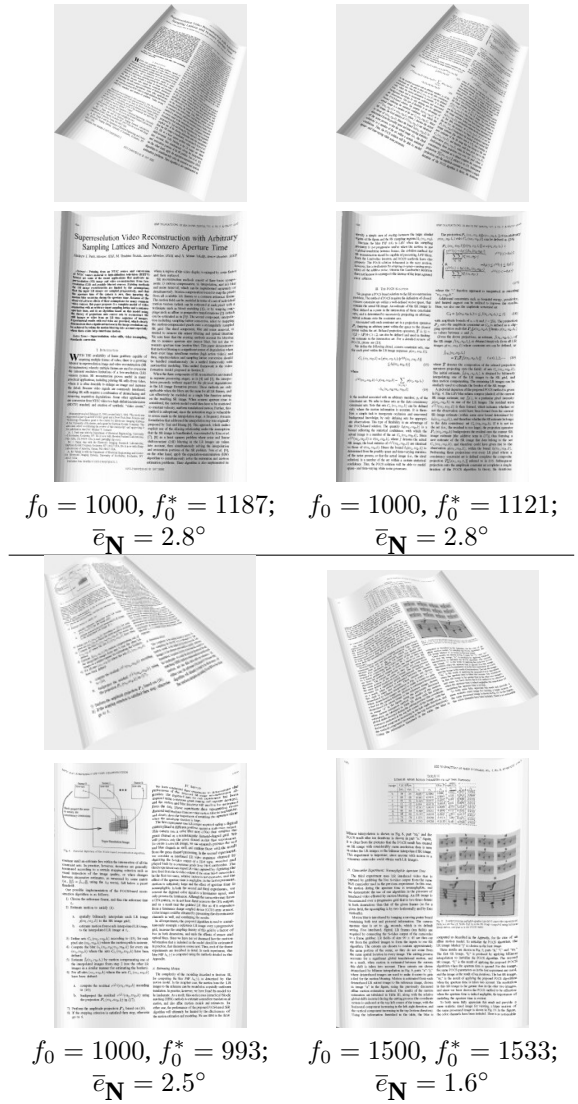
camera. With synthetic data, we can evaluate the estimated results such as texture flows, projected rulings, ruling vanishing points, surface normals and focal length against the ground truth.

Fig. 4 shows four synthetic images of warped documents and the unwarped images. It also compares the ground truth focal lengths to the estimates, and shows the average error of surface normals. In Fig. 5 two real images of warped documents and their unwarped images are shown. As we can see in both Fig. 4 and Fig. 5, the text lines are mostly straight and horizontal. Some text line still have some curve due to the errors in major texture flow detection, which is more evident around corners or margins.

The errors in estimated surface normals are measured by the angles between them and corresponding true surface normals. We do not, however, measure the focal length estimation by the difference between it and the true value, because when focal length is large, the reconstructed shape is less sensitive to the change in focal length, which means we can tolerate a larger error. To factor that into the evaluation, we use *view angle* defined as following: a view angle for a given focal length f_0 is $2\mathrm{atan}(d/f_0)$ where d is the largest distance from any point in the image to the optical axis (or, roughly half of the image's dimension). The change in view angle w.r.t f_0 vanishes as f_0 increases, and thus is a better performance measurement.

Although we do not have explicit surface normal estimation from the method in [7], in order to compare it with the results obtained by global optimization we construct approximated surface normals from the results of previous method. In Table 1 we list the mean and standard deviation of view angle errors and surface normal errors from the estimation by previous method, by initialization and final optimization of current method. The results of previous method is obtained from 32 images in which documents contain only text, and the results of global optimization is obtained from 44 images in which documents contain figures and tables. Because of the non-text elements, these 44 images are inherently more difficult to process. Nevertheless, our current method has a great lead over the previous one. This is due in part to the refinements we made in other parts of our code but the main reason is still the new optimization method, especially the shape initialization by DP. In [7] without an explicit objective function representing all constraints we initialized the shape based on local information, and it is not surprising that the initial shape is not as good as that obtained by DP that takes into account global information. The benefit of an explicit objective function is also manifested by the improvement from the initial shape to the final result.

Currently, the parameters involved in image processing and shape estimation stages are manually set. The weight factors α_i are set by experiment in such a way that μ_i be-



$f_0 = 1000$, $f_0^* = 1187$; $\overline{e}_{\mathbf{N}} = 2.8°$

$f_0 = 1000$, $f_0^* = 1121$; $\overline{e}_{\mathbf{N}} = 2.8°$

$f_0 = 1000$, $f_0^* = 993$; $\overline{e}_{\mathbf{N}} = 2.5°$

$f_0 = 1500$, $f_0^* = 1533$; $\overline{e}_{\mathbf{N}} = 1.6°$

Figure 4. Unwarped synthetic document images: f_0 is true focal length and f_o^* is estimation (both in pixel unit); $\overline{e}_{\mathbf{N}}$ is the average normal error measured in degree

come comparable to each other. In the future we will address the automatic parameter selection problem. Nevertheless, among several different settings for each procedure we have not found significant changes in the results. In our experiments we used very conservative parameter values in order to ensure accuracy for arbitrary images. In practice with some knowledge of the image, it is possible to tighten some parameters for better speed.

Among the images that have unsatisfactory results, the major problem comes from the text area detection step. If a background object or a picture in the document gets identified as text, it can interrupt the texture flow detection, and

	Previous method [7] (32 tests)	With global optimization (44 tests)	
		Initial estimation	Final optimization
Ave. view angle error	12.7	8.3	7.3
Std. view angle error	20.5	8.0	7.6
Ave. surface normal error	14.0	6.5	4.8
Std. surface normal error	13.9	4.4	3.6

Table 1. Shape estimation evaluation (error measured in degrees)

Figure 5. Unwarped real document images

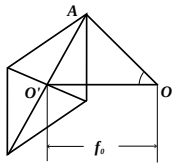


Figure 6. View angle definition: OO' is optical axis; f_0 is focal length; A is the farthest point in the image from OO'.

break the following procedures. Since many researchers have proposed various techniques for identifying text in images, we believe that we can solve this problem by choosing one of them.

6 Conclusion

In this paper we describe how we optimize the page shape estimation globally for unwarping images of curved documents captured by cameras. The document surface is modeled by a developable surface, and we show that the textual content (text lines in particular) provides enough information for recovering the page shape. Compared to the results of our previous method, improvement is obtained by introducing a global optimization into the shape estimation process. From the OCR point of view, the geometry of the reconstructed page is definitely within the acceptable tolerance. However, other challenges still exist, including varying shade, non-uniform blur, fusion of multiple views. We will address them in our future work.

References

[1] M. S. Brown and W. B. Seales. Image restoration of arbitrarily warped documents. *IEEE Trans. PAMI*, 26(10):1295–1306, October 2004.

[2] H. Cao, X. Ding, and C. Liu. Rectifying the bound document image captured by the camera: A model based approach. In *Proc. ICDAR*, pages 71–75, 2003.

[3] P. Clark and M. Mirmehdi. On the recovery of oriented documents from single images. In *Proc. Adv. Concepts for Intelligent Vision Sys.*, pages 190–197, 2002.

[4] C. R. Dance. Perspective estimation for document images. In *Proceedings of SPIE Document Recognition and Retrieval IX*, volume 4670, pages 244–254, 2002.

[5] N. Gumerov, A. Zandifar, R. Duraiswarni, and L. S. Davis. Structure of applicable surfaces from single views. In *Proc. ECCV*, pages 482–496, 2004.

[6] D. C. Knill. Contour into texture: Information content of surface contours and texture flow. *J. Opt. Soc. Am. Ass.*, 18(1):12–35, Jan 2001.

[7] J. Liang, D. DeMenthon, and D. Doermann. Flattening curved documents in images. In *Proc. CVPR*, pages 338–345, 2005.

[8] M. Pilu. Undoing paper curl distortion using applicable surfaces. In *Proc. CVPR*, volume 1, pages 67–72, 2001.

[9] Ø. D. Trier and T. Taxt. Evaluation of binarization methods for document images. *IEEE Trans. PAMI*, 12(3):312–315, 1995.

[10] Y.-C. Tsoi and M. S. Brown. Geometric and shading correction for images of printed materials a unified approach using boundary. In *Proc. CVPR*, pages 240–246, 2004.

[11] Z. Zhang and C. L. Tan. Correcting document image warping based on regression of curved text lines. In *Proc. ICDAR*, volume 1, pages 589–593, 2003.

[12] Z. Zhang, C. L. Tan, and L. Fan. Restoration of curved document images through 3D shape modeling. In *Proc. CVPR*, pages 10–15, 2004.

A Robust Method for Tracking Scene Text in Video Imagery

Gregory K. Myers and Brian Burns
SRI International
Menlo Park, CA 94025 USA

Abstract

Text on planar surfaces in 3-D scenes in video imagery can undergo complex apparent motion and distortion as the surfaces move relative to the camera. Tracking such text and its motion through a contiguous sequence of video frames in which it is visible is desirable primarily for two reasons. First, reliable tracking of text enables the images of text persisting across multiple frames to be grouped, processed, and understood as a single unit. Second, text tracking aids the mapping of corresponding text and background pixels across multiple frames to enhance image quality and resolution before character recognition. Existing text tracking approaches, however, are limited to approximate pixel-based correspondences of adjacent frames without any explicit, rigorous modeling of 3-D scene geometry. To this end, we describe an approach that tracks planar regions of scene text that can undergo arbitrary 3-D rigid motion and scale changes. Our approach computes homographies on blocks of contiguous frames simultaneously using a combination of factorization and robust statistical methods. In spite of low resolution and noisy imagery, this approach produces a more accurate and stable motion estimate than existing methods using only two adjacent frames. In addition, our method is robust enough to tolerate imperfections in the spatial localization of text. Our results demonstrate that the mean offset pixel error of our tracker is as small as 1.1 pixels.

1. Introduction

Recognition of text that appears in real-world scenes, such as protest signs and name tags, is of utility for automated characterization and annotation of video imagery because of its valuable contribution to the video content. Such a capability enables information retrieval systems to index videos in a convenient and meaningful way for later reference. Text in video can take the form of artificially generated text that is overlaid on the imagery (such as superimposed captions in broadcast news programs and other commercially produced videos), or text that is part of the video scene itself (such as a sign outside a place of business or placards in front of conference participants). In this work we focus on scene text.

The recognition of scene text in video imagery involves several major processing steps, including text detection, text tracking, and OCR. This paper focuses on tracking of scene text. Since the same text can be visible on multiple consecutive frames in video, tracking of text is desirable so that all the images of text in the multiple frames can be grouped, processed, and understood as a single unit. In real-time applications with live video, such as portable road sign translators for tourists and soldiers, recognized text can be treated as an event that immediately triggers additional automated processes, such as machine translation and speech synthesis. An automated surveillance system may trigger database lookup of a recognized vehicle license plate number. Therefore, reliable means of text tracking is important to ensure a single response for each distinct text event.

There are two aspects to text tracking: (1) frame-to-frame association of text regions, and (2) frame-to-frame motion estimation of each region. The former involves determining the temporal continuity of regions and assigning an ID to each tracked text region. The latter involves computing a pixel-to-pixel mapping to establish localized frame-to-frame geometrical correspondence. Frame-to-frame association enables a single OCR result to be produced and reported for multiple contiguous appearances of the same text object. Furthermore, frame-to-frame geometrical corresponddence is required for the video OCR process to take advantage of temporal redundancy of text that appears in multiple frames to create an enhanced image before subsequent OCR processing. Such multiframe integration includes multiple image averaging [1,2] and superresolution [3,4].

Scene text has a number of characteristics that make it difficult to detect and track. Scene text exists in 3-D space and can be slanted, tilted, or modulated by the surface of the object on which the text is printed. Although scene text often lies in a plane, several types of distortion can be introduced when the plane containing the text is at an angle relative to the image plane. In the most general case, the distortion is described as a projective transformation between the plane containing the text and the image plane [5]. The tracking of scene text is complicated by the fact that its appearance can change drastically during its presence in the video. In addition to camera pan, tilt, rotation, and zoom affecting the text, the size and viewing angle of text on moving objects can vary significantly.

Figure 1. Subarea in consecutive frames of video shot from a moving vehicle.

In addition, video sequences that are generated with a nonstationary camera (e.g., a handheld camcorder or vehicle-mounted video camera) may contain a significant amount of random, jerky camera motion, which can blur the image of the text, cause interlace shearing, and/or make the text hard to track from frame to frame. This is especially true of imagery collected at high-magnification lens settings. For example, Figure 1 illustrates some of these artifacts on a subarea extracted from seven consecutive frames of a video captured from a moving vehicle. Furthermore, a camera's automatic focus mechanism may take time to adjust while the camera zooms and pans or the contents of the scene change or move, resulting in some intermittently out-of-focus frames. Scene text may be partially obscured temporarily by objects moving in the foreground (e.g., a person walking in front of a sign).

2. Prior work

Due to the difficulties in tracking scene text that were outlined in Section 1, many previous text tracking approaches were designed for overlay text [6,2,7], assumed the text was horizontally oriented, and used a translational motion model. Crandall et al. [8] used motion vectors in MPEG compressed video and a least-square-error search of a small neighborhood for tracking of text regions; for text captions that rotated or changed scale, features extracted from the connected components within the text region were matched in consecutive frames. Li and Doermann [1] and Li et al. [9] aligned blocks of scene text in adjacent frames using a translational motion model and correlation within a search window, and when the detected motion did not fit the translational model, the contour of the text region was determined by tracking a blob created by horizontal smearing of edges found in the text region. Tracking failures were detected [1] by using criteria such as straightness of the motion trail of the text region center.

In all the previous work cited above, motion was computed from adjacent frames only; therefore, these tracking methods may fail if the video is intermittently degraded, or if portions of the text regions are temporarily occluded. In addition, previous methods that relied on trajectory-based motion prediction would have difficulty tracking text in video with random, jerky camera motion. Furthermore, previous methods assumed that the text region boundaries are accurately defined. However, text detectors may not always find text region boundaries reliably. For example, non-text image patterns that have characteristics similar to text and are adjacent to text in the scene might incorrectly be included as part of the text region in some frames. Finally, previous methods estimated motion only in the 2-D image plane and did not attempt to explicitly model the 3-D motion of text in the scene.

In addition to the above text-specific tracking approaches, previous work on tracking regions in general include systems based on the brightness constraint equation [10–12], and motion estimation from two-frame point correspondences [13]. In the former method, the pixel intensity difference between two frames is expressed as a function of the motion between the frames and is minimized using nonlinear optimization techniques. This method, for the case of planar motion, has been extended to multiple frames in [14] using factorization. The expression assumes that all changes in intensity at each pixel are due to motion, instead of brightness change, occlusion, and other effects. This makes the method problematic in situations where there is clutter and occluding surfaces in the region that are not moving the same way. In addition, for the motion to be solvable by successive linearization, it has to either be small, or the region has to be large enough relative to the motion to be suitable for multiscale methods. This is not always the case for text regions of short height and viewed through a handheld camera. The other approach [13], tracking individual points followed by robust estimation of the whole region motion, is more suitable when there is clutter, occlusion, and large motion. In the method of Hartley and Zisserman [13], not all of the points have to be successfully tracked or consistent with the motion, as long as there is a large enough inlier set that is. However, the method proposed by Hartley and Zisserman [13] estimates tracking descriptors only on two frames at a time. Since text is often visible on 30 or more video frames (at 30 fps), estimation based on just two frames is limiting and is clearly not optimal.

3. Proposed approach

One of the primary contributions of the current work is its generalization of the work described in Hartley and Zisserman [13] to simultaneous multiple frame analysis. In our work we track points of interest across all the frames being considered, and then, within each detected text region, estimate the planar transformation simultaneously and robustly over blocks of multiple frames. We assume that the region to be tracked is planar in the scene and that there are a sufficient

number of points in the region with enough texture to be tracked over the frames that are compared. For six parameter affine motion, we require a minimum of three points tracked over all frames in a block. These assumptions are typically valid for text regions, since text regions are typically highly textured and bimodal in intensity. Using the process described below, the region of interest is tracked a block of images at a time. The motion in the previous block is used to locate the region in the first frame of the next block, and so on, until the end of the video or a point is reached where the minimum set of inliers goes below a threshold (four tracked points), indicating that the region cannot be tracked further. Ideally, the block size should be selected automatically to maintain a sufficient point correspondence count. In the experiments discussed here, the block sizes were manually fixed to five or ten frames, depending on the magnitude of image change and the size of the region.

3.1. Point location and tracking

Tracking a point has three steps: (1) initially selecting the point in the image, (2) localizing it in subsequent frames, and (3) determining when it is no longer trackable (termination). In our system, different points are selected and terminated in different frames, so that any given pair of relatively close frames will have points in common. Points are selected using two criteria: *texture* and *coverage*. The presence of *texture* aids reliable localization of the point across frames, which is true if the local intensity variation in the image in different directions is high. The peaks of the Laplacian-of-Gaussian image and/or the peaks of the Shi and Tomasi texture operator [11] can be used as indicators of high texture regions. We have found that these two methods have similar and reasonable results. When there are points detected and tracked over all parts of the region, the region is said to have high *coverage*. To improve coverage, we iteratively select detected points, greatest texture first, until the maximum distance from each pixel to the nearest detected point is below a preselected minimum threshold. Selected points are tracked and localized in subsequent frames using normalized correlation of a small image patch centered at the current position of the point. The new position of the point is located to sub-pixel precision by quadratic interpolation of the correlation surface. The tracking of a point is terminated when the correlation drops below a threshold, which is typically 0.65 out of a range of [−1, 1], where 1 is perfect. The image patch size is a function of the magnitude of the image transformation, the size of the scene surfaces, the magnitude of visual texture, and the video quality. Small patches tend to be tolerant of large image transformations and complex scene geometry, but less tolerant of poor image texture and poor video quality. Image patches of 15×15 pixels (relatively small) were used for the experiments discussed here since text tends to support good texture and the video quality was minimally adequate for this size.

3.2. Point selection and motion estimation

A large number of the points in the automatically extracted region of interest may be unusable for the estimation of the region motion for various reasons: They may be on a different surface, their trajectories may be misestimated due to noise and low resolution, or the points may be optical artifacts such as specularities and moving shadows. To make text tracking reliable, the unusable points must be detected as outliers. This outlier detection is challenged by the fact that there are many degrees of freedom (six or eight) in the projected motion of the scene surface.

Once points are tracked, we estimate the transformation of the whole region in blocks of multiple frames. Our method combines two approaches: (1) robust parameter estimation, such as RANSAC [15], which simultaneously estimates the parameters and determines the inlier data set; and (2) simultaneous, multiframe reconstruction of the projecting points and the projecting transformations for all the frames, which further reduces the transformation error and reduces outliers.

3.2.1. Application of RANSAC. Applying RANSAC, our algorithm randomly selects minimal subsets (of 3 points each) of point tracks in the frame block, estimates the projective transformation for each frame given the set (using the factorization approach discussed below), and then counts the number of point tracks that are consistent with the transformations (inliers). The largest inlier set is then used for the multiframe reconstruction, again using the factorization approach. Point track consistency is measured as the root-mean-squared projection error across all the frames after reconstructing the scene plane position of this point and reprojecting it in all frames. A projection error of two pixels was used as a consistency cutoff.

3.2.2. Multiframe reconstruction and motion estimation. The multiframe reconstruction of the scene plane and the projective transformations is done using a 2-D version of the 3-D factorization technique developed in [16]. Since the scene structure we are recovering is planar, we can force the factorized matrices to be of rank 2, which further constrains the reconstruction (beyond the 3-D case) and leads to a more accurate solution for our purposes. As in the original 3-D version, we construct a $2m \times n$ data matrix W, where each column is a point track $[x_1, y_1, x_2, y_2, \ldots, x_m, y_m]^T$, (x_i, y_i) is the point position in the ith frame minus the point centroid of that frame, and there are m frames and n points. Assuming an affine camera model,

$W = M \times S$, where M is a $2m \times 2$ motion matrix, with each pair of rows i representing the nontranslational components of the affine projection for frame i, and where S is a $2 \times n$ matrix with each column j representing point j's two component position on the scene plane. Using SVD, W can be factorized into $U \times D \times V^T$, where $U \times \sqrt{D}$ and $\sqrt{D} \times V^T$ are M and S up to a 2-D affine transformation. Since we are only interested in the 2-D motion between frames, this ambiguity can be ignored. The motion between the first frame 0 in the block and any other frame t is $H(0,t) = Q_t \times Q_0^{-1}$, where Q_t and Q_0 are the affine transformations between the scene plane and frames t and 0 respectively. Q_t is constructed by using the point centroid of the frame as the translation component and rows $2t$ and $2t - 1$ of M for the other four parameters. Q_0 is constructed analogously.

4. Preliminary Results

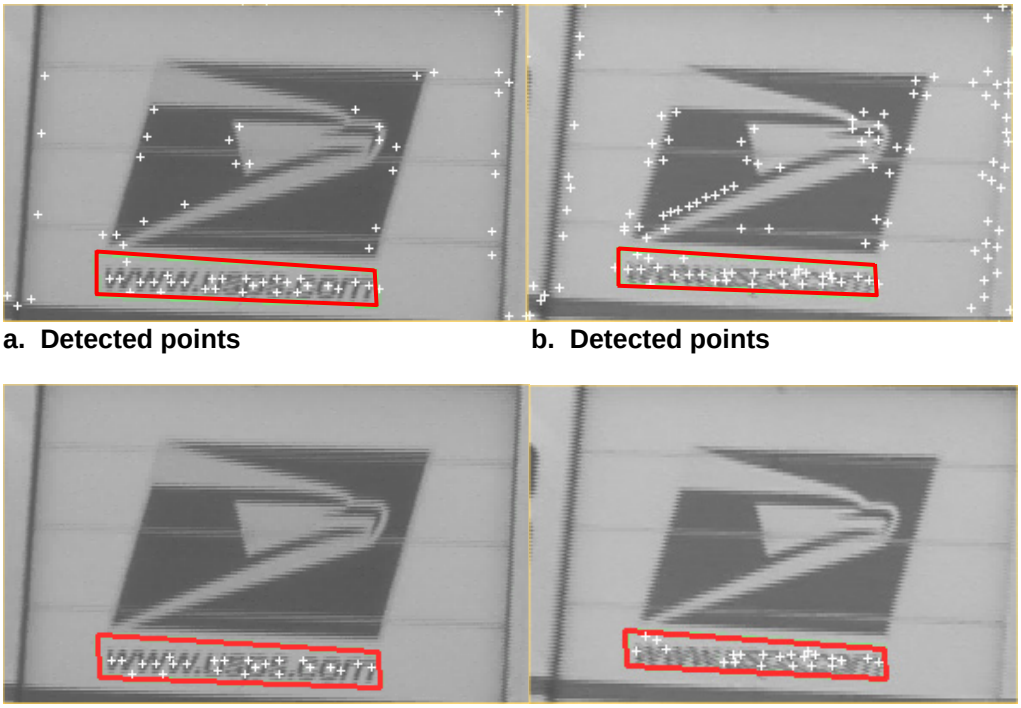
Figures 2 through 4 show some examples of the tracking performance. For each of these video sequences, the tracker was initialized by manually specifying a bounding box to simulate the results of a text detection process. Figure 2 shows the results of tracking the text region shown in Figure 1. Figures 3 and 4 show three frames extracted from two other sequences. The imagery in Figures 2 and 3 was taken with a handheld video camera from a moving vehicle. The middle frame in Figure 3 shows one of three consecutive frames in which the text region (the "Hallmark Cards" sign) was occluded by a pole in the foreground. The text region in Figure 4 was successfully tracked throughout the zoom out by a factor of 6, even to the point of the text being unreadable. To assess performance quantitatively, we calculated the mean offset pixel error by comparing the relative positions of 4 points in the first and last frames of 9 tracked text regions, each in a different video sequence. The mean offset pixel error of the tracker was 1.1 pixels.

Figure 5 shows some of the details of the point detection and selection. Figures 5a and 5b show all of the points that were detected on a portion of the back of a moving postal truck in video frames taken about 4 seconds apart; Figures 5c and 5d show the inlier points used for tracking the text region in frames 5a and 5b, respectively. Notice that only a subset of the detected points is used for tracking, and the set of points detected and tracked at the two video frame times is not identical.

Figure 2. Example with jerky camera motion.

Figure 3. Example with jerky camera motion and occlusion.

Figure 4. Example with large change of scale.



a. Detected points b. Detected points

c. Inlier tracked points d. Inlier tracked points

Figure 5. Details of point detection and selection on moving postal truck.

5. Discussion

Our method ascertains that, in a block of frames of nontrivial size, the trajectory of every point is consistent with the region motion over several frames. The likelihood of the trajectory of an outlier point being spuriously consistent over many frames is much lower than the chance of it being consistent over a single frame pair. Therefore, the multiframe nature of the proposed reconstruction approach assists the detection of outliers during the motion estimation process compared with the methods considering only two frames at a time [e.g., 13]. The multiframe reconstruction approach being used here, factorization in combination with robust statistical search methods, also reduces the effects of noise or low magnitude tracking errors in the inlier point set. Using factorization, a least squares reconstruction of the original scene points is generated from all the projections in the frame block, simultaneously with the projecting homographies from this point reconstruction to the individual frames. Thus, the estimated transformations are with respect to a reconstructed point set, not between two noisy point sets as in the two-frame approach.

As in 3-D factorization, our approach can be extended to handle the projective case using an iterative approximation [13]. However, the full eight parameter projective model should be used with caution on small text regions in noisy video, and it is typically unnecessary for tracking the text.

Unlike tracking methods such as Kalman filtering that rely on a particular motion model, this method does not require any knowledge about how the motion in different frames is related, and therefore can track text in video with random, jerky camera motion.

Future designs will include enhancements of both the point tracking and the geometric analysis. Currently, the point tracker can fail when there are temporary interruptions in the point visibility or quality of the video since the tracking stops once the correlation drops below a threshold. Instead, the tracker should continue to search for the point in subsequent frames and report only the parts of the track with high enough correlation. The geometric analysis can be improved by not forcing the factorization step to use exactly every frame in the block. The quality could improve by selecting only frames of high enough quality, essentially an outlier detection process for frames analoguous to the outlier detection already performed over the points.

6. Summary

Text on planar surfaces in 3-D scenes in video imagery can undergo complex apparent motion and distortion as the surfaces move relative to the camera. Tracking such text and its motion through a contiguous sequence of video frames in which it is visible is desirable primarily for two reasons. First, reliable tracking of text enables the images of text persisting across multiple frames to be grouped, processed, and understood as a single unit. Second, text tracking aids

the mapping of corresponding text and background pixels across multiple frames to enhance image quality and resolution before character recognition. Existing text tracking approaches, however, are limited to approximate pixel-based correspondences of adjacent frames without any explicit, rigorous modeling of 3-D scene geometry. To this end, we describe an approach that tracks planar regions of scene text that can undergo arbitrary 3-D rigid motion and scale changes. Our approach computes homographies on blocks of contiguous frames simultaneously using a combination of factorization and robust statistical methods. In spite of low resolution and noisy imagery, this approach produces a more accurate and stable motion estimate than existing methods using only two adjacent frames. In addition, our method is robust enough to tolerate imperfections in the spatial localization of text. Our results demonstrate that the mean offset pixel error of our tracker is as small as 1.1 pixels.

7. Acknowledgment

This material is based on work supported in whole by the U.S. Government. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government.

8. References

[1] H. Li and D.S. Doermann, "Text Enhancement in Digital Video Using Multiple Frame Integration", *Proc. ACM Multimedia '99*, Orlando, Florida, 1999, pp. 19–22.

[2] C. Wolf, J.-M. Jolion, and F. Chassaing, "Text Localization, an Enhancement, and Binarization in Multimedia Documents", *Proc. of International Conference on Pattern Recognition (ICPR),* Vol. 4, August 2002, pp. 1037–1040,.

[3] Huiping Li and David Doermann, "Superresolution-Based Enhancement of Text in Digital Video", International Conference on Pattern Recognition (ICPR'00), Barcelona, Spain, 2000.

[4] Katherine Donaldson and Gregory K. Myers, "Bayesian Super-Resolution of Text in Video with a Text-Specific Bimodal Prior", *International Journal on Document Analysis and Recognition*, Volume 7, Numbers 2-3, July 2005, pp. 159 – 167.

[5] G.K. Myers, R.C. Bolles, Q.-T. Luong, and J.A. Herson, "Recognition of 3-D Scene Text," Fourth Symposium on Document Image Understanding Technology (SDIUT01), Columbia, Maryland, April 2001, pp. 85–99 (http://www.esd.sri.com/projects/vace/docs/SDIUTMyers2.pdf).

[6] R. Lienhart, "Indexing and Retrieval of Digital Video Sequences based on Automatic Text Recognition", in *4th ACM International Multimedia Conference*, Boston November 1996.

[7] R. Lienhart and A. Wernicke, "Localizing and Segmenting Text in Images and Video", *IEEE Trans on Circuits and Systems for Video Technology*, Vol. 12, No. 4 April 2002.

[8] David Crandall, Sameer Antani, Rangachar Kasturi, "Extraction of Special Effects Caption Text Events from Digital Video", IJDAR 5(2-3): 138–157, 2003.

[9] H. Li, D. Doermann, and O. Kia, "Automatic Text Detection and Tracking in Digital Video", *IEEE Trans. Image Processing—Special Issue on Image and Video Processing for Digital Libraries*, Vol. 9, No. 1, 2000, pp. 147–55.

[10] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", *IJCAI*, 1981.

[11] J. Shi and C. Tomasi, "Good Features to Track", *IEEE Conf on CVPR*, June 1994.

[12] M. Irani and P. Anandan, "About Direct Methods", in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski (eds), Springer, June 2000.

[13] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University *Press*, 2000.

[14] L. Zelnik-Manor and M. Irani, "Multi-frame Estimation of Planar Motion", IEEE PAMI, 22(10):1–12, October 2000.

[15] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Comm ACM*, 24 (6):381–395, 1981.

[16] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams Under Orthography", *IJCV*, 9(2):137–154, November 1992.

Section II

Recognition

Grayscale Feature Combination in Recognition based Segmentation for Degraded Text String Recognition

Jun Sun

Fujitsu R&D Center
Eagle Plaza B10th floor, Xiaoyun Rd. No.26
Chaoyang Dist.Beijing,100016, P.R.China
sunjun@frdc.fujitsu.com

Yoshinobu Hotta, Katsuhito Fujimoto
Yutaka Katsuyama, Satoshi Naoi
Fujitsu Laboratories Ltd
Kawasaki, Japan
y.hotta, fujimoto.kat@jp.fujitsu.com
katsuyama, naoi.satoshi@jp.fujitsu.com

Abstract

Grayscale feature is very effective for degraded character recognition. While many papers focus on different feature extraction algorithms on single character recognition, few deals with the impact of the selected feature on segmentation. For recognition-based segmentation, a good recognition performance on single character may not always have good performance on segmentation. In this paper, two types of grayscale feature, the R-Feature and the S-Feature, are proposed based on dual-eigenspace decomposition. The R-Feature is suitable for single character recognition. The S-Feature is suitable for text string segmentation. These two feature are combined to further improve the performance for degraded Japanese text string recognition.

1. Introduction

Degraded character recognition is a very important topic in digital camera based document image processing. For degraded character image, as the degradation level increases, the performance of binarization drops dramatically, which causes big problem for binary-image-based feature extraction. In such case, feature directly extracted from grayscale image has more advantages in keep the shape and structure information of the character.

While many grayscale feature extraction methods have been proposed in recent years[1][2][3][4], they all concentrated on single character recognition. Few paper deals with the performance of the feature on character segmentation. For real text string recognition, recognition-based segmentation is a commonly used strategy[7]. As shown in Figure 1,the first row is the grayscale image of the degraded text string. The second row is the binarization result. Based on connected component analysis on the binary image and the estimated average character width, the dissection step produces two kind of segments: the basic segment (the third row) and the synthesized segment (the fourth row). The basic segment is the un-separable basic unit in image based segmentation. One character image might contains one or more basic segments (for example, left and right structure character image or broken character image). The synthesized segment includes the combination of two or more basic segments under the constraint of the aspect ratio of the segment. During dissection(image based segmentation), no recognition is involved. Therefore one segment might contain one or more characters(touching cases). Also, one character might contain one or more basic and/or synthesized segments. After character dissection, every segment is recognized as some character category. Finally, DP searching is performed to search the segment combination with the minimum total cost. The advantage of recognition-based segmentation is that there is no strict requirements on the dissection step. The recognition is embedded inside the segmentation.

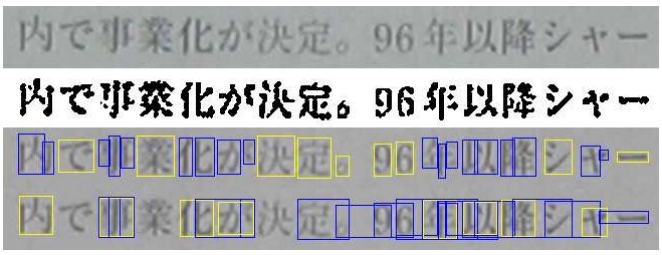


Figure 1: Segmentation of degraded text string.

In the case of degraded text string recognition, more difficulties come out for recognition based segmentation:

1) Broken characters increase the number of segments: a broken character image will be dissected into two or more basic segments. As a result, the number of basic and synthesized segments increases dramatically (the 3rd and 4th row in Figure 1). The correct segments (yellow rectangles) are merged into the incorrect segments (blue rectangles).

2) Difference between character image and non-

character image is diminished by heavy degradation from blurring and low resolution.

It should be noticed that the "recognition" in segmentation is not the same as the "recognition" in single character recognition. The former focuses on how to separate character from non-character. The latter always assumes the input is an character image with unknown category. This situation is similar with the difference between face detection and face recognition.

In this paper, two types of grayscale features derived from dual-eigenspace decomposition are compared. The R-Feature is good at single character recognition. The S-Feature is good at separating character image from non-character image. A combination of the two features can further improve the performance of text string recognition.

2. Grayscale feature extraction and recognition

Many camera based document images contain heavy degradation caused by low resolution, blurring and distortion, which will bring great trouble to binarization. One solution to overcome this problem is to extract the feature directly from the grayscale character image. The grayscale features proposed in this paper are based on eigenspace decomposition [5][6].

2.1 Dual eigenspace decomposition

Dual eigenspace decomposition includes two kind of eigenspace: the unitary eigenspace and the individual eigenspace. The unitary eigenspace serves as the 1st level feature extraction and coarse classification. The individual eigenspace is built on the feature extracted from the unitary eigenspace and is used for the 2nd level feature extraction and fine classification. This coarse to fine recognition structure efficiently improves the processing speed, which makes real time recognition possible.

The unitary eigenspace is built from character images of all categories. suppose a character image with size of $ndim = w * h$ is represented by a vector $x = [x_1, x_2, \ldots, x_{ndim}]^T$ using the raster scanning order. The unitary eigenspace is constructed by Principal Component Analysis (PCA) on the covariance matrix of character template images of all categories:

$$COV = \frac{1}{P_c}\sum_{i=1}^{P}\sum_{j=1}^{N_c}(m_{ij}-m)(m_{ij}-m)^T, \qquad (1)$$

where P is the number of the character categories. N_c is the number of templates for every category. $P_c = P * N_c$ is the number of total character template images in all categories. m is the mean vector for all character template images. m_{ij} is the jth character template image in the ith category. The first n eigenvectors of matrix corresponding to the first largest n eigenvalues spans the unitary eigenspace, $U = [u_1, u_2, \ldots, u_n]^T$.

Since the unitary eigenspace is constructed on the samples of all categories, the discrimination power is not strong enough. In order to further improve the recognition performance, an individual eigenspace is built for every character category using the PCA feature from the unitary eigenspace. The covariance matrix for the ith category is obtained as:

$$\begin{aligned} COV_i &= \frac{1}{M_i}\sum_{k=1}^{M_i}(y_i^{(k)}-c_i)(y_i^{(k)}-c_i)^T, \\ y_i^{(k)} &= U^T(x_i^{(k)}-m) \\ c_i &= \frac{1}{N_c}\sum_{j=1}^{N_c}c_{ij} \qquad i = 1, 2, \ldots, P \end{aligned} \qquad (2)$$

where $y_i^{(k)}$ is the PCA feature of the kth training sample in the ith category. c_{ij} is the PCA feature of the jth character template image in the ith category, m_{ij}. c_i is the mean feature of the ith category. M_i is the number of training samples for the ith category. The first n_i eigenvectors of COV_i corresponding to the first n_i largest eigenvalues, $\tilde{U}_i = [u_1^i, u_2^i, \ldots, u_{n_i}^i]^T$, spans the individual eigenspace for the ith category.

2.2 Character image normalization

In order to remove the influence of degradation, precise image registration is first performed on every segment image[4]. The registered segment image has uniform size and the brightness of the image pixel value is compensated.

Figure 2 shows the result of the registration of some segments in Figure 1. The top row is the images of the correct segments. The second row shows the images of the basic segments. The third and the fourth row are the images of synthesized segments.

In order to further improve the recognition performance against degradation, definite canonicalization [8] is used to filter the mean value of the image:

$$c = (1/\sqrt{n}, ..., 1/\sqrt{n}), \qquad (3)$$

$$x' = x - (c \cdot x)c \qquad (4)$$

Finally, the energy of normalized vector, x', is regulated into unit length. Intuitively, the normalization step transfer the character image from inside a hypercube with lattice length of 255 into the surface of a hypersphere.

Figure 2: Segment image registration.

2.3 R-Feature extraction and recognition

The unitary eigenspace can be regarded as a transformation from the normalized image domain to the frequence domain. The individual subspace is built upon the frequence space. The R-Feature is extracted in the frequence domain.

First, the PCA feauture is extracted from the normalized character image as in Equation (5) by the unitary eigenspace, U:

$$y = U^T(x' - m). \tag{5}$$

Then, coarse classification is performed by matching the feature, y, with all template feature, c_{ij}. Assuming the first N_{cand} candidate character categories are selected by minimum Euclidean distance, the R-Feature is the reconstructed feature for every category as in Equation (6) by the individual eigenspace.

$$\begin{aligned} \eta_k &= \tilde{U}_k^T(y - c_k), \\ \hat{y}_k &= \tilde{U}_k^T \eta_k + c_k \end{aligned} \tag{6}$$

During the recognition phase, the recognition distance of the R-Feature is taken as the norm of the difference of the PCA feature and its reconstruction:

$$d_k^R = \| y - \hat{y}_k \| \tag{7}$$

The classification is accomplished by sorting the N_{cand} character categories according to the minimization of their recognition distance.

2.4 S-Feature extraction and recognition

Different with the R-Feature, the S-Feature is extracted from the original image domain. First, the R-Feature is recovered back into the normalized image domain:

$$\hat{x}_k = U \cdot \hat{y}_k + m \tag{8}$$

Then the recovered vector is scaled into the original image space:

$$\begin{aligned} \ddot{x}_k(i) &= 255 * (\hat{x}_k(i) - m_1)/(m_2 - m_1) \\ m_1 &= \min\{\hat{x}_k(i)\}, \qquad i = 1, 2, \ldots, n \\ m_2 &= \max\{\hat{x}_k(i)\}, \qquad i = 1, 2, \ldots, n \end{aligned} \tag{9}$$

The physical meaning of the S-Feature, $\ddot{x}_k$, is very clear: it is the restored version of the input segment image by the two-fold eigenspace.

Equation (9) is the reverse transformation of the normalization step in section 2.2, which leverages the basic nature of the registered character image: dark background with pixel value near 0, bright character stroke with pixel value near 255.

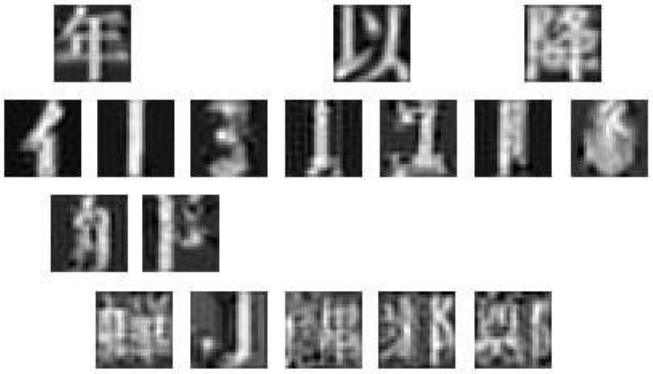


Figure 3: S-Feature of Figure 2

Figure 3 shows the corresponding S-Feature image of the segments displayed in Figure 2. By comparing Figure 2 with Figure 3, we can see that for character image, the restored image can well preserve the structure information. However, for non-character image like the segments in the 2nd, 3rd, and 4th row in Figure 2, much more noise are introduced by the reverse transformation.

This phenomenon is because the unitary eigenspace and the individual eigenspace are all built using character images only. The noise in the reconstruction caused by a degraded version of a character image is much smaller than the noise caused by a non-character image. If the input is a non-character image, the individual eigenspace will choose a most similar character category and calculate the reconstructed R-Feature. Then, the noise introduced by the R-Feature is transferred by the unitary eigenspace back into the image space. The bigger the noise is, the more uneven happens for the pixel value among the background pixels and the pixel value among the stroke pixels. Finally, these unevenness is enhanced by the image reconstruction operation in Equation (9). This special property of the S-Feature makes it very suitable for the segmentation task.

In order to better calculate the difference between two character structure, angle based distance is used instead of Euclidean distance in calculating the recognition distance of the S-Feature:

$$d_k^S = 1 - \frac{x \bullet \ddot{x}_k}{\| x \| \| \ddot{x}_k \|} \quad (10)$$

3. Performance evaluation

For the R-Feature and the S-Feature defined in the previous section, the performance of recognition and segmentation are evaluated separately.

Two testing sets are used here. Testing set 1 is used to evaluate the performance of single recognition, which includes Level-1 Japanese Kanji characters with 19 fonts. These Kanji characters are first typed into slides and the slides are projected onto a screen. Then digital camera is used to capture the character image in three different distances, S1, S2 and S3. Total number of Kanji characters in every distance is 56,335.

Testing set 2 is used to evaluate the performance on segmentation , which includes degraded text string images captured from "Nikkei Business Magazine" with a Canon PowerShot A80 digital camera under maximum 4 MegaPixels resolution. Total number of text string is 292. Total number of characters is 4491. The correct segmentation result (groundtruth) is labelled manually for every image in testing set 2.

3.1 Performance evaluation for single character recognition

For comparison, PCA feature extracted from unitary eigenspace is also evaluated for single character recognition along with the R-Feature and the S-Feature. Table 1 lists the recognition rate of the R-Feature, S-Feauture and PCA feature on testing set 1.

Table 1: Single character recognition rate of R-Feature, S-Feature and PCA feature(%)

Dataset	S1	S2	S3
R-Feature	**92.80**	**97.46**	**98.51**
R-Feature*	91.47	96.26	97.09
R-Feature**	92.63	97.39	98.40
S-Feature	90.88	96.31	97.82
PCA feature	83.06	89.77	91.97

"R-Feature*" is the result using dual-eigenspace trained by registered character images. "R-Feature**" is the result using dual-eigenspace trained by definite canonicalized registered character images. "R-Feature" is the result using dual-eigenspace trained by the images after normalization operation defined in section 2.2.

The comparison between R-Feature, R-Feature*, and R-Feature** shows the effectiveness of different normalization methods. What's more, we can see from Table 1 that for all 3 distance, $S1$, $S2$, and $S3$, the Dual-eigenspace based features are better than PCA feature. That is because the unitary eigenspace is based on all character categories, the discrimination power is not strong.

Finally, the performance of the R-Feature is better than that of the S-Feature in single character recognition. That means for character image, recovery from the common unitary eigenspace "blurs" the difference between similar characters. Hence, the S-Feature is not a suitable feature for single character recognition.

3.2 Performance evaluation for segmentation

3.2.1 Segmentation error rate

The performance of segmentation is represented by the segmentation error rate. For every feature, recognition based segmentation is evaluated on testing set 2. The segmentation result is compared with the groundtruth. The segmentation error rate is defined as:

$$Err_{seg} = \frac{num.\ of\ mismatched\ segments * 100}{num.\ of\ total\ groundtruth\ segments} \quad (11)$$

If the region of a segment cannot overlap completely with any of the groundtruth, we call that segment a "mismatched segment". Therefore, the segmentation error rate is the complement of the recall rate of the segmentation. Table 2 lists the segmentation error rate for the R-Feature, S-Feature and PCA feature. The number of groundtruth segments is 4523.

Table 2: Segmentation performance of R-Feature, S-Feature and PCA feature.

	$mismatch\ num.$	Err_{seg}
$PCAfeature*$	2146	47.45%
$PCAfeature$	500	11.05%
$R-Feature$	121	2.68%
$S-Feature*$	130	2.87%
$S-Feature$	**23**	**0.51%**

The "PCA feature*" and "S-Feature*" in Table 2 are the result without the image reconstruction step defined in Equation 9. Two conclusions can be drawn from the segmentation experiments:

(1) The individual eigenspace is helpful in segmentation.

(2) The image reconstruction step is very effective. Simply transferring the R-Feature back to n-dimensional space by the unitary eigenspace cannot necessarily improve the

segmentation performance. The error rate of both PCA feature and the S-Feature all dramatically decreases by the image reconstruction operation.

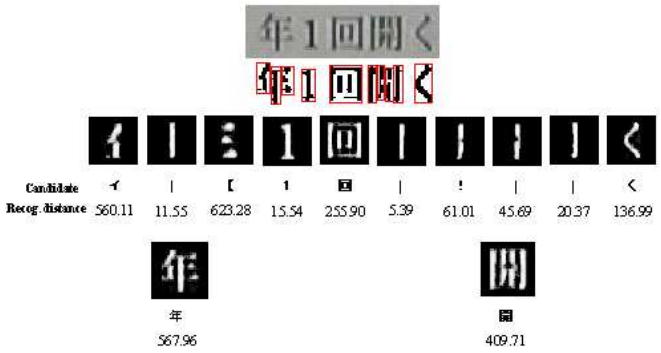


Figure 4: Segmentation result using R-Feature

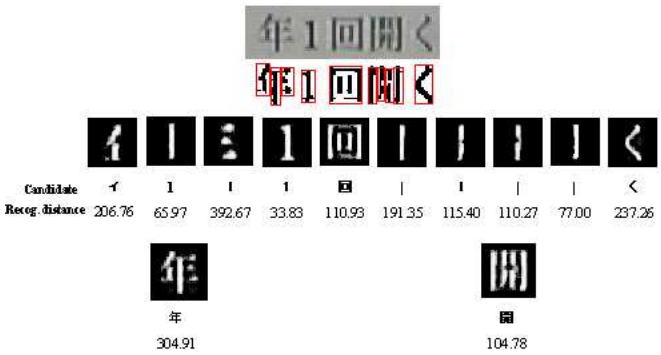


Figure 5: Segmentation result using R-Feature

Figure 4 and Figure 5 show the segmentation results using R-Feature and S-Feature respectively. The first row is the original degraded grayscale text line image. The second row is the binarization result that is used for coarse segmentation. The third row is the normalized image for the basic segments. The fourth row is the normalized image for two synthesized segments. Below the normalized image is the recognition result and the corresponding recognition distance. Notice the recognition result in two figures are not exactly the same because different feature is used. Compared with R-Feature, the recognition distance of S-Feature is more reasonable. The fourth character in Figure 4 will be mis-segmented into 4 components. But in Figure 5, the character can be correctly segmented because the recognition distance is far less than the sum of its four components.

3.2.2 Relative recognition distance

The reason of the success of the S-Feature in segmentation is further analyzed by the distribution of the relative recognition distance (RRD). In order to calculate the RRD for a segment image, the segments generated from a text string are first classified into 3 categories: correct-segment, under-segment, and over-segment.

The correct-segment is a basic segment or synthesized segment that can overlap with a groundtruth segment perfectly. The segment images in the first row of Figure 2 are correct-segments.

The under-segment is a basic segment or synthesized segment that is embedded inside a groundtruth segment. The segment images in the second and the third row in Figure 2 are under-segments.

The over-segment is a synthesiz segment that intersects two or more groundtruth segments. The segments in the fourth row in Figure 2 are over-segments.

The original recognition distance obtained by Equation 7 and Equation 10 is very sensitive to the degradation level of a particular text image. Hence for every segment, the RRD is defined according to the relationship between the position of the segment and the position of the interfered correct-segment:

$$RRD_c = 1 \tag{12}$$

$$RRD_u = \frac{d(SEG_u)}{d(SEG_c)},\ SEG_u \subset SEG_c \tag{13}$$

$$RRD_o = \frac{d(SEG_o)}{\sum_{j=1}^{m} d(SEG_c^j)/m},\quad SEG_o \bigcap SEG_c^j \neq \emptyset. \tag{14}$$

Where SEG is the image region of the segment. $d()$ stands for the recognition distance obtained by the R-Feature or the S-Feature. m is the number of correct-segments that intersect with the over-segment.

Figure 6 shows the histogram of RRD_u of the R-Feature and the S-Feature. If a correct-segment can be decomposed into two or more under-segments, and the summed RRD value of the under-segments is less than 1, segmentation error will occur. We can see from the figure that the number of small-value RRD for the S-Feature is much less than that of the R-Feature. That is very helpful for the segmentation task. Figure 7 shows the histogram of RRD_o of the R-Feature and the S-Feature. Again, the S-Feature has more large value RRD_o than that of the R-Feature.

The fact that the S-Feature has less small-RRD-valued segments than the R-Feature proves that the S-Feature is a more suitable feature for recognition based segmentation.

4. Feature combination for degraded text string recognition

The analysis in the previous section shows that the S-Feature and the R-Feature are complementary: the S-Feature is better in recognition-based segmentation. The

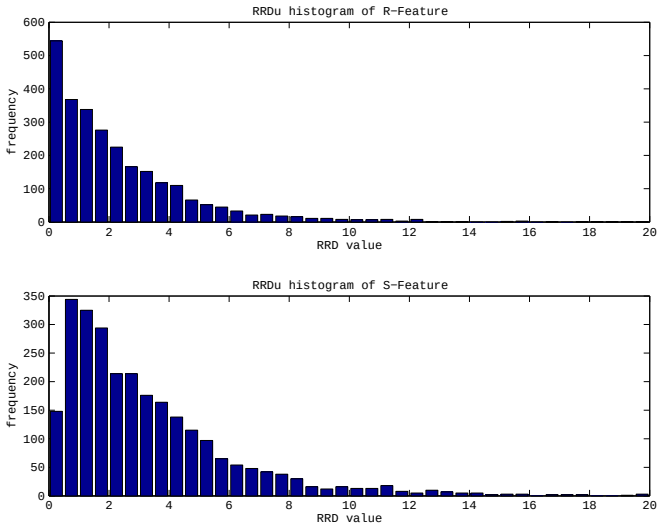


Figure 6: RRD_u histogram of R-Feature (top) and S-Feature (bottom)

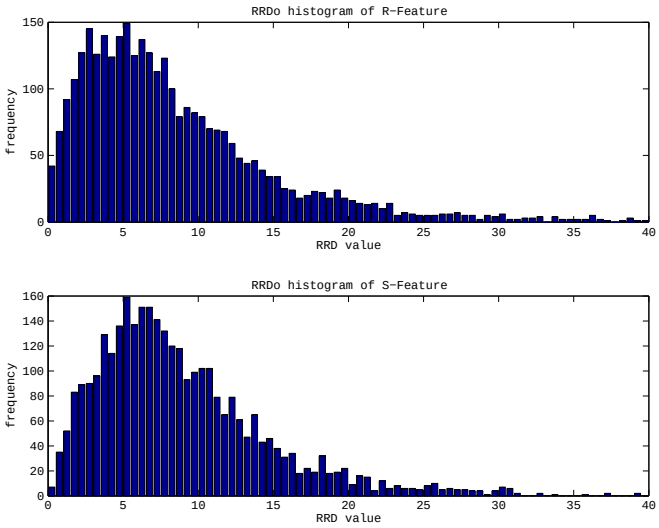


Figure 7: RRD_o histogram of R-Feature (top) and S-Feature (bottom)

R-Feature is better in single character recognition. Thus, a combination strategy is proposed for degraded text string recognition.

(1) Characters in the text string is segmented using the S-Feature by recognition-based segmentation.

(2) The recognition result of every segmented character images is refined by the R-Feature.

Notice that both R-Feature and S-Feature are all derived from the same dual-eigenspace decomposition. Therefore the computation time can be effectively reduced.

Performance evaluation of feature combination is conducted using testing set 2 and testing set 3. The images in testing set 3 include 90 text blocks with variant character size, the minimum size of character is as small as 10*10 pixels. Total number of characters in testing set II is 10253.

Table 3: Recognition of feature combination.(%)

Dataset	R-Feature	Feature combination
testing set 2	93.94	**95.79**
testing set 3	76.82	**80.88**

As shown in Table 3, by combining the S-Feature with the R-Feature, the overall recognition performance can be improved effectively.

5. Conclusions

The impact of grayscale feature on text string segmentation is discussed in this paper. Two kinds of features are proposed for degraded text string recognition: the S-Feature is used for text string segmentation, the R-Feature is used to recognized segmented individual characters. Experiments prove the effectiveness of the feature combination.

References

[1] Wang X. W., Ding X. Q., and Liu C. S. Optimized Gabor filter based feature extraction for character recognition. *Proc. of ICPR*, pp. 223-226, 2002.

[2] Yoshimura, H., Etoh, M., Kondo, K., et al. Gray-scale character recognition by gabor jets projection. *Proc. of ICPR* pp.335-338, 2000

[3] Wang, L., Pavlidis, T. Direct Gray-Scale Extraction of Features for Character Recognition. *IEEE trans. Pattern Analysis and Machine Intelligence* 15(10) pp.1053-1067, 1993

[4] Sun, J., Hotta, Y., Katsuyama, Y., Naoi, S. Low resolution character recognition by dual eigenspace and synthetic degraded patterns. *1st ACM workshop on Hardcopy Document Processing* pp.15-22, 2004.

[5] Duda, R. O., Hart, P. E., Stork, D. G. *Pattern classification, second edition.* A Wiley-Interscience Publication John Wiley & Sons, Inc. pp.568 569, 2001.

[6] Zhang, D., Peng, H., Zhou, J., Sankar, K. P. A novel face recognition system using hybrid neural and dual eigenspace methods. *IEEE trans. System, Man and Cybernetics - part A* 32(6) pp.787-792, 2002

[7] Casey, R. G., Lecolinet, E. A Survey of Methods and Strategies in Character Segmentation. *IEEE trans. Pattern Analysis and Machine Intelligence* 18(7) pp.690-706, 1996

[8] Iijima, T. Theory of pattern recognition. Series of basic information technology 6. *Morikita Publishing Company Ltd.,* 1989

Recognition of low-resolution characters by a generative learning method

Hiroyuki Ishida, Shinsuke Yanadume, Tomokazu Takahashi,
Ichiro Ide, Yoshito Mekada and Hiroshi Murase
Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Aichi, 464-8601, Japan
{hishi, yanadume, ttakahashi, ide, mekada, murase}@murase.m.is.nagoya-u.ac.jp

Abstract

Using appropriate training data is necessary to robustly recognize low-resolution characters by the subspace method. Former learning methods used characters actually captured by a camera, which required the collection of characters of all categories in various conditions. In this paper, we propose a new learning method that generates training data by a point spread function estimated beforehand by captured images. This method is efficient, since it eliminates collection of the training data. We confirmed its usefulness by experiments.

1. Introduction

Technologies to recognize low-resolution characters have gained attention in recent years for their potential applications to portable digital cameras. Recognizing documents with camera-equipped cellular phones is especially of practical concern [1]. Various attempts have been carried out on the recognition of printed characters on documents [2], [3], [4]. However, few studies have focused on very low-resolution characters captured by portable digital cameras. Even with the improvements of digital cameras, the characters in a captured image are still often small, when the target documents contain many characters.

The purpose of this work is to devise a new method for the ef cient recognition of low-resolution characters using the subspace method [5], [6]. Generally, training data are collected from actual captured images. Since it is dif cult and unrealistic to collect character images of all categories under various conditions, we propose a generative learning method in which training data are generated automatically from original character images. Since our goal is to recognize low-resolution characters, training data should be generated in accordance with actual degradation. The outline of this generative learning method is as follows: First, we estimate a point spread function (PSF) [7] of a camera from captured images. Next, we generate degraded training data by applying PSF to the original character images. This method is ef cient, since it eliminates the collection of training data of all categories from captured images. This method is also suitable for the recognition of low-resolution characters compared with a simple learning method that only uses original character images without taking the actual degradation into consideration.

This paper is organized as follows: Section 2 describes the degradation process of the captured characters and our approach to simulate the degradation process. Section 3 describes the main idea of the generative learning method. Section 4 describes the recognition steps using the subspace method. Section 5 demonstrates the experimental results. Section 6 concludes the paper.

2. Degradation process

Understanding the actual degradation process of a captured character image is crucial for a generative learning method. A target character image is small and in low quality, which makes it dif cult to recognize.

We divide the causes of character degradation into two factors: (1) Optical blur is caused by a process where the characters on a target document are projected onto CCD sensors through a camera lens, and (2) resolution decline is caused through the sampling process, where the character image is turned into a low-resolution digital image. Besides, captured character images vary with every frame even if captured from the same original image, since the light quantities of each CCD sensor shift due to slight camera vibration. Even such slight changes cannot be ignored for low-resolution character recognition.

We propose two generation models based on these factors: optical blur and resolution decline. The rst generation model copes only with the low-resolution factor. In this model we generate training data by reducing the resolution of the original character images. Font data on a computer is used as the original image, based on the assumption

(a) original image (b) optically blurred image (c) low-resolution image

Figure 1. Degradation factors.

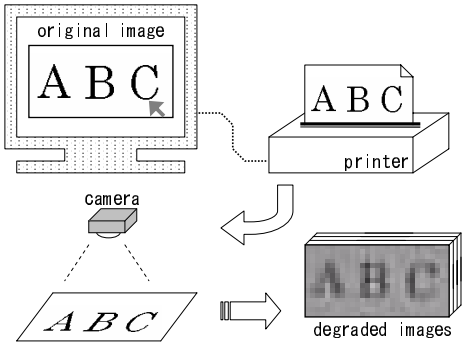


Figure 2. Flow of preparation for PSF estimation.

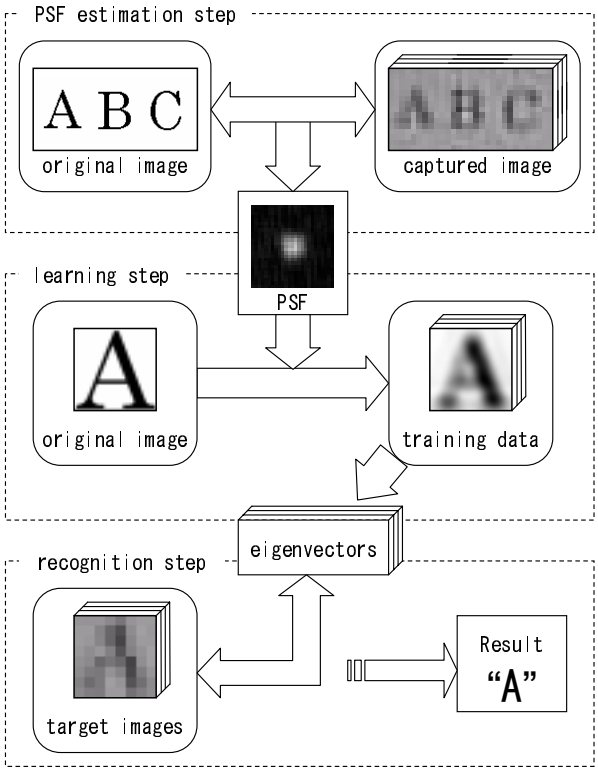


Figure 3. Flow of generative learning method based on PSF estimation.

that the characters to be recognized are printed. The second generation model copes with both degradation factors (low-resolution and optical blur). We generate training data for this second model by applying PSF estimated from actual captured images (Section 3). We use the same camera for both PSF estimation and recognition, and that enables us to simulate degradation features peculiar to the camera. In later sections, we discuss the effectiveness of these two generation models by experiments, and also compare them with a non-generative learning method in which none of the degraded factors are included.

3. Generative learning

In this paper, we propose a learning method that learns from arti cially degraded training data. In the following sections, we compare two models for generative learning. The rst generates training data only by reducing the resolution of the original images. For convenience, we call this method "Generative learning method (type-A)". The second generates training data by applying estimated PSF together with the reduction of the resolution. Similarly, we call this "Generative learning method (type-B)", whose o w is shown in Figure 2 and 3. In this section we focus on generative learning method (type-B).

3.1. Preparation for the PSF estimation

First we need to obtain images for PSF estimation. For the original image, we printed some characters and obtained degraded images for PSF estimation by capturing the printed gure with a camera (Figure 2). We need to capture a certain quantity of images for noise reduction.

Although PSF itself is independent from the original image, an appropriate PSF cannot be acquired if it is composed only of monotonous structural elements; an image composed of various structural elements is more appropriate as the original image for the PSF estimation.

3.2. PSF estimation

PSF is estimated from the original image and a number of degraded images captured by a camera. Image degradation is represented as

$$g(x, y) = f(x, y) * h(x, y) + n(x, y), \qquad (1)$$

where f is the original image, g is a degraded image, h is PSF, and n is the noise function. This equation indicates that a degraded image is generated by the convolution

of the original image and PSF [8]. To obtain h, we apply a two-dimensional fourier transformation to Equation 1 as follows:

$$H(u,v) = \frac{G(u,v)}{F(u,v)} - \frac{N(u,v)}{F(u,v)}, \tag{2}$$

where noise component N is unknown. Since the captured images contain a lot of noise, it is not possible to obtain an appropriate PSF from a single image. Thus, this method averages $H(u,v)$ calculated from variously degraded images to restrain noise [9]. Assuming that we use k degraded images $G_i(u,v)$ $(i = 1, \cdots, k)$, averaged $\hat{H}(u,v)$ can be calculated from multiple $H_i(u,v)$ as

$$\begin{aligned} \hat{H}(u,v) &= \frac{1}{k}\sum_{i=1}^{k} H_i(u,v) \\ &= \frac{1}{k}\sum_{i=1}^{k} \frac{G_i(u,v)}{F(u,v)} - \frac{1}{k}\sum_{i=1}^{k} \frac{N_i(u,v)}{F(u,v)}. \end{aligned} \tag{3}$$

Since we consider that no relation exists among the noise components of each image, consequently,

$$\lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} \frac{N_i(u,v)}{F(u,v)} = 0. \tag{4}$$

Thus, when k is large enough, $\hat{H}(u,v)$ can be approximated as

$$\hat{H}(u,v) = \frac{1}{F(u,v)}\frac{1}{k}\sum_{i=1}^{k} G_i(u,v). \tag{5}$$

PSF is obtained by the inverse Fourier transformation of $\hat{H}(u,v)$.

Figures 4, 5, and 6 show the estimated PSF of each camera (digital video camera, digital camera, and camera equipped on cellular phone).

3.3. Generation of training data

Training data is generated with estimated PSF with various degrees of degradation. Here we introduce degradation parameter d to designate the degree of degradation. When $d = 0$, the generated image is equivalent to the original image, and when $d = 1$, the generated image is equivalent to the convolution of the original image and estimated PSF. We apply a PSF lter the size of $(R_h + 1) \times (R_h + 1)$ (See Figure 7) that transforms the original image into a degraded image. Each pixel of the training data is calculated using the PSF lter expanded in proportion to degradation parameter d as:

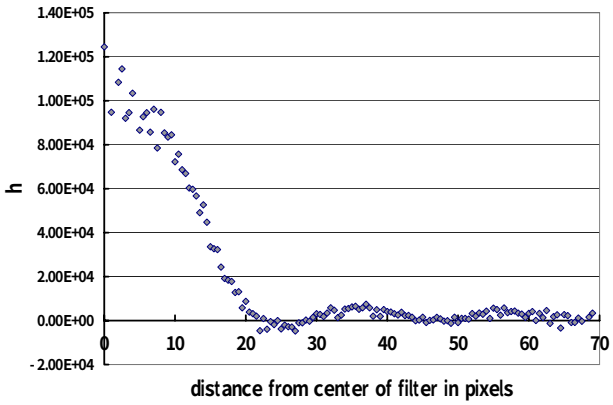


Figure 4. PSF of digital video camera.

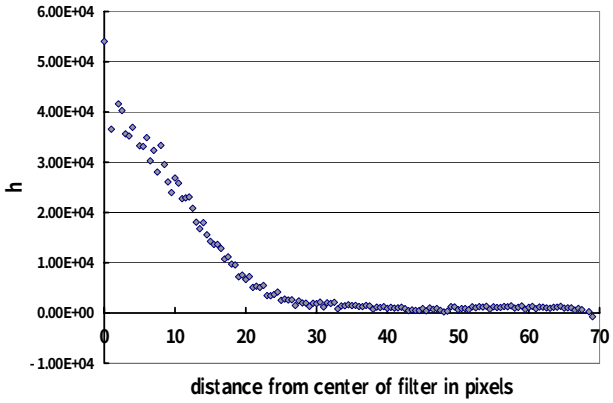


Figure 5. PSF of digital camera.

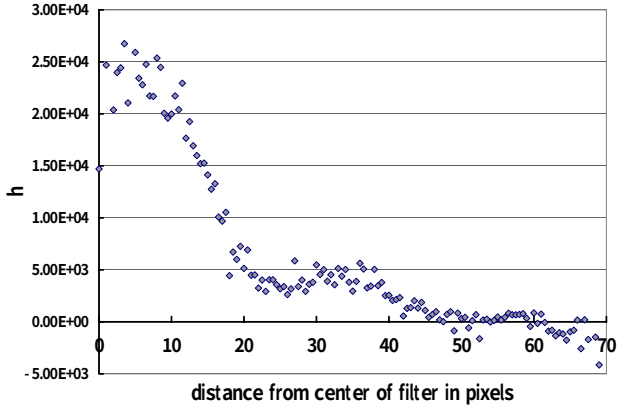


Figure 6. PSF of camera equipped in cellular phone.

$h(-\frac{R_h}{2}, -\frac{R_h}{2})$	$\cdots$	$h(0, -\frac{R_h}{2})$	$\cdots$	$h(\frac{R_h}{2}, -\frac{R_h}{2})$
$\vdots$		$\vdots$		$\vdots$
$h(-\frac{R_h}{2}, 0)$	$\cdots$	$h(0, 0)$	$\cdots$	$h(\frac{R_h}{2}, 0)$
$\vdots$		$\vdots$		$\vdots$
$h(-\frac{R_h}{2}, \frac{R_h}{2})$	$\cdots$	$h(0, \frac{R_h}{2})$	$\cdots$	$h(\frac{R_h}{2}, \frac{R_h}{2})$

Figure 7. PSF filter.

$$g(p,q) = \sum_{\substack{i=-\frac{R_h}{2},\cdots,\frac{R_h}{2} \\ j=-\frac{R_h}{2},\cdots,\frac{R_h}{2}}} h(i,j) f\Big(\frac{R_f}{R_g}(p-di), \frac{R_f}{R_g}(q-dj)\Big), \tag{6}$$

where

f : Original character image
g : Generated training image
h : Point spread function
R_f : Size of the original character image
R_g : Size of the training image
R_h : Size of the PSF lter
d : Degradation parameter.

3.4. Construction of a subspace

We construct a subspace that approximates the patterns of characters. Now we have training data generated in accordance with the proposed generation model. All the training data are converted to a unit vector. The vectorized image is represented as $x_{m,n}(m = 1, \cdots, M,\ n = 1, \cdots, N)$, where M denotes the number of character categories and N denotes the number of generated images per category. Autocorrelation matrix $\boldsymbol{X}_m$ is represented as

$$\boldsymbol{X}_m = \Big[\boldsymbol{x}_{m,1} \ \cdots \ \boldsymbol{x}_{m,N}\Big]\Big[\boldsymbol{x}_{m,1} \ \cdots \ \boldsymbol{x}_{m,N}\Big]^t. \tag{7}$$

Next we calculate the eigenvalues and the eigenvectors of this matrix $\boldsymbol{X}_m$. The eigenvectors are sorted in order of the magnitude of their corresponding eigenvalues, and we use the largest L eigenvectors $\boldsymbol{u}_{m,l}$ $(l = 1, \cdots, L)$, where generally $L \leq N$. Some examples of the eigenvectors are illustrated in Figure 8.

Figure 8. Top three eigenvectors.

4. Character recognition by the Subspace method

The subspace method identi es target characters by comparing similarity between a character and subspaces for each category. Several studies have been conducted on the recognition of low-resolution characters. The recognition method, which uses multiple frames of video data, is presented by Yanadume et al. [12]. In this section, we describe this method that recognizes characters captured by a digital video camera. The target character image is classi ed to a category that marks the largest similarity with the image. Similarity is calculated by projecting target images onto the subspace for each category. According to Yanadume et al., integrating information from multi-frame images drastically improves recognition accuracy. Since low-resolution characters are dif cu lt to recognize by themselves, various image restoration methods have been proposed [8] – [11]. In Yanadume et al.'s method, however, the target image does not need to be restored since the information on the target characters is obtained from multi-frame images. Given F frames of the same character, and if the j-th vectorized target image is represented as $\boldsymbol{y}_j$, the similarity between the target image and category m is de ne d as

$$s_m = \sum_{j=1}^{F} \sum_{l=1}^{L} (\boldsymbol{u}_{m,l} \cdot \boldsymbol{y}_j)^2, \tag{8}$$

where $\boldsymbol{u}_{m,l}$ $(l = 1, \cdots, L)$ denotes the eigenvector of category m. After calculating s_m for all M categories, the category that marks the largest similarity is accepted.

We employ this method in our work in the recognition step.

5. Experiments

5.1. Learning methods

We compared the effectiveness of the proposed generative learning method with a method that only learns from the original character images (non-generative learning method). The details of these methods are as follows:

1. Non-generative learning method

(a) Training data for non-generative learning method.

(b) Training data for generative learning method (type-A).

(c) Training data for generative learning method (type-B).

Figure 9. Training data for each learning method.

The training data were the original character images normalized in size to 32×32 pixels. We used one training data per category. Unlike the other learning methods, we evaluated the similarity between the training data and a target image in recognition step; the similarity was de ned as the sum of squared inner products between the two vectorized images.

2. Generative learning method (type-A)

   The training data were low-resolution characters (8×8 to 32×32 pixels) generated from the original character images by reducing resolution without using PSF. We used 25 training data per category. The size of the training images was normalized to 32×32 pixels. We calculated the eigenvectors from the training data and used them with the ten highest ranks for recognition.

3. Generative learning method (type-B)

   The training data were generated with the estimated PSF, as described in Section 3. First, we captured the degraded images for PSF estimation with a digital video camera (DV camera) and a digital camera (DC) at a distance of 70 cm and with a camera-equipped cellular phone (phone camera) at a distance of 20 cm. Second, we estimated the PSF of each camera from these captured images. Third, we generated training data. The size of the original character image was 128×128 pixels, and the size of the generated image was set to 32×32 pixels. We generated the training data by changing degradation parameter d from 0.05 to 1.00 by 20 steps. We calculated the eigenvectors from the training data and used them with the ten highest ranks for recognition.

Table 1. Size of characters (DV camera, DC).

distance	22 cm	35 cm	50 cm	60 cm	70 cm
DV	16×16	10×10	7×7	6×6	5×5
DC	17×17	13×13	10×10	9×9	8×8

Table 2. Size of characters (Phone camera).

distance	20 cm	32 cm
Phone	7×7	5×5

Several samples of training data for each learning method are shown in Figure 8.

5.2. Test data

We captured images containing 62 characters (A - Z, a - z, 0 - 9) with a DV camera, a DC, and a phone camera. Each character was printed with an alphanumeric 'Century' font. The original size of the character on the target documents was approximately 1×1 cm. We automatically segmented the characters from the captured images. The segmented area was the smallest square that included the whole character; Tables 1 and 2 show the relations between camera distance and the average size of the test data. After segmentation, we normalized the size of all characters to 32×32 pixels. Figures 10, 11 and 12 shows some examples of the test data.

5.3. Experimental results

Figures 13, 14, and 15 show the experimental results for each camera. The test data consisted of 62 letters: uppercase characters (A - Z), lowercase characters (a - z), and numbers (0 - 9). We calculated the recognition rate of these 62 letters using ten successive frames in the video and averaged the recognition rates obtained from 50 video data, as described in Section 4.

5.4. Discussion

Experimental results showed the effectiveness of the generative learning method for low-resolution characters. Generative learning methods (types A and B) exhibited high recognition rates compared with the non-generative learning method. This was peculiar for extremely low-resolution characters whose size was below 10×10 pixels, since the eigenvectors calculated from the degraded training data coped with the degradation. This result shows the effectiveness of generating arti cially degraded training data.

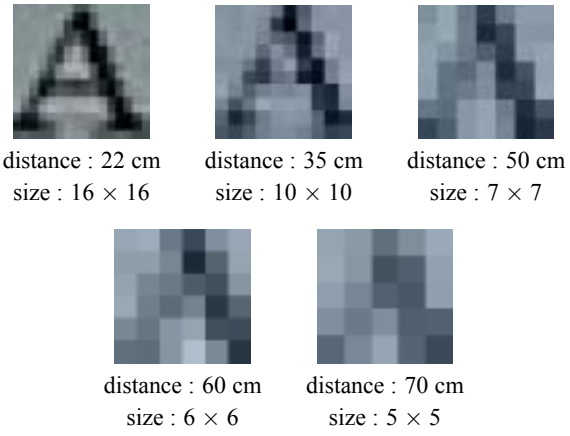


Figure 10. Test data captured with DV camera.

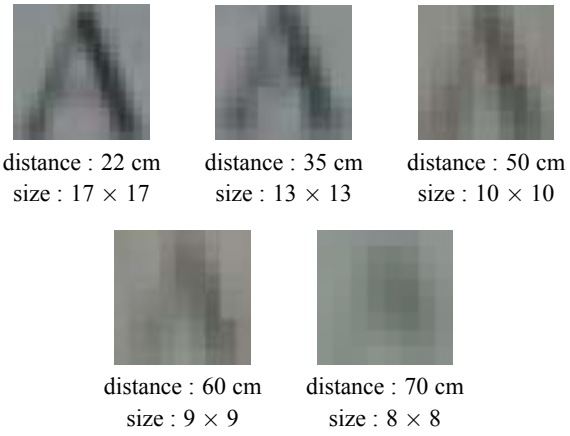


Figure 11. Test data captured with DC.

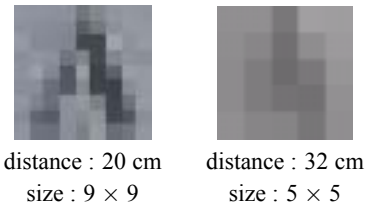


Figure 12. Test data captured with phone camera.

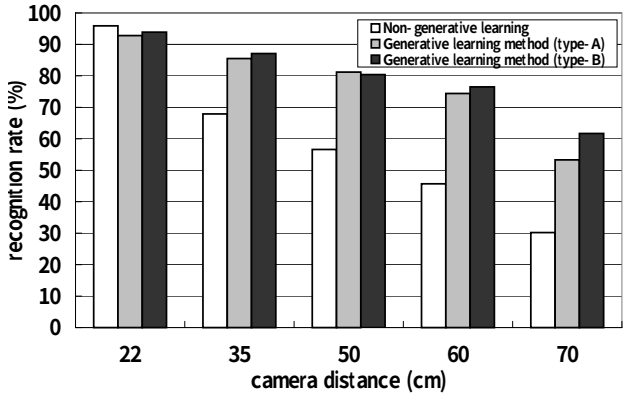


Figure 13. Recognition results (DV camera).

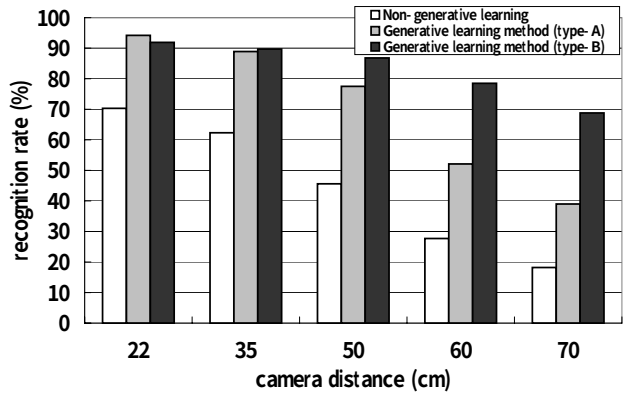


Figure 14. Recognition results (DC).

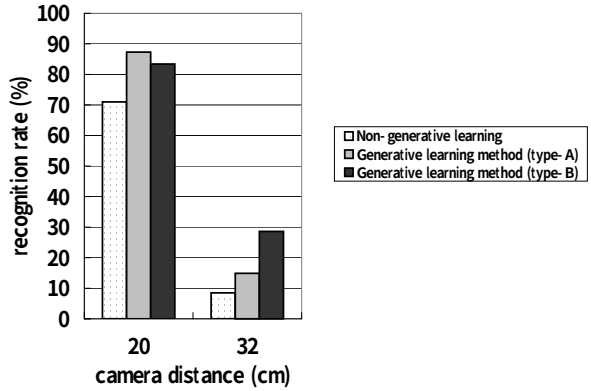


Figure 15. Recognition results (Phone camera).

The recognition rates of generative learning method types A and B were almost comparable where the size of the target characters were over 10×10 pixels. But the generative learning method (type-B) marked higher recognition rates for low-resolution characters. Results of experiments using a digital camera indicated the comparative robustness of generative learning method (type-B) while the recognition rates of other methods suddenly dropped in proportion to camera distance, indicating that generative learning method (type-B) is robust to the in uen ce of optical blur. The shape of PSF estimated by a digital camera (Fig. 5) shows from its smooth waveshape that images captured by a digital camera are affected by optical blur. These experimental results showed that a generative learning method using estimated PSF is suitable for severely degraded and low-resolution characters.

6. Conclusion

In this paper, we proposed a learning method for the ef - cient recognition of low-quality characters. We proposed a generative learning method that arti cially generates training data instead of collecting them from actual captured images. We applied a generative learning method with PSF estimated from captured images and examined the effectiveness of the method by experiments with three types of cameras. A generative learning method based on PSF proved to be ef cient for our purposes.

Acknowledgment

Parts of this research were supported by the Grant-In-Aid for Scienti c Research (16300054) and the 21st century COE program from the Ministry of Education, Culture, Sports, Science and Technology.

References

[1] D. Doermann, J. Liang and H. Li, Progress in Camera-Based Document Image Analysis. *Proc., 5th International Conference on Document Analysis and Recognition* : 606–616, August, 2003.

[2] V. Govindan and A. Shivaprasad, Character Recognition – A Review. *Pattern Recognition*, 23(7): 671–683, July 1990.

[3] S. Mori, K. Yamamoto, and M. Yasuda, Research on Machine Recognition of Handprinted Characters. *IEEE Trans. PAMI*, 6(4): 386–405, July 1984.

[4] S. Kahan, T. Pavilidis, and H. Baird, On the Recognition of Printed Characters of Any Font and Size. *IEEE Trans. PAMI*, 9(2): 274–288, March 1987.

[5] E. Oja, *Subspace Methods of Pattern Recognition*. Research Studies, Hertfordshire, UK, 1983.

[6] H. Murase, H. Kimura, M. Yoshimura, and Y. Miyake, An Improvement of the Auto-correlation Matrix in the Pattern Matching Method and its Application to Handprinted "HIRAGANA" Recognition (in Japanese). *IEICE Trans.*, J64-D(3): 276–283, March 1981.

[7] H. Andrew and B. Hunt, *Digital Image Restoration*. Prentice-Hall, Englewood Cliffs, N.J., 1977.

[8] S. Hashimoto and H. Saito, Restoration of Shift Variant Blurred Image Estimating the Parameter Distribution of Point Spread Function. *Systems and Computers in Japan*, 26(1): 62–72, January 1995.

[9] N. Tsunashima and M. Nakajima, Estimation of Point Spread Function Using Compound Method and Restoration of Blurred Images (in Japanese). *IEICE Trans.*, J81-D-II(11): 2688–2692, November 1998.

[10] J. Hobby and H. Baird, Degraded Character Image Restoration. *Proc., 5th UNLV Symp. on Document Analysis & Information Retrieval*, Las Vegas (USA), April 1996.

[11] H. Li and D. Doermann, Text Enhancement in Digital Video using Multiple Frame Integration. *Proc. 7th ACM International Conference on Multimedia* : 19–22, November, 1999.

[12] S. Yanadume, Y. Mekada, I. Ide, and H. Murase, Recognition of Very Low-resolution Characters from Motion Images. *Proc. PCM2004, Lecture Notes on Computer Science Springer-Verlag*, 3331: 247–254, December 2004.

Using Adaboost to Detect and Segment Characters from Natural Scenes

Kaihua Zhu Feihu Qi
Renjie Jiang Li Xu
Shanghai Jiao Tong University

Masatoshi Kimachi Yue Wu
Tomoyoshi Aizawa
Omron Corporation, JAPAN

Abstract

We present a robust connected-component (CC) based method for automatic detection and segmentation of text in real-scene images. This technique can be applied in robot vision, sign recognition, meeting processing and video indexing. First, a non-linear Niblack method (NLNiblack) is proposed to decompose the image into candidate CCs. Then, we feed all these CCs into a cascade of classifiers trained by Adaboost algorithm. Each classifier in the cascade responds to one feature of the CC. We propose 12 novel features which are insensitive to noise, scale, text orientation and text language. The classifier cascade allows non-text CCs of the image to be quickly discarded while spending more computation on promising text-like CCs. The CCs passing through the cascade are considered as text components and are used to form the segmentation result. We have built a prototype system and the experimental results prove the effectiveness and efficiency of the proposed method.

1. Introduction

Text detection and segmentation from a natural scene is very useful in many applications. With the increasing availability of high performance, low priced, portable digital imaging devices, the application of the scene text recognition is rapidly expanding [1]. By using cameras attached to cellular phones, PDAs, or standalone digital cameras, we can easily capture the text occurrences around us, such as, street signs, advertisements, traffic warnings or restaurant menus. Automatically recognition, translation or enunciation of these texts will be of great help for foreign travelers, visually impaired people and computer programs which perform the video indexing or meeting processing, etc. [1]

Fully automatic text extraction from images, especially from scene images, has always been a challenging problem. The difficulties underlie in variations of scene text in terms of character font, size, orientation, texture, language and color, as well as complex background, uneven illumination, shadows and noise of images (Fig. 1 shows one example). In addition, a high speed of processing is usually desired.

There are growing works focusing on the real scene text detection these years. Current text detection approaches can be classified into two categories.

The first category is the texture based methods. Shin et al. [5] use a star-like pixel mask to expose the intrinsic features of text occurrences. In [6], P. Clark et al. carefully propose 5 localized measures and use a combination of these measures to get candidate text regions. The frequency domain techniques are also used to detect text-like texture, such as: Fourier Transform on short scanning line [8], discrete cosine transform [13], Gabor Transform [4], Wavelet decomposition [2], Multi-resolution edge detector [10]. We find these methods perform quite well on relatively small characters such as text lines on a menu or a document, because smaller texts often possess stronger texture responses. However, for big characters such as road signs or shop names (like Fig. 1), the strong texture response of complex background will mislead these algorithms and leave the big characters undiscovered.

Fig. 1. A difficult natural scene image

The second category is the connected component (CC) based methods. Color quantization [14], Morphological operation [7] and Symmetric Neighborhood Filters [9] are often used to form the candidate CCs. We find these methods can effectively deal with the big characters as well as the small ones, but to choose the exact text CC from the candidate ones often relies on heuristic rules, such as: aspect ratio [7][12][14], aligning-and-merging analysis [14], layout analysis [10], Hierarchical Connected Components Analysis[9]. These rules are often instable and can not guarantee robust detection result.

In this paper, we propose a more stable and more robust CC-based algorithm. This algorithm can enable us to integrate the heuristic rules and features in a more regularized and effective way. Therefore, our algorithm can effectively tackle various difficulties in the natural scene: such as complex background, complex text layout, different text language, uneven illumination, wild variation of text size and orientation.

The framework of our proposed algorithm is showed in Fig. 2. The method is composed of three stages. In the first stage, we employ a novel Non-linear Niblack (NLNiblack) method, which can efficiently and effectively decompose the gray image into candidate CCs. In the second stage, every candidate CC is fed into a series of classifiers and each classifier will test one feature of this CC. If one CC is rejected by any of the classifiers in the cascade, then it is considered as a non-text CC and need no further judgment. In the last stage, the CCs passing through the whole classifier cascade will be processed by a post processing procedure and form the final segmentation result.

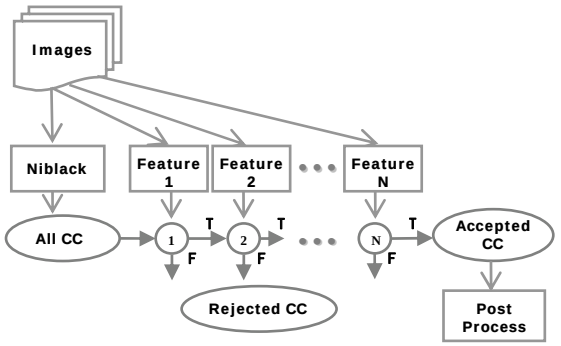


Fig. 2. The text detection algorithm

This framework substantially differs from the existing text detection algorithms in two key points.

The first point is that we utilize a classifier cascade, which can easily discard the majority of the non-text CCs and quickly focus on more promising text CCs. This idea is inspired by face detection technique [15] and is capable of processing images rapidly while achieving high detection rates. However, due to the essential difference between text detection and face detection, we originally propose a specific learning scheme for text detection problem.

The second point is that we propose a series of novel features, each of which has specific contribution to the text detection task. As we will show later, some of the features can take the advantage of texture characteristic of the image, some of them can exploit the spatial coherent information and some of them can efficiently speed up the whole algorithm, etc, By using these features, our algorithm can possess the advantages of both texture based methods and CC based methods, while suppressing their drawbacks. Our work is an innovative attempt to formulate a series of features for text detection.

We developed a prototype system using a mobile phone, Sony Ericsson S700c, attached with a 120 Mega pixel sensor and exhibited this system in Shanghai International Industry Fair 2004. It can automatically detect, segment and translate the English and Japanese signs into Chinese and prove the effectiveness and the efficiency of our algorithm.

The paper is organized as follows. In section 2, we present the non-linear Niblack decomposition method. Section 3 gives twelve features for effectively discriminating the text CCs from non-text ones. Then we describe how to train the classifier cascade using these features in section 4 and we also describe the post process in this section. Issues in system development and the experimental result are discussed in section 0. Section 6 gives the conclusion.

2. Non-linear Niblack decomposition

As we know, decomposing the image into a set of CCs is a very crucial step in CC-based methods. If the decomposition step gets poor results, the performance of the whole algorithm will drop dramatically. There are several existing methods [7][14][9] aiming at effective and robust decomposition. Beside this concern, the efficiency of computation and the low complexity of implementation also concern us. Therefore, we propose a very efficient non-linear Niblack (NLNiblack) thresholding method inspired by [16]:

$$NLNiblack(x,y)=\begin{cases}1 & f(x,y)>T_{+}(x,y)\\ -1 & f(x,y)<T_{-}(x,y)\\ 0 & others\end{cases} \quad (1)$$

$$T_{\pm}(x,y)=\hat{\mu}_{p1}(x,y,W_B)\pm k\cdot\hat{\sigma}_{p2}(x,y,W_F)$$

$$\hat{\mu}_{p1}(x,y,W_B)=Order\left[Mean\left(f(x,y),W_B\right),p1,W_B\right]$$

$$\hat{\sigma}_{p2}(x,y,W_F)=Order\left[Deviation\left(f(x,y),W_F\right),p2,W_F\right]$$

where

k is set to be 0.18 as standard Niblack method.

f(x,y) is the input pixel intensity at position (x,y).

Mean(• , W) is the mean value filter with W width.

Deviation(• , W) is the standard deviation filter with W width.

Order [• , p,W] is the ordered statistics filter with p percentile and W width.

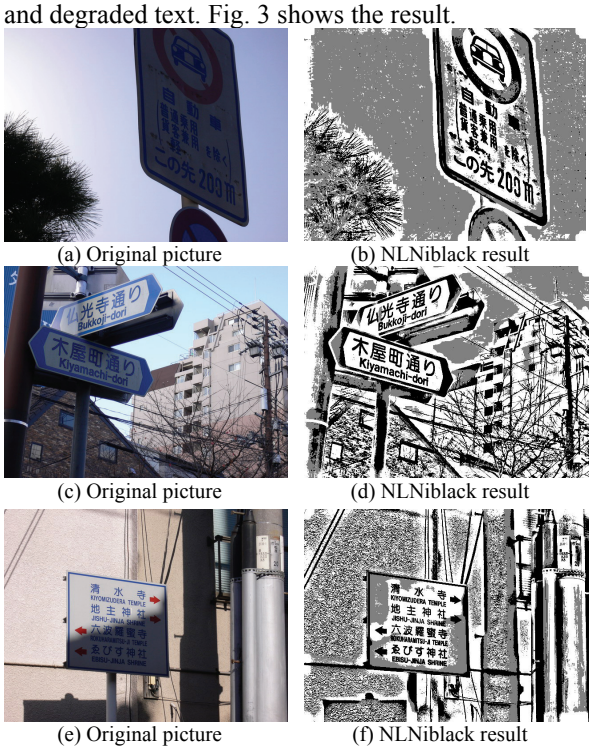
The difference between the NLNiblack and the original Niblack is that we just add two ordered statistics filter Order[. , p,W] to the background filter $\hat{\mu}_{p1}(x,y,W_B)$ and foreground filter $\hat{\sigma}_{p2}(x,y,W_F)$.

In the background filter $\hat{\mu}_{p1}(x,y,W_B)$, the filter width, W_B, is equal to 1/16 of image width and p1 is set to be 50%. It is because the large median filter can extract the background objects while not

excluding their high frequency components. This background filter can handle the uneven lighting in natural scenes.

In the foreground filter $\hat{\sigma}_{p2}(x, y, W_F)$, the filter width W_F, is 1/5 of W_B and p2 is set to be 80%. This high percentile filter can effectively 'spread' the influence of small areas with high variance to neighboring regions and can effectively increase local noise suppression.

Then we label the CCs in two thresholded layers, 1 and -1, respectively. The proposed NLNiblack decomposition can effectively handle the difficult conditions, such as low contrast, uneven illumination and degraded text. Fig. 3 shows the result.



(a) Original picture (b) NLNiblack result

(c) Original picture (d) NLNiblack result

(e) Original picture (f) NLNiblack result

Fig. 3. NLNiblack result: black 1, white -1

3. Features to detect characters

After decomposing the image into a set of CCs, we convert the segmentation problem into a classification problem – all we need to do is to classify all candidate CCs into 2 categories, text or non-text. Then we propose 12 novel features to expose the intrinsic characteristics of text CCs.

3.1. Geometric Features

The first three features are *geometric feature*. They are just some common features but can effectively discard a large proportion of apparently non-text CCs with very small computational expense. So they can dramatically decrease the execution time of the whole algorithm.

Area Ratio is used to discard too big or too small CCs:

$$Feature\_AreaRatio = \frac{Area(\mathrm{CC})}{Area(\mathrm{Picture})} \quad (2)$$

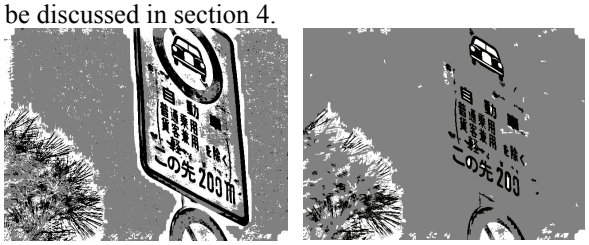
Length Ratio is used to discard too long or too short CCs:

$$Feature\_LengthRatio = \frac{\max\{w,h\}}{\max\{\mathrm{PicW},\mathrm{PicH}\}} \quad (3)$$

Aspect Ratio is used to discard too thin CC:

$$Feature\_AspectRatio = \max\{w/h, h/w\} \quad (4)$$

According to these three features, we can build three classifiers, each of which will test one feature. The effect of these geometric features can be viewed in Fig. 4.—after the filtering process of the geometric classifiers, apparently non-text CCs are filtered out. The way of training the classifiers will be discussed in section 4.



(a) Input: NLNiblack Result (b) after geometric classifiers

Fig. 4. Effect of geometric classifiers

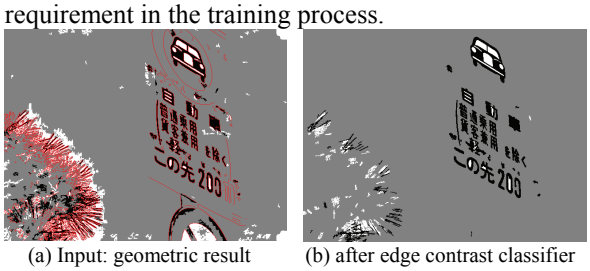
3.2. Edge Contrast Feature

The Edge Contrast Feature plays the most important role in the whole algorithm. Proposing this feature is based on a very common observation – regardless the complex background and the uneven lighting, text CCs are often 'highly closed' by edge response. Therefore, we use Eq.(5) to measure the edge closure degree of a CC. This feature fully takes the advantages of the texture based detection methods and moreover it also has a very strong response to large characters.

$$Feature\_EdgeContrast = \frac{Border(CC) \cap Edge(Picture)}{Border(CC)} \quad (5)$$

$$Edge(Picture) = Canny(Picture) \cup Sobel(Picture)$$

where Canny(Picture) and Sobel(Picture) mean the normalized Canny and the Sobel response of the image, respectively. And Border(CC) means the border pixels of the CC. This feature provides an image independent measurement of every CC's edge contrast. This kind of independency is a key requirement in the training process.



(a) Input: geometric result (b) after edge contrast classifier

Fig. 5. Effect of edge contrast classifiers

In Fig. 5.(a) the red mask is the *Edge(Picture)* response and we can find CCs with small edge closure degree are discarded.

3.3. Shape Regularity Feature

Text CCs often possess more regular shape than arbitrary noise CCs in the natural scene. Based on this observation, we propose 4 features: *Holes*, *Contour Roughness*, *Compactness* and *Occupy Ratio* (Eq.(6)). We can find text CCs often have smaller value in *Holes* and *Contour Roughness*, but larger value in *Compactness* and *Occupy Ratio*, while non-text CCs behave just the opposite. These features are used to suppress the noise which have irregular shape but have strong texture response.

$$Feature\_ContourRoughness = \frac{\left|CC - open\left(imfill\left(CC\right), 2\times 2\right)\right|}{\left|CC\right|} \quad (6)$$

$$Feature\_CCHoles = \left|imholes\left(CC\right)\right|$$

$$Feature\_Compact = \frac{Area\left(CC\right)}{\left|Border\left(CC\right)\right|^2}$$

$$Feature\_OccupyRatio = \frac{Area\left(CC\right)}{Area\left(BoundingBox\left(CC\right)\right)}$$

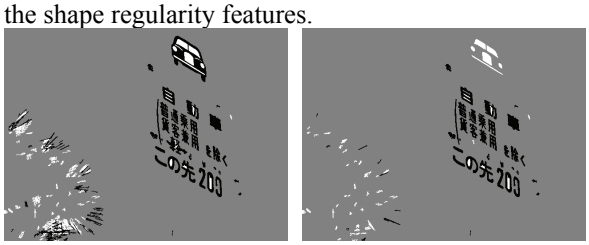
where

imfill(·) fills the holes in the CC.

imholes(·) count the holes in the CC.

BoundingBox(·) is the bounding box of the CC.

In Fig. 6, we can see the irregular noises with high texture and contrast responses are effectively reduced. For instance, it is very difficult to discard the small 'CAR' symbol on the board without using the shape regularity features.



(a) Input: edge contrast result (b) after shape reg classifier

Fig. 6. Effect of shape regularity classifiers

3.4. Stroke Statistics Feature

Character is composed of strokes, so we proposed 2 computational demanding features which expose the stroke statistics about CC. These two features check other aspects of 'irregularity' in the term of character stroke.

The first feature is *Mean Stroke Width* based on the observation that character stroke width is often relatively small:

$$Feature\_Stroke\_Mean = Mean\left(strokeWidth\left(skeleton\left(CC\right)\right)\right) \quad (7)$$

The second feature is *Normalized stroke deviation* based on the observation that strokes of character often have similar width and the CC with big stroke variance is more likely to be noise:

$$Feature\_Stroke\_std = \frac{Deviation\left(strokeWidth\left(skeleton\left(CC\right)\right)\right)}{Mean\left(strokeWidth\left(skeleton\left(CC\right)\right)\right)} \quad (8)$$

In Eq. (7) and (8), skeleton(·) stands for the morphological skeleton operation and strokeWidth(·) stands for the shortest distance between the pixel on the CC skeleton to the outside pixels.

(a) Input: NLNiblack Result (b) after stroke classifiers

Fig. 7. Effect of stroke statistic classifiers

In Fig. 7, we can find that the big characters survive after these classifiers while noises are effectively reduced.

3.5. Spatial Coherence Features

The last two spatial coherence features exploit the spatial coherence information to filter out the non-text CCs. Noises will have less spatial regularity and coherence, so we propose these two features:

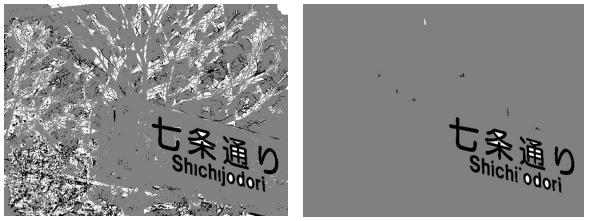
Spatial coherence area ratio

$$Feature\_AreaRatio\_S = \frac{Area\left(imdilate\left(CC, 5\times 5\right)\right)}{Area(\text{Picture})} \quad (9)$$

Spatial coherence boundary touching

$$Feature\_Boundary\_S = Bound\left(imdilate\left(CC, 5\times 5\right)\right) \quad (10)$$

In Eq.(9) and (10), imdilate(·, strel) stands for the morphological dilation operation with structural element, strel. In this stage, the apparently non-text CCs have already been discarded. Then in every layer, if some CC expands significantly after being dilated with a small structural element, it is more likely to be spatially correlated random noise. On the contrary, the text CCs will not act like this because of the structural nature of characters. By using the spatial coherence features, we can efficiently reduce the noises (In Fig. 8).



(a) Input: stroke result (b) after spatial classifiers

Fig. 8. Effect of spatial coherence classifiers

4. Classifier cascade training

Since we have already had a set of CCs and for every CC we have 12 features which can effectively separate text CC from non-text CC, the remaining

problem is how to use these features. The easiest solution is heuristically setting all thresholds manually. This is a very unstable approach. Therefore, a better solution may be the machine learning method.

Then the further problem is which machine learning method to use. Since some of the features we use are computational demanding, such as, *stroke statistics* features and *edge contrast* feature, it is unwise to calculate all of the 12 features together during classification. We need a mechanism to discard most of the non-text CCs by less computation. This requirement reminds us of the Adaboost scheme and the attentional cascade architecture used in face detection [15].

Although we use Adaboost to train all the classifiers and also build an attentional cascade, our method substantially differs from the techniques used in [15] because text detection and face detection are two essentially different tasks. Table 1 gives a comparison between these two tasks.

Table 1. Text detection vs. face detection

	Text detection	Face detection
Basic unit	Connected component	24x24 detect window
Feature num	12 / CC	45,396 / Window
Feature Quality	High	Vary violently
Negative sample	Easy to find	Need careful consideration
Performance Information	Not known in advance	Known after feature selection

4.1. Notation

Before going into training scheme details, we will clarify the notation we use at first. See Table 2.

Table 2. Notations

f	False positive rate: $\frac{area(error)}{area(negative)}$
d	Detection rate: $\frac{area(hit)}{area(positive)}$
FR	False rejection rate: $1-f=\frac{area(negative)-area(error)}{area(negative)}$
P	positive training set
N_i	ith negative training set
f_i	maximum false positive rate of ith layer
d_i	minimum detection rate of ith layer
F	overall false positive rate
D	overall detection rate
M	number of classifier in the cascade
h_i	ith weak classifier in the cascade
w_i	weight of ith classifier in Adaboost learning scheme

4.2. Important Assumption

We will feed all the CCs into the classifier cascade. If one CC is rejected by any of classifier, it is regarded as non-text CC. Therefore, it is easy to know that we have the following relationship:

$$F=\prod_i^M f_i \qquad D=\prod_i^M d_i \tag{11}$$

$$\log(F)=\sum_{i=1}^M \log(f_i) \quad \log(D)=\sum_{i=1}^M \log(d_i)$$

In the logarithm conversion form of the basic relationship, we can find that the overall detection rate is linearly dispatched to all the classifiers. Then we can assume the logarithm form of minimum detection rate is linearly dispatched according to the 'quality' of each classifier. Therefore, we will have the dispatching formulation as follows:

$$d_i=\left(D_{dispatch}\right)^{\gamma} \tag{12}$$

where $D_{dispatch}$ is the detection rate can be dispatched and γ stands for the 'quality' portion of the ith classifier. We find that this formula has close relationship to the idea of indifference curve proposed by J. Sun. et.al [17].

4.3. Cascade Building Process

First, we will use the standard Adaboost training scheme [15] to train a *strong classifier*, a linear combination of 12 weak classifiers. Every weak classifier only responds to one single feature of CC and makes the decision whether the CC is text or not.

- User selects overall minimum detection rate D_{target}.
- Random Select 200 pictures from total 368 pics
  - P = *set of positive examples*
  - N = *set of negative examples*
- $F_0 = 1.0$; $D_0 = 1.0$; $i = 0$
- *Feature = { feature$_j$ | j = 1 to M }*

for i = 1: M
  $D_i = D_{i-1}$
  foreach feature$_j$ in *Feature*
    get distribution of feature$_j$ based on {P,N}
    calculate d_j (D_i), f_j (D_i) FR_j (D_i,1- D_i)
  end
  choose the feature k with Biggest f_k (D_i)
  $\gamma = FR_k(D_i,1-D_i)/SUM_j(FR_j(D_i,1-D_i))$;
  $d_i = (D_{target}/D_i)^{\gamma}$;
  training: $d_i = h_i(d_i, P, N)$
  $N = \varnothing$
  evaluate the current cascaded detector h_i on the set of non-text CCs and put any false detections into the set *N. and* $D_i = D_i * d_i$;
  Feature=Feature- feature$_k$;
end

Fig. 9. Cascade training algorithm

Second, based on the combination weight we get from Adaboost, we use the following algorithm to train the attentional cascade (Fig. 9). Our method differs from the existing methods in adaptively

dispatching the entire expected detection rate into 12 classifiers.

Third, we will add a post process part after the cascade. The *strong classifier* we train in the first step will be used as the 13th classifier in the cascade. All 12 features of the CC passing through the previous cascade have been calculated, so only a linear combination operation is needed for the *strong classifier*, which can further improve the accuracy.

The last but not least, we will combine the CCs in the black layer and white layer together to form the finial result. We will compare the adjacent CCs' confidence margin which is obtained by the 13th classifier, and then omit the CCs with smaller margin. The remaining CCs are considered as final result.

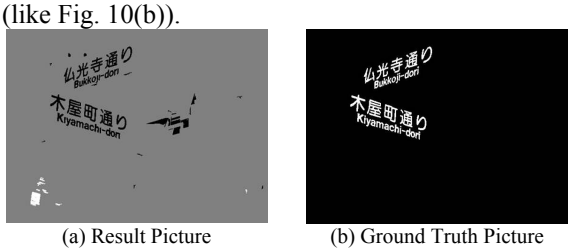
5. Experimental Results

5.1. System architecture

We implemented a prototype system and exhibited it in Shanghai International Industry Fair 2004. We use Sony Ericsson S700c, which is attached with a 120 Mega pixel sensor, to take a photo of the natural sign. Then this image is transferred through Bluetooth OBEX protocol to a processing server, 1.6GHz CPU and 256M RAM. After seeing the image arrive, the server will do the detection and segmentation of the image. The segmented regions are regularized and then sent to the recognition and translation module. Finally, the result image is sent back to the mobile. The whole process is done in less than 1 s, and this demo shows that our segmentation algorithm is very robust and fast.

5.2. System evaluation

To better evaluate our algorithm, we built a database containing 368 difficult scene images (640x480) and labeled all the ground truth manually (like Fig. 10(b)).



(a) Result Picture (b) Ground Truth Picture

Fig. 10. Evaluation the result

We employ a very strict pixel-wise evaluation criterion to measure performance, showed as follows:

$$\begin{aligned} hit &= area\left(Result \,\&\, GroundTruth\right) \\ error &= area\left(Result \,\&\, \overline{GroundTruth}\right) \\ miss &= area\left(\overline{Result} \,\&\, GroundTruth\right) \\ precision &= \frac{hit}{hit + error}, \quad recall = \frac{hit}{hit + miss} \end{aligned} \tag{13}$$

The evaluation score is showed below (Table 3) and we can find that our algorithm is very robust:

Table 3. Overall performance

	Precision	**Recall**
Training set	92.3%	98 %
Testing set	88.9%	97.5 %

Besides the standard evaluation, we also establish experiments to prove the effect of every feature (see Fig. 11):

Fig. 11. Effect proof for every feature

We omit one feature in the whole cascade and then evaluate the final precision without this feature. We can find that without the *edge contrast* feature, the overall performance drops sharply, which indicates the *edge contrast* feature contributes most on the performance. On the contrary, the *geometric* features almost contribute nothing in the precision.

The average running time of the algorithm processing one picture is 0.34s. In a more detailed experiment, we also omit the features one by one to see their contribution to the average running time. In Fig. 12, we can find that without the *geometric* features, the running time will increase to 1.72s. It is saying that the *geometric* features can effectively discard a lot of non-text CCs in very small computational cost. On the contrary, stroke features are the most computational demanding features, but thanks to the previous classifiers, it will just exam the most promising text CCs, so it will not impose great burden on the algorithm efficiency.

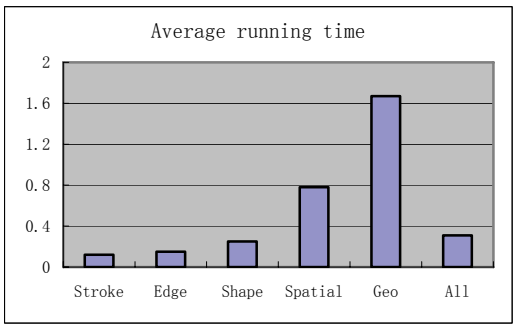


Fig. 12. Efficiency proof for every feature

6. Conclusions

In this paper, we present a novel detection algorithm for scene text. In sum, our contributions are:

- Propose a fast and robust decomposition method called NLNiblack.
- Propose 12 novel features for connected component based detection method.
- Propose an Adaboost modification to train the cascade on text detection problem.
- Implement a fast and robust prototype system.

Acknowledgements

The authors would like to thank reviewers for their comments on this paper. This work was performed at Computer Vision Laboratory, SJTU, and was supported by OMRON under PVS project.

References

[1] Doermann, Progress in camera-based document image analysis, 7th ICDAR Conference, 2003

[2] Li, D. Doermann and O. Kia. Automatic Text Detection and Tracking in Digital Video. IEEE Transactions on Image Processing. Vol. 9, No. 1, pp. 147-156, Jan. 2000.

[3] Rainer Lienhart. Automatic Text Recognition for Video Indexing. Proc. ACM Multimedia 96, Boston, MA, pp. 11-20, Nov. 1996.

[4] MulS. Ferreira, C. Thillou, B. Gosselin, 2003, From Picture to speech: an innovative OCR application for embedded environment, 17th ProRISC 2003, Veldhoven

[5] C.S. Shin, K.I. Kim, M.H. Park, H.J. Kim. Support Vector Machine-based Text Detection in Digital Video. Proceedings of the IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing X, Vol. 2, pp. 634-641, 2000.

[6] P. Clark and M. Mirmehdi. Finding Text Regions Using Localised Measures. Proceedings of the 11th British Machine Vision Conference, pp. 675-684, September 2000.

[7] Yassin M. Y. Hasan and Lina J. Karam, Morphological Text Extraction from Images, IEEE Transactions on Image Processing, Vol. 9, No. 11, November 2000

[8] Byung Tae Chen, Younglae Bae, Tai-Yun Kim, Automatic Text Extraction in Digital Videos using FFT and Neural Network, IEEE International Fuzzy Systems Conference Proceedings, August 22-25, 1999, seoul, Korea

[9] Ismail Haritaoglu, Scene text extraction and translation for handheld devices, 2001 IEEE

[10] J. Gao and J. Yang, An Adaptive Algorithm for Text Detection from Natural Scenes, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2001

[11] Takuma Yamaguchi, Minoru Maruyama, Character Extraction from Natural Scene Images by Hierarchical Classifiers. ICPR 04, 687-690, 2004

[12] Nobuo Ezaki, Marius Bulacu, Lambert Schomaker, Text Detection from Natural Scene Images: Towards a System for Visually Impaired Persons. ICPR, 683-686, 2004

[13] Yu Zhang, et al., Automatical caption localization in compressed video, IEEE Trans. Pattern Anal. Mach. Intell. 22 (4) (2000) 385–392.

[14] Kongqiao Wanga, Jari A. Kangasb, Character location in scene images from digital camera, Pattern Recognition 36 (2003) 2287 – 2299

[15] Paul A. Viola, Michael J. Jones: Robust Real-Time Face Detection. ICCV 2001: 747

[16] L. Winger, J.A. Robinson, M. ED Jernigan, Low-Complexity Character Extraction In Low-Contrast Scene Images, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14, No. 2 (2000) 113-135

[17] Jie Sun Rehg, J.M. Bobick, A. , Automatic cascade training with perturbation bias, CVPR 2004, II-276- II-283 Vol.2, July 2004

Fig. 13. More experiment results: (a)~(e),(k)~(o),(u)~(y),(F)~(J)original pictures, (f)~(j),(p)~(t),(A)~(E),(K)~(O) result pictures

Data Embedding for Camera-Based Character Recognition

Seiichi Uchida
Kyushu University
Fukuoka-shi, 812–8581 Japan
uchida@is.kyushu-u.ac.jp

Masakazu Iwamura
Osaka Prefecture University
Sakai-shi, Osaka, 599–8531 Japan
masa@cs.osakafu-u.ac.jp

Shinichiro Omachi
Tohoku University
Sendai-shi, 980–8579 Japan
machi@aso.ecei.tohoku.ac.jp

Koichi Kise
Osaka Prefecture University
Sakai-shi, Osaka, 599–8531 Japan
kise@cs.osakafu-u.ac.jp

Abstract

In this paper, the embedment of class information into each character image is investigated for camera-based character recognition as easy and accurate as bar-code reading. Each character image is printed with a horizontal stripe pattern, called a cross ratio pattern, and the class information is represented as a cross ratio derived from the pattern. Since the cross ratio is invariant to projective distortion, the class information is extracted correctly regardless of camera angle. Experimental results showed that the cross ratios are extracted from distorted character images with very high accuracy and thus very effective to attain high character recognition rates.

1. Introduction

Camera-based character recognition [1] is a promising way for acquiring various textual information from real scenes. Several hurdles, however, should be cleared for practical and accurate camera-based character recognition. For example, the character images often undergo geometric distortions, such as projective distortion.

The aim of this paper is to realize accurate camera-based character recognition by embedding class information into each character image. Specifically, each character image is printed with a horizontal stripe pattern, called a *cross ratio pattern*. The cross ratio derived from the cross ratio pattern represents the class information of the character. Since the cross ratio is invariant to the projective distortion [2], the class information will be correctly extracted even from character images captured from an arbitrary camera angle.

In Section 2, we describe how a cross ratio is embedded into a character image for providing the class information of the character. The extraction of the embedded cross ratio from the character image is also discussed in this section.

When the variations of the cross ratios are fewer than character classes, the same cross ratio is assigned to several different character classes. In this case, we cannot determine the character class uniquely from the extracted cross ratio. Thus, in Section 3, we use a shape similarity between reference and input character images as well as the cross ratio for the unique determination. In Section 4, we point out that the assignment of the cross ratios to the character classes affects the recognition performance attained by the combination of the cross ratio.

In Section 5, we evaluate the proposed technique quantitatively through recognition experiments. In Section 6, the proposed technique is compared to other strategies where class information is provided in different manners. Finally, we present our conclusions and future works in Section 7.

2. Embedment of cross ratio pattern to character image

2.1. Cross ratio pattern

In the proposed technique, a horizontal stripe pattern, called a *cross ratio pattern*, is embedded to each character image. Figure 1(a) shows a character image "K" printed with a cross ratio pattern. Characters of a certain class is printed with the same cross ratio pattern. The cross ratio pattern is comprised of five horizontal stripes. The first and the last stripes are guides which have a fixed width and define the beginning and the end of the cross ratio pattern, respectively. The remaining three stripes have variable widths, l_1, l_2, and l_3.

Instead of using l_1, l_2, or l_3 directly, we use the following numerical value r, called the *cross ratio*, for representing

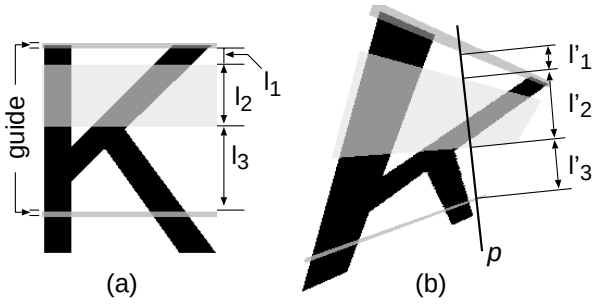


Figure 1. (a) A character image "K" where a cross ratio pattern is embedded. (b) Projective distortion. Note that a high-contrast cross ratio pattern is intentionally used here for visual emphasis.

class information:

$$r = \frac{(l_1 + l_2)(l_2 + l_3)}{l_2(l_1 + l_2 + l_3)}. \quad (1)$$

It is well-known that the cross ratio is invariant to projective distortions. Thus, by using the cross ratio, we can extract the class information correctly regardless of camera angle.

Since character classes are discrete, the cross ratio r is discretized into K levels, r_k $(k = 1, 2, \ldots, K)$, and assigned to $|\boldsymbol{C}|$ classes, where $\boldsymbol{C}$ is the set of character classes. The detail of the assignment will be discussed in Section 4.

2.2. Extraction of cross ratio

The cross ratio r_k can be extracted from a character image printed with a cross ratio pattern by the following procedure:

Step 1: Draw a line p which crosses two guides (Fig. 1(b)).

Step 2: Measure the widths of the three stripes on p (l'_1, l'_2, and l'_3 of Fig. 1(b)).

Step 3: Using l'_1, l'_2, and l'_3 instead of l_1, l_2, and l_3, obtain r_k according to (1).

The value r_k obtained by this procedure is theoretically invariant to projective distortions. This means that we can extract the same cross ratio r_k regardless of camera angle. In addition, the value r_k is also invariant to the position and the slope of the line p.

The accuracy of the extracted cross ratio may be degraded due to insufficient camera resolution. In order to avoid this degradation, we use the following robust estimation strategy: (i) we draw the line p on the character image P times changing its position and slope randomly, (ii) obtain P cross ratio values, (iii) quantize each of those values into one of r_k, and (iv) choose the most frequent r_k as the cross ratio embedded.

2.3. Design of cross ratio patterns

The K cross ratios, $r_1, \ldots, r_k, \ldots, r_K$, are prepared by changing the proportion of l_2 and l_3. Specifically, assuming $L = l_1 + l_2 + l_3$ and l_1 are constant, r_k is determined by (1) with the following l_2 and l_3:

$$\begin{cases} l_2 &= \dfrac{(L - l_1 - 2\epsilon)(k-1)}{K-1} + \epsilon, \\ l_3 &= L - l_1 - l_2, \end{cases} \quad (2)$$

where ϵ is a positive constant specifying the minimum of l_2 and l_3.

The above strategy is based on a simple linear quantization and may be weak against errors on the stripe widths l_1, l_2, and l_3 due to the insufficient camera resolution. In fact, larger k becomes, closer r_k and r_{k+1} become. Thus, a small error on the stripe widths may confuse those close cross ratios. Future work should focus on a more sophisticated strategy to avoid the confusion as possible.

3. Recognition by cross ratio and shape similarity

In most cases, we cannot expect one-to-one assignment of K cross ratios to $|\boldsymbol{C}|$ classes. Specifically, $|\boldsymbol{C}|$ is often large (e.g., $|\boldsymbol{C}| > 1000$ for Chinese characters) whereas K is bounded by $L - l_1 - 2\epsilon$ ($\sim$ character height in pixel) according to (2). Thus, the same cross ratio r_k will be assigned to several classes $\boldsymbol{C}_k \subset \boldsymbol{C}$, where $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_k, \ldots, \boldsymbol{C}_K$ are disjoint subsets of $\boldsymbol{C}$, and therefore the class c of an input character image cannot be determined by the extracted cross ratio r_k. In other words, there are $|\boldsymbol{C}_k|$ candidates of the correct class when r_k is extracted.

For choosing the most reliable class from the $|\boldsymbol{C}_k|$ candidates, we employ some shape similarity between two character images. Assuming that a reference character pattern (i.e., a template) is prepared for each class, the complete recognition procedure based on a combination of the cross ratio and the shape similarity is as follows:

Step 1: Extract the embedded cross ratio r_k from an input character image by the procedure of Section 2.2.

Step 2: For each class in $\boldsymbol{C}_k$, calculate the shape similarity between the reference character image of the class and the input character image.

Step 3: Choose the class with the highest shape similarity.

Note that this procedure totally relies on the extracted cross ratio r_k. If a wrong r_k is extracted, the correct class

Figure 2. Character images printed with different cross ratio patterns (i.e., $K = 26$).

is never chosen by the procedure. Fortunately, the cross ratio r_k can be extracted with high accuracy (around 99%, as shown in Section 5.2), thus good performance is expected.

4. Assignment of cross ratios to classes

The assignment of K cross ratios to $|\boldsymbol{C}|$ classes, that is, the partition of $\boldsymbol{C}$ into the disjoint subsets $\{\boldsymbol{C}_k\}$, is crucial for better performance of the proposed technique. The recognition procedure of Section 3 provides correct recognition result if (i) the cross ratio r_k is correctly extracted and (ii) the correct class has the highest shape similarity in $\boldsymbol{C}_k$. Thus, for better recognition performance by satisfying the latter condition (ii), the subset $\boldsymbol{C}_k$ should be comprised of classes which are "less easy to confuse" for the shape similarity, as shown in the following example.

Assume that "H" and "N" are confusing classes (that is, "H" is often misrecognized as "N" by the shape similarity) and "H" and "N" are assigned to the same subset $\boldsymbol{C}_k$. In this case, we will suffer from the misrecognition between "H" and "N", even though their cross ratios are correctly extracted. Clearly, this is because they cannot be distinguished by their cross ratios. In contrast, if these two classes are assigned to different subsets, they can be distinguished by their cross ratios and therefore correct recognition results will be provided. As shown by this example, the assignment $\{\boldsymbol{C}_k\}$ should be optimized with a criterion that confusing classes are assigned to different subsets. In the experiment of Section 5, the assignment was optimized by the strategy of [6], where the so-called confusion matrix of the shape similarity is used to identify its confusing classes.

5. Simulation experiment

5.1. Experimental setup

5.1.1. Original character images. The 26 capital English letter images from the font-set called "Arial" were used as original character images. After embedding cross ratio patterns into them (according to the scheme of the following sections), those images are used as not only reference patterns but also the source patterns for synthesizing test patterns. Their heights were around 200 pixels. On the other hand, their widths were not the same; the maximum, the minimum, and the mean of widths were 251 (of "W"), 52 (of "I"), and 170, respectively.

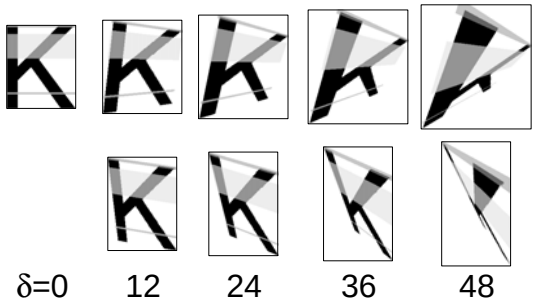


Figure 3. Test patterns.

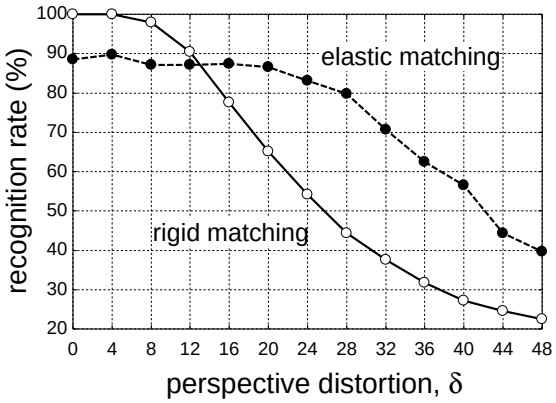


Figure 4. Recognition rates attained by using shape similarity alone.

5.1.2. Design of cross ratio patterns. According to the procedure of Section 2.3, $K(\leq 26 = |\boldsymbol{C}|)$ cross ratio pat-

Table 1. Confusion matrix by using the shape similarity by elastic matching.

input \ recognition result	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	14	0	0	241	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
E	0	26	0	0	230	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	250	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	256	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	182	0	0	0	0	64	10	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	42	0	0	0	0	0	0	0	0	0	0	214	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	252	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	6	0	0	0	0	0	0	0	0	0	0	24	0	226	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	256	0	0	0	0	0	0	0	0
S	0	4	0	0	0	0	58	0	0	0	0	0	0	0	0	0	0	0	194	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	256	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	113	0	0	0	0	0	0	0	140	3	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	175	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	256	0	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	256	0	0
Y	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	51	0	0	203	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	256

Table 2. Naive and optimal assignments of cross ratios to 26 classes. Note that the assignment is optimized for elastic matching.

class		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$K = 4$	naive	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
	opt.	1	1	1	1	2	3	3	2	1	2	1	2	3	4	4	2	2	4	1	4	2	1	1	4	3	2
$K = 12$	naive	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2
	opt.	1	2	3	4	5	6	7	5	4	8	2	9	6	10	11	8	9	10	1	11	12	3	4	11	7	12
$K = 20$	naive	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6
	opt.	1	2	3	4	5	6	7	8	9	10	11	12	6	13	14	15	16	17	18	14	19	3	9	20	7	19

terns, $r_1, \ldots, r_k, \ldots, r_K$, were designed. Figure 2 shows the original character images printed with $K = 26$ different cross ratio patterns. The width of the guide was 5 pixels. The widths L, l_1, and ϵ, were fixed at 150, 15, and 15 pixels, respectively. The assignment of K cross ratios to 26 classes, that is, the specification of $\boldsymbol{C}_k$ was done by the two strategies described in Section 5.1.5.

5.1.3. Test patterns. Test patterns were synthesized by applying projective distortions on the original character images with the cross ratio patterns. The projective distortion was simulated by displacing four corners of a character image for $\pm\delta$ pixels ($\delta = 0, 4, 8, \ldots, 48$) in their x and y directions. Thus, for each value of δ, 256 test patterns were created from a single original character image. Figure 3 shows several test patterns synthesized from the same character image of the class "K". This figure reveals that there are heavily distorted patterns in the test patterns.

5.1.4. Shape similarity. As noted in Section 3, the proposed technique can employ any shape similarity (or the score given by any conventional recognizer) between a reference pattern (i.e., an original character pattern) and an input pattern. In the experiment, the following two matching

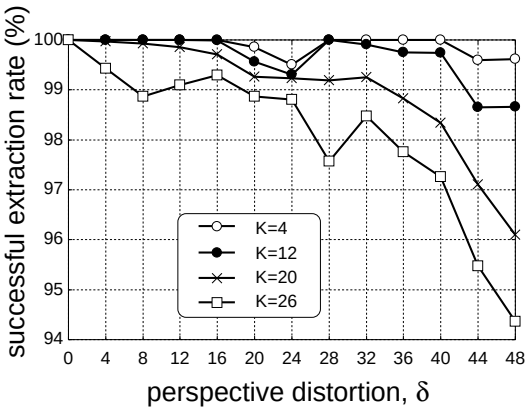


Figure 5. Extraction accuracy of cross ratios.

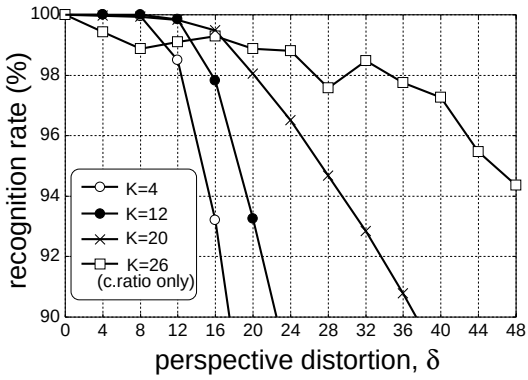


Figure 6. Recognition rate by *rigid* matching and cross ratio. The naive assignment was used here.

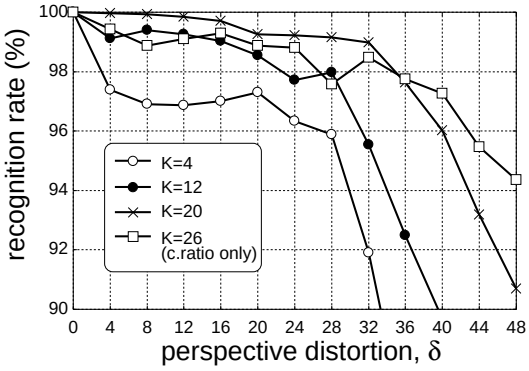


Figure 7. Recognition rate by *elastic* matching and cross ratio. The naive assignment was used here.

techniques are employed to evaluate the shape similarity.

- Rigid matching $\cdots$ This is a technique to obtain a similarity score by simple superimposing.
- Elastic matching$\cdots$ This is a technique to obtain a similarity score after fitting the reference pattern to the input pattern nonlinearly [7]. The elastic matching technique employed here possesses enough flexibility for compensating projective distortions.

Note that both techniques used simple gray-level as their pixel feature.

Figure 4 shows recognition accuracy attained by the shape similarities by the above two matching techniques. The rigid matching was very sensitive to projective distortions and its recognition accuracy decrease drastically according to the increase of δ. On the other hand, the elastic matching is rather robust to the projective distortions; its recognition accuracy does not decrease for $\delta \leq 28$. For more heavy distortions, however, its accuracy decreases like the rigid matching.

Table 1 is the confusion matrix by the shape similarity of the elastic matching for 256$\times$ 26 test data of $\delta = 4$. Most of "N" were misrecognized as "H" or "M" with the shape similarity alone because their shapes become similar after nonlinear fitting of the elastic matching.

5.1.5. Assignment of cross ratios to classes. The following two strategies were used for assigning cross ratios to classes.

- Naive assignment $\cdots$ K cross ratios are assigned to $|\boldsymbol{C}|$ classes by a simple numerical order.
- Optimal assignment $\cdots$ According to the discussion of Section 4, the assignment was optimized by the algorithm proposed in [6].

Table 2 shows the naive assignment and the optimal assignment for the elastic matching at $K = 4$, 12, and 20. As shown in this table, the same cross ratio is assigned to the classes "C" and "V", by the optimal assignment at any K. This fact means that "C" and "V" are not confusing classes for the elastic matching.

5.2. Extraction accuracy of cross ratios

Figure 5 shows the extraction accuracy of the cross ratios as a function of δ. This graph indicates that the cross ratios can be extracted very accurately even under heavy distortions. By comparing Figures 5 and 4, it is shown that this accuracy is often 10 (or more) times higher than the recognition rates by shape-similarities. Thus, the cross ratio is more reliable information than the shape similarities for camera-based character recognition.

The graph at $K = 26$ in Fig. 5 shows the recognition rate attained by using the cross ratio alone and that a recog-

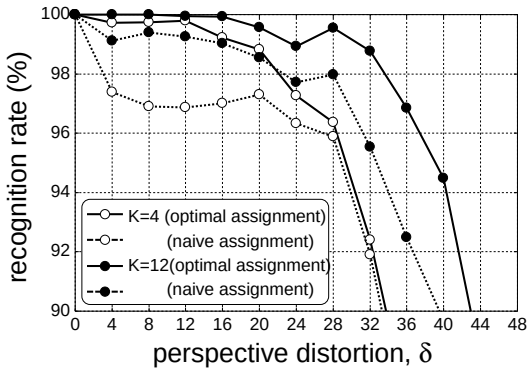


Figure 8. Naive assignment versus optimal assignment. The latter can attain higher rates under the same number of cross ratios, K.

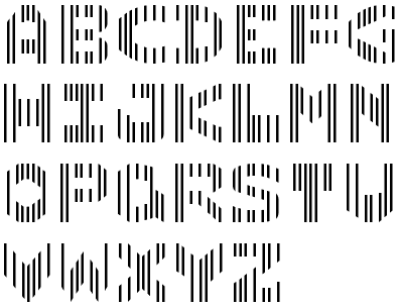


Figure 9. MICR font called C.M.C.7.

nition rate exceeds 98% without using any character shape information if $\delta \leq 24$.

Extraction failures are mainly due to slight errors (such as $\pm$ 1 pixel) of l'_1, l'_2, l'_3 by insufficient resolution. In fact, at $K = 26$, 85% of extraction failures are "near-misses" that r_k was detected as $r_{k\pm1}$. More serious failures that r_k was detected as $r_{k\pm\Delta}$ $(\Delta \geq 2)$ are 10%. The remaining 5% are the failures that the guide was not detected.

5.3. Recognition accuracy by using cross ratio and shape similarity

Figures 6 and 7 are the recognition rates attained by using the extracted cross ratios and the shape similarity according to the procedure of Section 3. As shape similarities, the rigid matching score and the elastic matching score were used in Fig. 6 and 7, respectively. In both cases, $K \in \{4, 12, 20, 26\}$ cross ratios were assigned to $|\boldsymbol{C}| = 26$ classes according to the naive assignment. (See Table 2, for the naive assignment at $K = 4, 12$, and 20.)

Those two figures firstly indicate that recognition rates are drastically improved from the rates of Fig. 4, that is, the rates attained by the shape similarities. At $\delta = 4$, for example, the recognition rate attained by the shape similarity by the elastic matching was 89.8% and improved to 97.4%, 99.1%, and 99.97% with $K = 4$, 12, and 20 cross ratios, respectively.

This improvement is achieved by removing the ambiguity in the shape similarity using the extracted cross ratio. For example, as indicated by the column "H" of the confusion matrix of Table 1, there are two correct classes candidates, "N" and "H", for an input character recognized as "H" by the elastic matching score. Fortunately, if its cross ratio is extracted correctly, the correct class is chosen from the candidates. This is because as shown in Table 2, different cross ratios are assigned to classes "N" and "H" by the naive assignment (for any K), and therefore the true class of the input character can be determined by the extracted cross ratio.

On the other hand, Figures 6 and 7 also show the usefulness of the shape similarities. In fact, when $\delta \leq 16$, the recognition rate attained by $K = 20$ exceeds that attained by $K = 26$ (where the recognition is done by only the extracted cross ratio). This is because, if many cross ratios are used (e.g., $K = 26$), their extraction accuracy is slightly degraded as shown in Fig. 5. Thus, it will be a reasonable strategy to (i) use as many cross ratios as possible under the condition that those cross ratios can be extracted near-perfectly and then (ii) remove the remaining ambiguity by a shape similarity.

5.4. Naive assignment versus optimal assignment

Figure 8 shows the recognition accuracies with two different assignment strategies, i.e., the naive assignment and the optimal assignment of Section 5.1.5. The shape similarity by the elastic matching was used here.

This result shows that the optimal assignment outperforms the naive assignment for any δ and K. This superiority comes from the fact that the optimal assignment can remove the ambiguity in confusing classes effectively. At $\delta = 4$, for example, the same cross ratio is assigned to "M" and "U" by the naive assignment as shown in Table 2, although "M" and "U" are confusing classes as shown in Table 1. On the other hand, different cross ratios are assigned to "M" and "U" by the optimal assignment since a confusion matrix used for the optimization confesses that "M" and "U" are confusing classes. It is noteworthy that optimally assigned 4 cross ratios outperform naively assigned 12 cross ratios for $\delta \leq 20$.

6. Relation to other techniques

6.1. OCR/MICR fonts and DataGlyph

The proposed technique is closely related to so-called "OCR fonts" and "MICR (magnetic ink character recognition) fonts", which were proposed in the dawn of OCR/MICR research [3]. In those fonts, class information is embedded into their shapes. Figure 9 shows the MICR font called C.M.C.7, where each character is comprised of six vertical lines with class-dependent intervals. DataGlyph [4, 5] is a more recent font where some data is embedded as a fine hatching pattern.

Those conventional fonts are not designed to be robust against perspective distortions. For example, the interval of the vertical lines of the C.M.C.7 font will vary according to perspective distortions. Thus, for camera-based recognition, some dewarping process, which itself is not a trivial task in general, should be performed on those fonts in advance.

6.2. Barcode and watermark

Barcodes are also related to the proposed technique. If a barcode represents a text, we can read the text by a barcode scanner with very high accuracy. Recently, two-dimensional barcodes, such as QR code, have been developed as pictorial codes having larger data capacities. Among them, the QR code is promising because it can be read under perspective distortion.

The barcodes have the following drawbacks when they are used for representing some text data:

- Barcodes are machine-readable and not human-readable. Thus, users cannot know in advance what a barcode represents.
- Barcodes cannot allow "partial reading" that a user tries to read only a part of an entire text.
- Barcodes are printed separately from character images. Thus, barcodes should be "conspicuous" enough to show their existence. This means that barcodes will spoil the design of documents. The longer texts becomes, the larger, i.e., the more unsightly, a barcode becomes.

Watermark is invisible or near-invisible data representation and often embedded into the background of a document. Watermark also has the first and the second drawbacks of the barcodes because generally it is encoded end embedded by a special manner (i.e., not human-readable) and has no explicit correspondence to individual characters (i.e., not partially readable). On the other hand, watermark may avoid the third drawback of the barcodes, i.e., unsightliness, because of its invisibility; however, this fact leads a conflicting situation. If a watermark is perfectly concealed on a document to avoid the unsightliness, a user cannot detect it and thus cannot extract an embedded text from the watermark. Hence, if a watermark is used, a "visible mark" that indicates the location of the watermark is necessary.

7. Conclusion and future work

For camera-based character recognition as easy and accurate as bar-code reading, the embedment of class information into each character image is investigated. The class information is represented as a horizontal stripe pattern, called a cross ratio pattern, and the character image is printed with the pattern. Since cross ratio is invariant to projective distortion, the same class information can be extracted from character images captured at an arbitrary camera angle. Experimental results showed that (i) the cross ratio can be extracted from distorted character images with very high accuracy and (ii) the cross ratio can enhance the recognition performance of conventional matching-based recognizers.

Future work will focus on the following points:

- Experiments using character images captured by a camera.
- Improvement of the design of the cross ratio patterns. It is also possible to use some distortion invariant other than cross ratio.
- Improvement of shape similarity. If the confusion matrix by the shape similarity can be sparse, the number of cross ratios can be saved.
- Embedment of data other than class information. For example, copyright information to character strings can be embedded by cross ratio patterns.

Pattern recognition research has a long history of the struggle to recognize image patterns having high human-readability and low machine-readability. Handwritten character recognition is a typical example. In upcoming ubiquitous computing age, image patterns are often exposed to computers via cameras and therefore often to be acquired/recognized by computers. Thus, it will become more important to use image patterns having not only high human-readability but also high machine-readability as a medium for human-machine communication. The proposed character image with the cross ratio pattern can be a promising candidate as the medium.

References

[1] D. Doermann, J. Liang and H. Li: "Progress in camera-based document image analysis", Proc. ICDAR'03, pp. 606–616 (2003).

[2] R. O. Duda and P. E. Hart: Pattern Classification and Scene Analysis, John Wiley& Sons, 1973.

[3] The British Computer Society: Character Recognition 1967, Unwin Brothers Limited (1966).

[4] D. L. Hecht: "Printed embedded data graphical user interfaces", IEEE Computer, vol. 34, no. 3, pp. 47–55 (2001).

[5] K. L. C. Moravec: "A grayscale reader for camera images of Xerox DataGlyphs", Proc. of The British Machine Vision Conference, pp. 698–707 (2002).

[6] M. Iwamura, S. Uchida, S. Omachi, and K. Kise: "Recognition with Supplementary Information — How Many Bits Are Lacking for 100% Recognition? —", CBDAR2005.

[7] S. Uchida and H. Sakoe: "A survey of elastic matching techniques for handwritten character recognition", IEICE Trans. Info. & Syst., vol. E88-D, Accepted. (2005).

Recognition with Supplementary Information —How Many Bits Are Lacking for 100% Recognition?—

Masakazu IWAMURA*, Seiichi UCHIDA†, Shinichiro OMACHI‡, and Koichi KISE*
*{masa,kise}@cs.osakafu-u.ac.jp, Osaka Prefecture University, Sakai, Osaka, Japan
†uchida@is.kyushu-u.ac.jp, Kyushu University, Fukuoka, Japan
‡machi@aso.ecei.tohoku.ac.jp, Tohoku University, Sendai, Japan

Abstract

In this paper, we propose a new model in which the classifier receives not only a pattern itself but also supplementary information that assists recognition. This model enables us to achieve a 100% recognition rate with a 0% rejection rate with certain bits of supplementary information required. For printed characters, experiments show that 4 bits of supplementary information were required in the leave-one-out method and 1 bit was in the resubstitution method. In addition, we generalize the discussion into the relationship among a quantity of supplementary information, a recognition rate and a rejection rate. The theory presented in this paper is applied to the data embedding of a font set for camera-based character recognition [9].

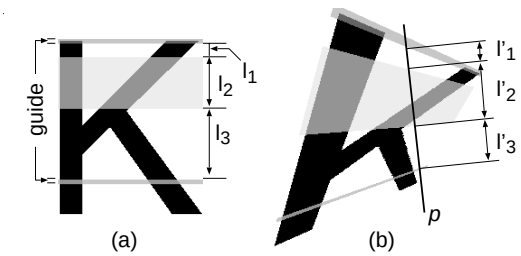


Figure 1. A data embedding method with cross ratio [9]. (a) A character image "K" where a cross ratio pattern is embedded. (b) Projective distortion.

1. Introduction

Let us imagine a task of recovering text data from printed papers without errors. It is not a good idea to employ an OCR undoubtingly since it cannot avoid causing recognition errors. How about using an extra media such as a 2D barcode for recording the whole text data? It is favorable regarding reading errors. However we cannot accept this solution because the original page is spoiled if a large barcode is next to the text (see the last page of this paper and the simulation in Sec. 5.2). It is just a great annoyance to us human beings though it is meaningful for computers.

Are there any solutions between these two extremes? It is true that the computer receives a certain amount of information from the recognition results of an OCR. Thus only the remaining information should be required for the retrieving of full text information. In this paper, such pattern recognition is considered.

In the setting of the problem, we have two channels: the first one is noisy (such as an OCR with recognition errors), and the second one is noiseless (such as a barcode). The problem here is how to design the second channel under the constraint that the amount of information sent through the second channel is minimized. We call the information on the second channel "supplementary information."

As an application of the supplementary information, we attempt a data embedding method with the cross ratio as in Fig. 1 for the task of document analysis and recognition with camera captured images [9].

We now turn to the real subject: how many bits of the supplementary information are required for a 100% recognition rate? We do not take rejection into account here. Let N be the number of classes. In the proposed model, K kinds of symbols, $K \leq N$, are assigned to the classes as the supplementary information. The K symbols has $\log_2 K$ bits of information. If N kinds of symbols are assigned, a 100% recognition rate with a 0% rejection rate (hereafter we call this "the state of 100% & 0%") is obviously achieved. However, $K = N$ is not always required since it is the same as transmitting a code without recognition as in Fig. 2(c). It is obvious that the better classi er performance becomes, the more K decreases. Thus quantity of information required to achieve "the state of 100% & 0%" is considered to depend

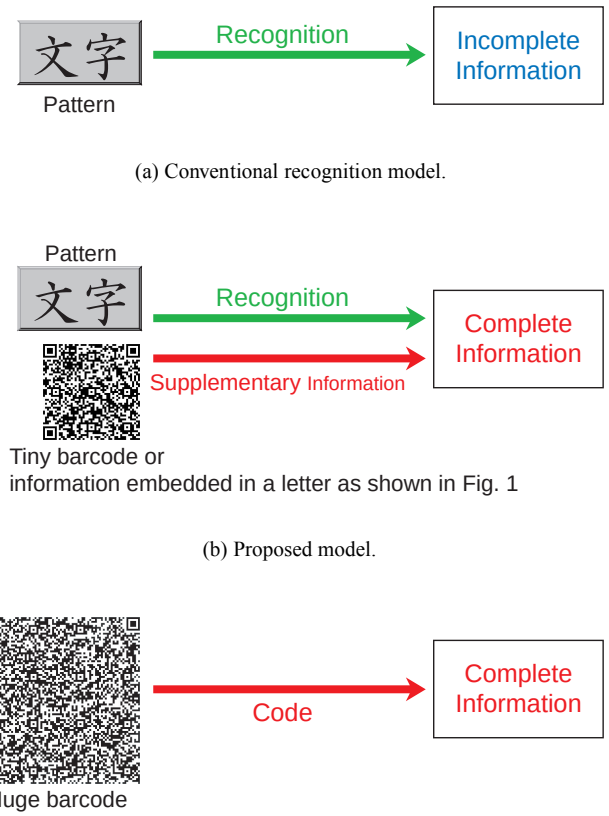


Figure 2. Recognition and data transmitting models.

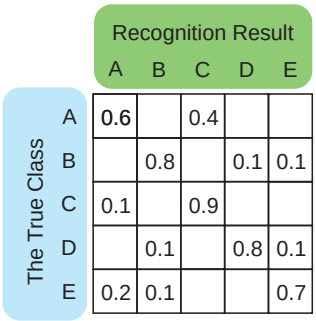


Figure 3. An example of a probabilistic confusion matrix. Empty elements represent 0.

on the performance of the classi er .

In this paper, we rst consider the condition that such supplementary information has to satisfy for achieving "the state of 100% & 0%" in Sec. 3. As a table which represents recognition performance of a classi er , we use a confusion matrix (CM) described in Sec. 2. For simplicity, supplementary information is assumed to be retrieved without errors.

As a similar idea of the supplementary information, there is the assist channel coding which is the key idea to improve recognition performance of an OCR with a scanner [5, 3, 2, 4]. The way of building the code is similar to that of assigning the supplementary information in Sec. 3.

Next, we generalize the discussion into the relationship among a quantity of supplementary information, a recognition rate and a rejection rate in Sec. 4. One of the most important points for a real use is the rejection technique. Therefore we also take rejection into account. There is also a case that "how much supplementary information is required to achieve a recognition rate higher than 95%?" Therefore we should consider not only a 100% recognition rate. As the quantity of supplementary information changes from 0 bits to $\log_2 N$ bits, the recognition rate and the rejection rate change. We investigate two kinds of relationships that (1) a quantity of supplementary information and a recognition rate without rejection, and (2) a quantity of supplementary information and a rejection rate with a 100% recognition rate. They are very useful to design a classi er with supplementary information.

In the experiments in Sec. 5, the relationship between quantity of supplementary information and recognition performance are investigated using real CMs, and a task to input a page of text into a computer without errors is simulated.

2. Confusion matrix and its probabilistic expression

2.1. Confusion matrix

A CM is a matrix representing the correspondence between true classes and recognition results. Let $\boldsymbol{C}$ be a CM of $N \times N$ matrix. Usually, the (i, j) element of $\boldsymbol{C}$ represents the number of occurrences where patterns of a class ω_i are recognized as those of a class ω_j.

2.2. Probabilistic confusion matrix

Let $\boldsymbol{W}$ be an $N \times N$ matrix whose (i, j) element represents the probability where a pattern of a class ω_i is recognized as one of a class ω_j, that is $P(\omega_j|\omega_i)$. The (i, j) element of $\boldsymbol{W}$ is calculated as $w_{ij} = \frac{c_{ij}}{C_i}$, where $C_i = \sum_{j=1}^{N} c_{ij}$ as in Fig. 3.

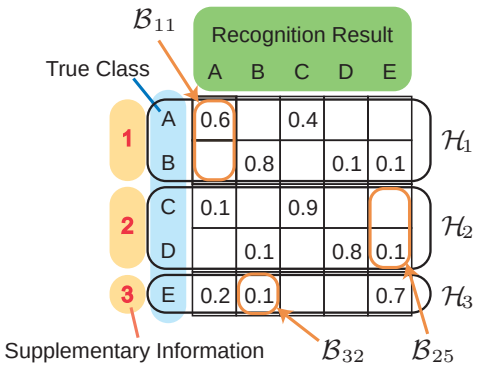


Figure 4. Symbols that achieve a 100% recognition rate with a 0% rejection rate. # of symbols:3, Recog. rate:100%, Reject. rate:0%.

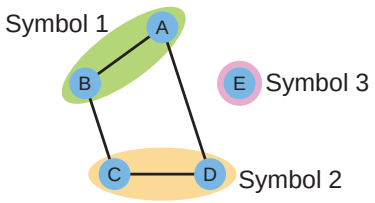


Figure 5. The division of the graph G which corresponds to Fig. 4. A symbol is assigned to each complete graph.

3. Supplementary Information that Achieves 100% Recognition Rate with 0% Rejection Rate

The proposed model in Fig. 2(b) can achieve a 100% recognition rate with a 0% rejection rate. The condition such supplementary information has to satisfy is derived and formulated with a graph. Hereafter the proposed classi er with supplementary information has the codebook of symbols, and the CM of probabilistic expression, that is the matrix $\boldsymbol{W}$. For simplicity, a priori probability of each class is assumed to be equiprobability. Namely, $P(\omega_i) = \frac{1}{N}$ is asumed.

3.1. Partition of matrix $\boldsymbol{W}$

In the proposed model, symbols are assigned to all the rows in the matrix $\boldsymbol{W}$. A combination of rows to which the k-th symbol is assigned is de ned as

$$\mathcal{H}_k = \left\{ l | l = l_1, \ldots, l_{|\mathcal{H}_k|} \right\}, \tag{1}$$

where $|\mathcal{H}_k|$ is the number of rows to which the k-th symbol is assigned. For example, $\mathcal{H}_1 = \{1, 2\}$, $\mathcal{H}_2 = \{3, 4\}$ and $\mathcal{H}_3 = \{5\}$ in Fig. 4. Note that rows in $\mathcal{H}_k$ are not necessarily continuous.

Next, $\mathcal{H}_k$ is partitioned into each column. A combination of elements in $\mathcal{H}_k$ and in the j-th column is de ne d as

$$\mathcal{B}_{kj} = \left\{ (l, j) \left| l = l_1, \ldots, l_{|\mathcal{H}_k|} \right. \right\}. \tag{2}$$

For example, $\mathcal{B}_{11} = \{(1, 1), (2, 1)\}$, $\mathcal{B}_{25} = \{(3, 5), (4, 5)\}$ and $\mathcal{B}_{32} = \{(5, 2)\}$ in Fig. 4.

3.2. Condition that supplementary information has to satisfy

With Fig. 3, we consider the condition that the supplementary information for "the state of 100% & 0%" has to satisfy. The gure shows that if a recognition result is the class A, the true class can be either the class A, C or E; the true class cannot be determined by the classi er . If the classi er outputs the class A, it will cause misclassi cation when the true class is either a class C or E. Therefore, supplementary information is required to distinguish the three classes. This means that at least three symbols are required here.

Similarly, if a recognition result is the class B, the true class can be either the class B, D or E. Thus, different three symbols are also required. Consequently, "the state of 100% & 0%" can be achieved when different three symbols are assigned to either "A and B," "C and D," and "E" as in Fig. 4 or "A and D," "B and C," and "E." Therefore the condition of the supplementary information which achieves "the state of 100% & 0%" is that "for all k and j, $\mathcal{B}_{kj}$ have less than two nonzero elements."

3.3 Problem of finding the smallest supplementary information

The problem of nding the smallest quantity of supplementary information is formulated with a graph as in Fig. 5. Let $G = (V, E)$ be a graph where V is a set which consists of N nodes each of which represents a class, and E is a set which consists of edges each of which links two nodes. An edge between two nodes is created if the same symbol can be assigned to the two corresponding classes without misclassi cation. After all such edges are created, the graph G is divided into complete graphs. A symbol is assigned to each complete graph. Thus the problem is to nd a division of the graph G where the number of complete graphs is the smallest. The problem formulated here is a minimization version of PARTITION INTO CLIQUES [1], which is NP-hard.

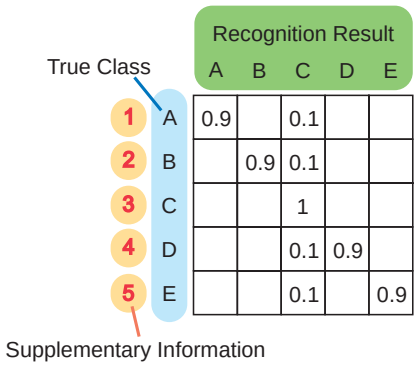


Figure 6. An example of a probabilistic confusion matrix which includes the class into which patterns are often misclassified. The samples of all classes can be recognized as those of the class C.

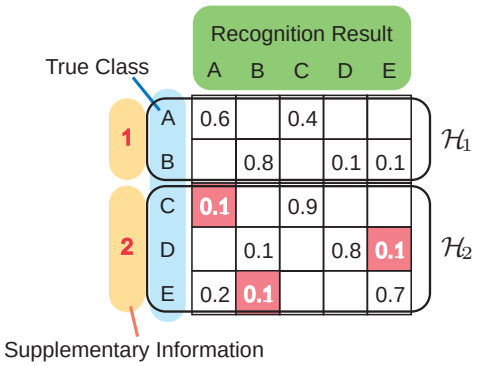


Figure 7. Symbols that accept recognition errors without rejection. The reverse colored elements are the causes of misclassification. # of symbols:2, Misclassi. Rate:0.3/5=6%, Recog. rate:94%, Reject. rate:0%.

Let N_{symmin} be the smallest number of complete graphs. Then, N_{symmin} symbols, which is $\log_2 N_{\text{symmin}}$ bits of information, can achieve "the state of 100% & 0%." N_{symmin} is determined by the largest number of non-zero elements in a column in the CM. To achieve "the state of 100% & 0%," no misclassi cation is permitted even if a classi cation result corresponds to many possible true classes according to the CM.

As the evaluation method of classi ers, the quantity of supplementary information has a different nature from the recognition rate. For example, in the case of the CM in Fig. 3, $N_{\text{symmin}} = 3$ and the recognition rate is 76%. In the case of the CM in Fig. 6, $N_{\text{symmin}} = 5$ and the recognition rate is 92%. This shows that a CM with higher recognition rate do not always require less information to achieve "the state of 100% & 0%." This is con rm ed in the experiment in Sec. 5.1.

4. Relationship Between Quantity of Supplementary Information and Recognition Performance

In Sec. 3, we discussed "the state of 100% & 0%." In this section, we generalize the discussion into the relationship between quantity of supplementary information and recognition performance. Actually, a quantity of supplementary information, a recognition rate and a rejection rate have a trade-off relationship where if two are determined, the rest is automatically determined. Here we focus on (1) relationship between a quantity of supplementary information and a recognition rate without rejection in Sec. 4.2, and (2) relationship between a quantity of supplementary information and a rejection rate with a 100% recognition rate in Sec. 4.3.

4.1. Number of nonzero elements in $\mathcal{B}_{kj}$

To generalize the discussion in Sec. 3, a function q_{kj} which returns the number of nonzero elements in $\mathcal{B}_{kj}$ is de ned. First, a function $z(x)$ is de ned as

$$z(x) = \begin{cases} 0, & \text{for} \quad x = 0 \\ 1, & \text{otherwise.} \end{cases} \tag{3}$$

Then, the function q_{kj} is de ned as

$$q_{kj} = \sum_{(l,j)\in\mathcal{B}_{kj}} z(w_{lj}). \tag{4}$$

4.2. Relationship between quantity of supplementary information and recognition rate without rejection

As mentioned in Sec. 3, if $\mathcal{B}_{kj}$ for all k and j has less than two nonzero elements, that is $\forall k, j \; q_{kj} \leq 1$, recognition errors do not occur. On the other hand, if $\mathcal{B}_{kj}$ has two or more nonzero elements for some k and j , that is $\exists k, j \; q_{kj} \geq 2$, recognition errors occur. In the case that $q_{kj} \geq 2$, there are q_{kj} possible true classes. The best way to minimize the recognition error is to choose *the most feasible class*. For example, in Fig. 7, if the recognition result is the class A and the symbol is 2, the class E should be chosen. The above consideration leads the formula of the

Algorithm 1 A greedy algorithm.

1: Assign different symbols to all classes. Namely, $\mathcal{H}_k = \{k\}$, for $k = 1, \ldots, N$.
2: Let $L(k)$ be a loss function which is either the misclassi cation rate or the rejection rate. Obviously, $L(N) = 0$.
3: **for** $K = N - 1$ to 1 **do**
4: To calculate the loss $L(K)$, the number of symbols is decreased by one. First, choose a pair of sets of rows that is assigned the same symbol. In other words, choose $\mathcal{H}_s$ and $\mathcal{H}_t$ which satis es $s \neq t$ and $\mathcal{H}_s, \mathcal{H}_t \neq \emptyset$. Then, the same symbol is assigned to $\mathcal{H}_s$ and $\mathcal{H}_t$. Namely,

$$\mathcal{H}_s \leftarrow \mathcal{H}_s \cup \mathcal{H}_t \tag{6}$$
$$\mathcal{H}_t = \emptyset. \tag{7}$$

5: **end for**

misclassi cation rate R_{error} as

$$R_{\text{error}} = \frac{1}{N} \sum_j \sum_k \left\{ \sum_{(l,j) \in \mathcal{B}_{kj}} w_{lj} - \max_{(l,j) \in \mathcal{B}_{kj}} w_{lj} \right\}. \tag{5}$$

In Eq. (5), the rst term in the parentheses is the sum of the elements of the matrix $\boldsymbol{W}$ in $\mathcal{B}_{kj}$, and the second term is the corresponding elements to the class where the classi er outputs.

In this paper, to minimize the misclassi cation rate or the rejection rate, we utilize a greedy algorithm shown in Algorithm 1. In the algorithm, as the number of symbols K decreases from N by one, the misclassi cation rate or the rejection rate is calculated with K symbols. Note that in the case of $K = 1$, it is the same as the usual pattern recognition method without supplementary information. The transition of the symbols and the misclassi cation rate R_{error} when Algorithm 1 is applied to the CM in Fig. 3 is shown in Fig. 8. Figs. 4 and 7 are the cases when the numbers of symbols are 3 and 2, respectively.

4.3. Relationship between quantity of supplementary information and rejection rate with 100% recognition rate

As mentioned in Sec. 3 and Sec. 4.2, if $\mathcal{B}_{kj}$ for all k and j has less than two nonzero elements, that is $\forall k, j\ q_{kj} \leq 1$, recognition errors do not occur. In the case that $\exists k, j\ q_{kj} \geq 2$, in order to achieve a 100% of recognition rate, all the possible true classes have to be rejected because they all can cause recognition errors. For example, in Fig. 9, if the recognition result is the class A and the symbol is 2, both the classes C and E are rejected.

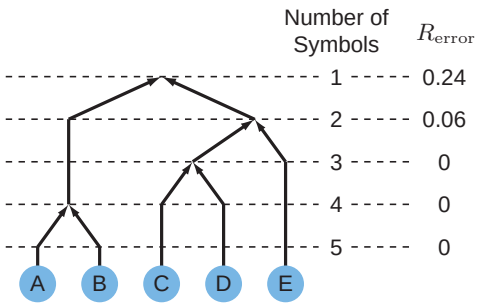


Figure 8. Transition of the symbols and the misclassification rate R_{error} with Algorithm. 1 for the confusion matrix in Fig. 3.

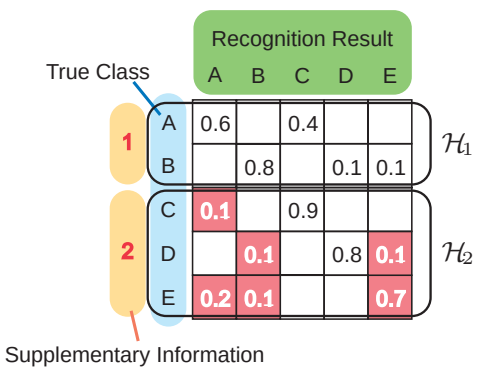


Figure 9. Symbols that achieve a 100% recognition rate by rejection. The reverse colored elements are the causes of rejection. # of symbols:2, Recog. rate:100%, Reject. rate:1.3/5=26%.

The above consideration leads the formula of the rejection rate R_{reject} as

$$R_{\text{reject}} = \frac{1}{N} \sum_j \sum_k s_{kj}, \tag{8}$$

where

$$s_{kj} = \begin{cases} 0, & \text{for} \quad q_{kj} \leq 1 \\ \displaystyle\sum_{(l,j) \in \mathcal{B}_{kj}} w_{lj}, & \text{otherwise.} \end{cases} \tag{9}$$

As in Sec. 4.2, the transition of the symbols and the rejection rate R_{reject} when Algorithm 1 is applied to the CM in Fig. 3 is shown in Fig. 10. Figs. 4 and 9 are the cases when the numbers of symbols are 3 and 2, respectively.

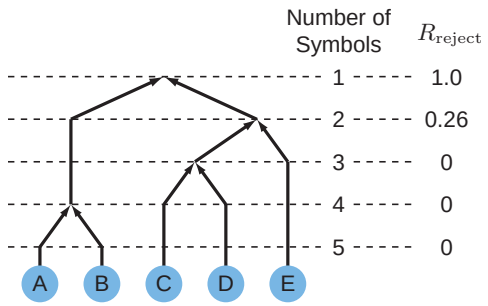


Figure 10. Transition of the symbols and the rejection rate R_{reject} with Algorithm. 1 for the confusion matrix in Fig. 3.

5. Experiments

5.1. Relationship between quantity of supplementary information and recognition performance

In the experiments, eight kinds of CMs which were created by recognition of real data sets were used. For the eight CMs, two kinds of data sets, two kinds of classi ers, and two kinds of recognition experiments, L and R methods, were combined. The L method is the leave-one-out method and the R method is the resubstitution method.

As the two kinds of data sets, handwritten and printed characters were used. Note that embedding supplementary information in handwritten characters is not easy; we used them just for the demonstration of the proposed method. As the handwritten data set, the ETL9B [7], which consists of 3036 Japanese characters and each character has 200 samples, was used. As the printed one, 25 fonts were used. The same 3036 characters as contained in ETL9B were extracted from the data set of printed characters. Each character image in both data sets was normalized nonlinearly [10] to t in a 64×64 square, and the 196-dimensional directional element feature [8] was extracted. The Euclidean distance and the SQDF [6] were used in classi ers. The conditions of recognition experiments and the corresponding recognition rates are shown in Table 1.

For the eight CMs, relationship between the number of symbols (the quantity of supplementary information) and the recognition rate mentioned in Sec. 4.2, and that between the number of symbols (the quantity of supplementary information) and the rejection rate mentioned in Sec. 4.3 are shown in Figs. 11 and 12. From the gures, we can estimate how much information is required to achieve a certain recognition rate or rejection rate. This information is useful to design a classi er in certain performance. Note that the values in the gur es are not the best values and possible recognition performance may be better.

In Table 1, the quantity of supplementary information required to achieve "the state of 100% & 0%" is also shown. The uncertainty of 3036-class problem is $\log_2 3036 \sim 11.57$ bits. For printed characters, the experiments show that 4 bits were required in the leave-one-out (L) method and 1 bit was in the resubstitution (R) method. Though the greedy algorithm does not guaranteed to give the ideal value, quantity of supplementary information required to achieve "the state of 100% & 0%" in the table was conrmed to be the same as the ideal one.

By comparison with the Euclidean distance, the SQDF had an advantage of the recognition rate. However, it did not always have an advantage of the quantity of supplementary information required to "the state of 100% & 0%." This is due to a few classes into which patterns are often misclassi ed as mentioned in Sec. 3.3. Therefore, a classi er which minimizes the quantity of supplementary information should be developed.

5.2. Comparison to transmitting character code

We have explained that the proposed model in Fig. 2(b) combines advantages of the conventional approach of pattern recognition in Fig. 2(a) and transmitting identi cation codes in Fig. 2(c). The proposed model is valuable when patterns are available but recognition errors occur. For example, let us assume a task to input a page of text into a computer without errors. If you have a media that can contain all the information, the problem is solved. However, when you use a printable media such as a 2D barcode, you will realize another problem occurs: it requires large area. On the other hand, in the proposed model a smaller 2D barcode can reduce the area. This is con r med by a simulation using the QR code (ISO/IEC 18004). We assumed a page contains 1000 Japanese characters in the simulation.

There are 40 variations of the QR code which differ in printed area and containable data size. Since 12 bits are required for a code of 3036 categories,

$$12(\text{bits}) \times 1000(\text{characters})/8 = 1500(\text{bytes})$$

of supplementary information are required per page. On the other hand, since 1 bit is required for the proposed method,

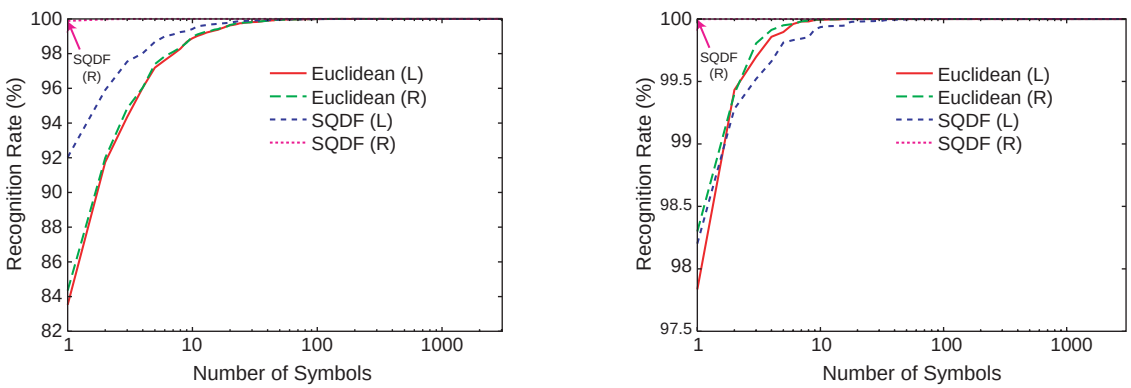
$$1(\text{bit}) \times 1000(\text{characters})/8 = 125(\text{bytes})$$

are required per page.

In the binary mode with the error correction level M, the version 8 and 32 are the smallest ones whose capacities are more than 125 and 1500 bytes, respectively. The QR codes of version 8 and 32 are shown in Fig. 13. In comparison to version 32, the QR code of version 8 is about one third in

Table 1. Conditions of recognition experiments, recognition rates and quantity of information required to achieve a 100% recognition rate with a 100% recognition rate ("the state of 100% & 0%").

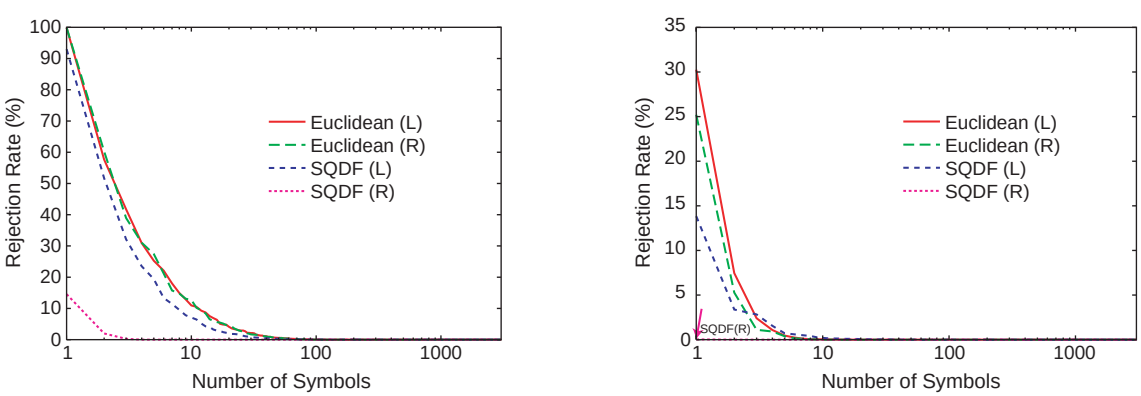
Data set	Distance in Classi er	L / R method	Recognition rate (%)	Quantity of supple. info. required to "the state of 100% & 0%" (Bits) (# of symbols required to "the state of 100% & 0%" in parentheses)
Handwritten Characters	Euclidean	L	83.53	7.69 (206)
		R	84.35	7.69 (206)
	SQDF	L	92.03	8.84 (459)
		R	99.89	2.00 (4)
Printed Characters	Euclidean	L	97.84	4.00 (16)
		R	98.30	3.91 (15)
	SQDF	L	98.20	6.15 (71)
		R	99.99	1.00 (2)



(a) Handwritten characters.

(b) Printed characters.

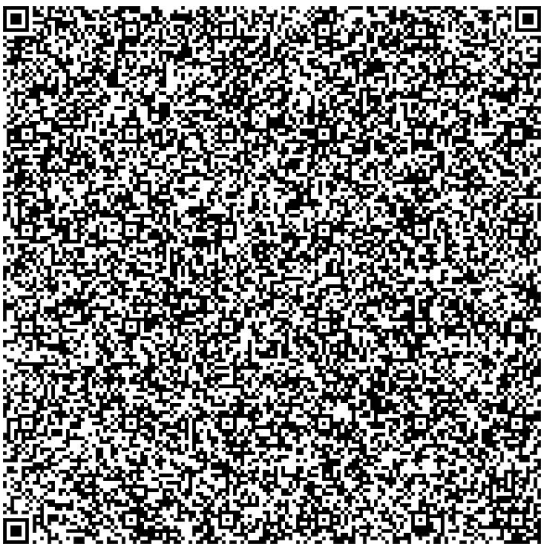
Figure 11. Relationship between the number of symbols and the recognition rate without rejection.



(a) Handwritten characters.

(b) Printed characters.

Figure 12. Relationship between the number of symbols and the rejection rate with a 100% recognition rate.



(a) Version 32. This equals the character code.

(b) Version 8. This equals the supplementary information of the proposed method.

Figure 13. QR code required to represent a page of Japanese text.

width and height, and about one ninth in area. It is demonstrated that the proposed model can reduce the quantity of information required. Consequently it is con rmed that the proposed model is ef cient in such situations in comparison to a barcode containing the whole text information as in Fig. 2(c).

6. Conclusion

In this paper, we proposed a new model that the classi er receives not only a pattern itself but also supplementary information that assists recognition. In the model, we can achieve a 100% recognition rate with a 0% rejection rate ("the state of 100% & 0%"). For printed characters, experiments showed that 4 bits were required in the leave-one-out method and 1 bit was in the resubstitution method.

For the eight real CMs, (1) relationship between a quantity of supplementary information and a recognition rate, and (2) that between a quantity of supplementary information and a rejection rate were observed. These information is useful to design a classi er using supplementary information in certain performance.

Furthermore, we demonstrated that the quantity of supplementary information required to achieve "the state of 100% & 0%" has a different nature from the recognition rate; the quantity of supplementary information is a new evaluation criterion of classi ers. In this sense, we have to design a classi er which minimizes the supplementary information.

Acknowledgement Thanks are due to an anonymous reviewer for valuable comments on the assist channel coding.

References

[1] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman and Company, New York, 1979.

[2] D. H. Greene and E. H. Kuo. Assist channel coding with vertical block error correction. United States Patent 6,768,560.

[3] D. H. Greene, L. T. Niles, and E. H. Kuo. Assist channel coding with character classi cations. United States Patent 6,862,113.

[4] D. H. Greene and A. C. Popat. Assist channel coding with convolution coding. United States Patent 6,628,837.

[5] E. H. Kuo. Assist channel coding for improving optical character recognition. Master's thesis, MIT, 2000.

[6] S. Omachi, F. Sun, and H. Aso. A new approximation method of the quadratic discriminant function. *Lecture Notes in Computer Science*, 1876:601–610, Sept. 2000.

[7] T. Saito, H. Yamada, and K. Yamamoto. On the data base ETL9 of handprinted characters in JIS Chinese characters and its analysis. *Trans. IEICE*, J68-D(4):757–764, 1985. Written in Japanese.

[8] N. Sun, Y. Uchiyama, H. Ichimura, H. Aso, and M. Kimura. Intelligent recognition of characters using associative matching technique. In *Proc. Pacific Rim Int'l Conf. Artificial Intelligence (PRICAI'90)*, pages 546–551, Nov. 1990.

[9] S. Uchida, M. Iwamura, S. Omachi, and K. Kise. Data embedding for camera-based character recognition. In *Proc. First International Workshop on Camera-Based Document Analysis and Recognition*, Aug. 2005.

[10] H. Yamada, K. Yamamoto, and T. Saito. A nonlinear normalization method for handprinted kanji character recognition — line density equalization —. *Pattern Recognition*, 23:1023–1029, 1990.

Section III

Systems

Oblivious Document Capture and Real-Time Retrieval

Christoph H. Lampert[†], Tim Braun[*], Adrian Ulges[*], Daniel Keysers[†], Thomas M. Breuel[†]

[†]German Research Center for Artificial Intelligence (DFKI), 67608 Kaiserslautern, Germany
[*]University of Kaiserslautern, 67663 Kaiserslautern, Germany
{chl, braun, ulges, keysers, tmb}@iupr.net

Abstract

Ever since text processors became popular, users have dreamt of handling documents printed on paper as comfortably as electronic ones, with full text search typically appearing very close to the top of the wish list.

This paper presents the design of a prototype system that takes a step into this direction. The user's desktop is continuously monitored and of each detected document a high resolution snapshot is taken using a digital camera. The resulting image is processed using specially designed dewarping and OCR algorithms, making a digital and fully searchable version of the document available to the user in real-time. These steps are performed without any user interaction. This enables the system to run as a background task without disturbing the user in her work, while at the same time offering electronic access to all paper documents that have been present on the desktop during the uptime of the system.

1 Introduction

For capturing images of documents, digital cameras have many advantages over other commonly used devices like flatbed scanners. Most prominent among these advantages are their small physical dimensions and their ability to work in real-time and without physical contact at a distance. For many applications, these advantages outweigh possible disadvantages like lower image resolution and perspective distortions in the images. Another major point in favor of the use of digital cameras for document capture is their wide availability, which still growing, for example with the use of cameras built into mobile phones. Thus, if today there are hardly any products that make use of cameras for document image capture, this is largely not due to limitations of the hardware, but to missing software support.

Our aim with this paper is to bring camera-based document capture closer towards an actually useful product, by presenting a prototype system that relies on what we consider the biggest advantage of cameras for document capture: the ability to work completely without physical user interaction, and therefore without interrupting the user's everyday work-flow.

Most users will agree that it would often be very useful to have all their documents available in electronic form, if only to allow fast distribution by email or for full text search. However, it is unrealistic to hope that paper-based documents will disappear in the near or middle future, since paper offers too many advantages, e.g. excellent readability, low cost, and a basis for handwritten annotations or authentification marks like signatures and stamps.

Our setup therefore does not try to replace printed documents, but to enrich them by creating an additional digital version of each printed document that a user is studying during the day, that is, of each document that appears on his desktop. By saving, processing, and indexing all documents obtained that way, the user is provided with a digital archive of all the documents that ever crossed his desk.

To achieve this ambitious goal we let a digital camera constantly monitor the user's desktop in low resolution and trigger a high resolution capturing process whenever a document is detected. The resulting image is automatically processed, freed of typical distortions, and its textual content is extracted and stored along with the image itself. All this happens in a background process on a PC and no user interaction is required such that the user is never disturbed. If, however, the user wants to access such a document in its digital form at some time, he can do so by using a simple GUI for full text and meta search.

Even though there is a vast amount of literature on camera-based document capture and desktop surveillance, we are not aware of previous systems that aim in the same direction as we propose. However, we would like to mention the CamWorks system [10] that also targets real-time digital access to printed source, but concentrates on cut-and-paste operations using a low resolution video camera. Typical described uses for systems observing user desktops

Figure 1. The complete setup: a wooden desk with two cameras attached and a laptop PC running the software components.

Figure 2. The camera section of the setup in more detail.

are not targeted at the automatic capture of document images, but at extracting user behavior from objects and gestures (e.g. [8]) or at creating 'active areas' on a desktop, e.g. providing specialized functionality like a desktop calculator [17].

2 Hardware

To build a prototype system that mainly demonstrates the functionality of the software, we have abstained from using any special hardware devices, like infrared light or mechanical sensors. Instead, we rely on standard components only. These are two *Canon Powershot S50* digital cameras with a resolution of 5 megapixels each and a laptop PC. The cameras are mounted to a base plane that is attached to the rear panel of a wooden working desk, and a data connection from the cameras to the laptop is established using standard USB cables. Figures 1 and 2 show this setup. All further functionality is provided by software components as described in the following section. Note that our choice of two cameras is motivated only by one of the dewarping algorithms relying on stereo vision (see Section 3.3). A setup with only one camera or with more than two cameras would be just as feasible.

3 Software

Our software is designed as a set of independent modules. In the following sections, we will give overviews of each of the different software components.

3.1 Surveillance

Each digital camera can continuously send up to 25 low resolution viewfinder images per second via the USB interface to the PC. For our setup we analyze five input images per second to determine whether a new document has been placed within the field of view.

In the following, the employed document detection method will be introduced. A more detailed description of the approach can be found in [3].

The document detection method consists of two stages. The first stage extracts relevant features from the input image, while the second stage validates these features to discriminate between documents and other visible objects. Because nearly all printed documents are roughly rectangular, we use this a-priori information by using the outline of the object as the main feature to detect printed documents. Therefore, the extraction stage determines the contour outlines of all objects that enter the view volume of the camera. The subsequent validation stage then checks whether the outlines are approximately rectangular and rejects all non-matching contours. Valid outlines that are detected in a number of subsequent frames are accepted as a document; the corresponding image position is then passed to the document capture module which will be described in Section 3.2. In the following, the two stages of the document detection method are presented in more detail.

Feature Extraction. The feature extraction stage is subdivided into four image processing steps. The first step compensates the radial distortion present in the input images, which would otherwise deform the outlines of all visible objects.

Secondly, static background visible in the input image is removed using a standard 'image difference' background subtraction method, where a previously recorded background model without visible documents is subtracted from the current input image. The background model is recorded once during the setup of the system and then continuously updated to account for changing illumination. The absolute difference image between frame and background is thresholded to produce a binary image with highlighted foreground regions. To achieve some invariance to changes in illumination, the thresholding operation is performed only in the chrominance channels of the YUV color space.

The third processing step takes the generated binary image, performs morphological smoothing and then extracts the contours of all foreground regions using the hierarchical boundary detection algorithm presented by Suzuki and Abe [13]. With this algorithm, the outer boundary of every foreground object is found, while inner contours are ignored.

To decrease the runtime of the following validation step, each extracted contour is simplified in the fourth and final feature extraction step. This simplification is done with the Douglas-Peucker algorithm [7], which approximates contours by polygons of an adjustable number of edges.

Feature Validation. The simplified contours of foreground objects are examined to locate rectangular documents. To accomplish this in a noise-robust way, the feature validation stage first fits lines to each detected contour. Then, the algorithm checks whether the detected lines are pairwise orthogonal and satisfy several additional constraints. Since the line fitting effectively smoothes the object contours and tolerates small undetected or occluded parts, this process is able to robustly detect rectangular shapes even when the extracted outlines are degraded.

Lines are fit to each contour using the robust branch-and-bound algorithm RAST [4]. RAST can be employed to detect prominent linear structures in a similar fashion as the well known Hough transform. However, RAST does not suffer from quantization effects, is computationally more efficient, and can explicitly use short line segments as primitives instead of single pixels. This makes RAST especially suited for the presented system, because the contour extraction/simplification stage outputs short line segments, i.e. the elements of the polygon approximation to the contour. Since there are much fewer line segments than single points, the RAST algorithm gains an additional speedup in comparison to the Hough transform. This is of special importance in the presented system, because the document detection must run in real-time.

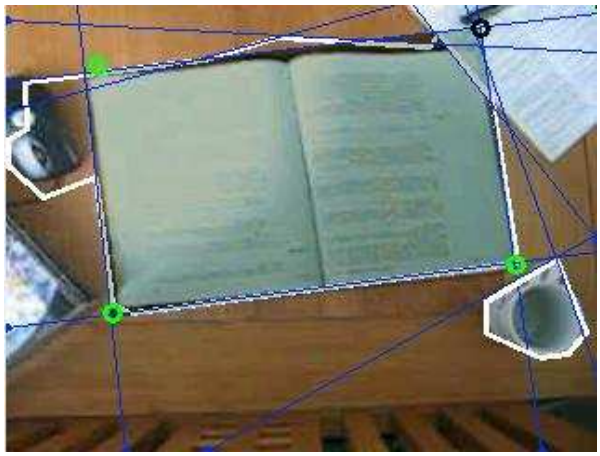


Figure 3. Internal debug image during the validation phase with highlighted contour lines. Edges and corners of a roughly rectangular document object are marked.

Before the fitted lines can be checked for orthogonal intersections, the perspective distortion caused by the particular placement of the cameras must be removed. To do so, a corrective projection is computed in an extrinsic calibration stage during system setup, in which an A4-sized sheet of paper is placed on the desktop and then used as a calibration object.

After applying the corrective projection to the fitted lines, the angles between them are identical to those between the corresponding contour elements in the real world. Thus, parallel lines correspond to parallel contour edges of an object in the camera field of view. To finally detect rectangular objects, parallel lines are collected into groups, and candidates for rectangular objects are generated from these groups by checking whether two groups are orthogonal to each other within a tolerance threshold. If this structural validation is true for any two groups, the largest rectangle that can be formed using lines contained in the two groups is determined and its four corner positions are calculated.

A rectangular object is accepted as a document only if the four corresponding corners are detected across several camera frames. We use a clustering of detected corner positions over time. The threshold for acceptance of a region as a document is then based on the number of corner positions in each cluster.

Whenever a document has been successfully detected using the described process, the actual capture process is initiated. Figure 3 shows an internal debug image of the system in which detected lines and corner points of a document are shown.

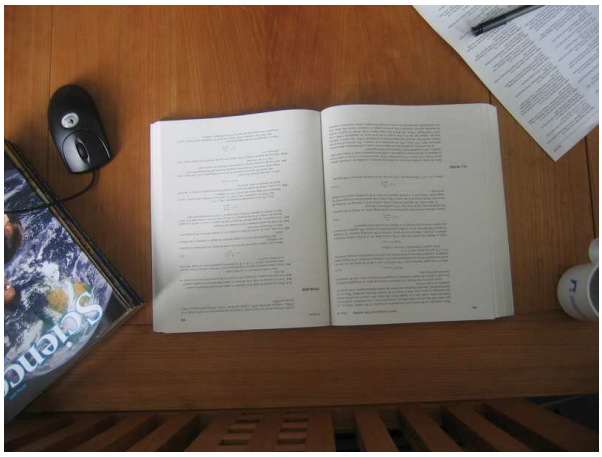
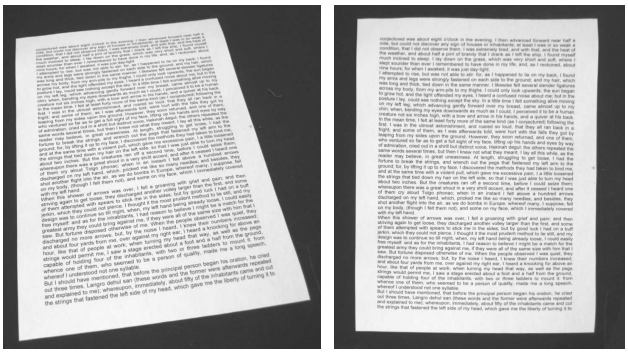


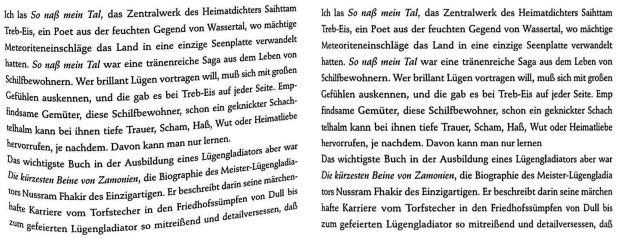
Figure 4. Captured high resolution image of the desktop, including a document.



(a) The captured image of a single sheet of paper. Before...

(b) ...and after perspective correction.

Figure 5. Removal of perspective distortion for a planar document.



(a) A captured image of a curled book surface. Before...

(b) ...and after the dewarping process.

Figure 6. Removal of geometric distortion using the straightening of text lines.

3.2 Capture

Each camera can capture a full resolution image (2614×1958 pixels) of the desktop, see Figure 4. The capture process itself is controlled by the free software library *libptp*[1], that allows to remotely control digital cameras compatible with the *Picture Transfer Protocol (PTP)* [1] via the USB interface. Because the surface of a desktop is rather large compared to the resolution of the digital camera, the resulting images have a resolution of only approximately 100 dpi for the document, which is close to the lower bound of what is useful for document capture. For a production system, more or higher resolution cameras should be considered.

3.3 Content Extraction

Preprocessing. The captured image shows the whole desktop and is first cropped to only contain the detected document itself. The coordinates of the corners of the document that were obtained during the surveillance step are reused for this. Cropping reduces the image size, usually by a factor of 4 to 8, thus allowing faster processing.

When capturing images with a camera, a perspective distortion occurs, which causes more distant parts of the image to appear smaller. The most severe effect of this distortion is that parallel lines in the real world do not appear parallel anymore in the image. However, since we have calibrated our cameras in advance it is easy to correct for this perspective distortion. At the same time we rotate the rectangular image object into an upright position, see Figure 5.

[1] http://sourceforge.net/projects/libptp

OCR. In the next step, we want to extract the textual information from the document by performing OCR. This is not a trivial task in this setting, because—in contrast to images obtained from flatbed scanners—document images captured by cameras often contain distortions even after perspective correction. The main reason for these remaining distortions is that the documents themselves are not necessarily planar but the pages may contain an inherent curl. This causes e.g. text lines on a book page not to be straight in the captured image, see Figure 6(a). Several methods for the removal of such effects have been proposed, and we will discuss some of them in more detail in the following section. However, to our knowledge none of them is capable of working in real-time, which would be necessary in our setting, because the user might want to access a document immediately after it has been captured.

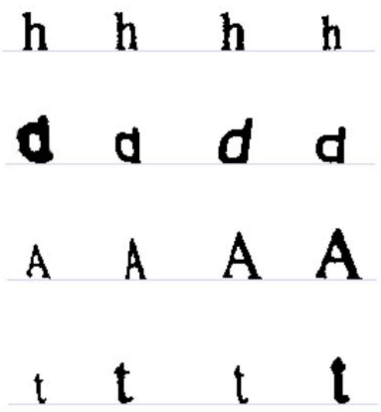


Figure 7. Typical distortions of characters within the same document as caused by non-linear warping of the document image and noise.

We therefore have chosen the following new approach: we use a fast OCR step that is specially designed to work well for text showing a certain amount of geometric distortion. Typically distorted characters are shown in Figure 7. Single characters are not only affected by the non-linear warping of the document image, but the binarized characters are also affected by low resolution, varying illumination, and noise.

To perform OCR robust to distortion, we train an artificial neural network with images of letters showing the same geometric distortion that occurs when taking pictures of non-planar pages. To reliably produce a large amount of training images they were created artificially using the FreeType typesetting library[2]. Noise and jitter were imitated using Baird's defect model [2]. In addition, the letters were scaled to several widths to simulate the horizontal and vertical shortening occurring when the page surface is not observed straight from above. Similar approaches are used for the recognition of handwritten digits, where the variability of the data is of a different type and for example simulated using smoothed random two-dimensional distortions [12].

Apart from the neural network, also the other steps of the OCR system are designed to work with non planar documents. For example, the tracking of text lines can only be done locally linear, since those are not straight along the whole image anymore. The RAST line finding algorithm [4] was adapted for these special requirements. Also, even on the scale of grouping letters into words, the curl of the base line can be significant, causing e.g. projection profiles or smearing approaches to fail. Instead, a robust clustering approach was used here. A detailed description of the OCR algorithm is given in [14].

[2]http://www.freetype.org

Image Flattening. Ultimately, the goal is of course to obtain document images without any distortion by flattening the image. This makes the image more pleasing to the human eye, see Figure 6(b), and also much easier to process with an OCR system.

Note that the flattening step is optional in our approach and the real-time system is based on directly processing the captured and preprocessed image without flattening. However, image flattening can be used to improve the results as a background task, while already preliminary OCR results are available for real-time search.

Many approaches have been proposed for image flattening, often estimating the 3D shape of the document surface using additional hardware like structured light sources [5], laser scanners [11] or a second camera [19]. In the latter case, printed text and graphics are exploited as texture for a stereo algorithm from which the 3D shape is reconstructed. Other approaches work without an explicit page model, but require additional assumptions with respect to the page content, like the existence of parallel horizontal text lines [6, 9, 18].

In our system, we have chosen to integrate two such methods as optional modules. The *straightlines* algorithm from [16] has the advantage that it essentially can reuse much of the information that has been obtained during an initial OCR step. Local estimates of the text line slope and the base line distance are used to determine the tilt angle of the surface and its distance to the camera on a mesh of points on the book surface. From this information, a scale factor for each letter is derived and the letters are warped one-by-one onto a rectangular grid of output image cells. The method works well if the documents contains regular text lines with fixed line spacing, see Figure 6(b). However, in its current state the algorithm is limited to single column documents containing only text.

The second method is computationally more expensive, but has the advantage of being able to process pages of arbitrary content, including e.g. handwritten text or illustrations. It generates a 3D page model using two camera images and stereo vision. The resulting 2D surface in 3D space is then flattened by inscribing a regular mesh of equilateral triangles which is conformally mapped onto a planar mesh and textured with patches from the camera images. For details on the stereo dewarping algorithm, see [15].

Since both methods have their own advantages, it is useful to let both try to dewarp the captured image. However, this needs large computing resources, and so we start them as background threads, working at a low CPU priority. That way, only when the system would otherwise be idle, the de-

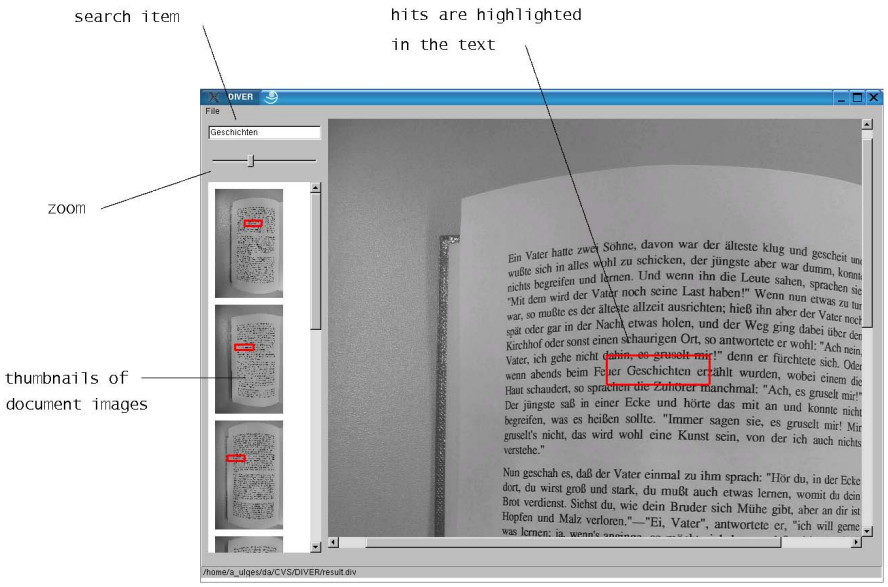


Figure 8. A screen-shot of the simple search user interface. Search strings can be entered, and matches in the document images are highlighted using colored boxes.

warping routines are called, and the system does not lose its ability to capture images and answer queries in real-time.

3.4 Archiving

Directly after the image capture and the initial OCR process, a digital image and the extracted text of the document are available. The text and the image are saved together along with a time stamp to the hard disk. To keep the prototype simple, the data is simply stored as individual files instead of database entries. For the text a simple ASCII format containing the actual words and their position within the document image is used. The images themselves are stored in JPEG, PNG, or DjVU format. Afterwards, the system is ready to capture the next document.

Whenever a background dewarping process for an image is finished, the flattened document image is saved to the same folder in which the original is located. Additionally, a second OCR process could be started now to improve the results of the initial OCR pass.

3.5 Retrieval

As we have emphasized before, the crucial part in oblivious document capture is that no user interaction is required at runtime. The system works continuously in the background, even if the PC happens to be used for other tasks at the same time. However, the user of course also needs a possibility to access the archive of captured documents. We have designed a very simple prototype user GUI for this. Assuming that the biggest advantages of electronic versus paper documents is their capability for fast and flexible search, we concentrate on this aspect here.

In the initial version, a simple but effective full text search is implemented with the additional possibility to specify an interval for date and time of capture to which the search is limited. After a successful query, all documents containing the search expression are presented in thumbnailed image form, additionally highlighting the search string using a colored box. Figure 8 shows a screen-shot of the simple search interface. We chose to present the image version and not the text version of the document to the user, because it might contain more information than the OCR had been able to extract, e.g. images or handwritten remarks.

Note that in a production system, special more emphasis would be placed on tailoring the text search to the special needs encountered here. Since we know that the text resulted from an OCR process, we would use a word similarity measure that takes into account possible errors and enables us to weight various mismatches. For example, the confusion of the lower-case letter ‘l’ with an upper-case letter ‘I’ would receive less penalty than other mismatches in the string matching.

Once a document has been selected for a larger view, it is possible to navigate to those which were captured directly

before or afterwards, using ‘forward’ and ‘backward’ buttons that users are familiar with from their web browser. This makes is possible to access all pages of a multi-page document in the same order they were read the first time the user had them on his desk.

4 Summary

We have described a prototype system that can perform document capture and archiving without any user interaction. The user’s desktop is monitored and whenever a new document (here, a roughly rectangular and reasonably sized object) is detected, a high resolution image is captured. The textual content of the document is extracted using a special OCR component that can deal with the distortion that camera captured document images typically show. In addition, background threads are started to create an image version free of these distortions. Both the image and the textual content are archived, ready to be accessed by the user at any later time using a customized GUI application that allows full text search.

The resulting system is able to support every day office work-flow without interrupting it by taking a step into the direction of full text search for documents printed on paper.

Apart from its functionality, the main advantage of the proposed system is its modular setup. Each step described in Section 3 is designed as a separate module that can easily be replaced by an improved one. Also, more dewarping modules than the two implemented so far can be integrated, e.g. specializing on different classes of document types.

5 Current Status and Future Work

At the moment, the system is still in the development stage. All individual modules were implemented and tested independently. The connecting data interfaces will be aligned soon, such that the system can then work completely without user interaction.

The system was designed to be a proof-of-concept. Therefore, some improvements will be necessary before it can be considered to be of practical use. The first issue is that the resolution of the digital cameras chosen seems not high enough. At least 200 dpi at document level are desirable, even better 300 dpi, and the current system cannot yield more than 100 dpi when monitoring a full desktop surface. Another practical problem is that the delay between triggering a capture and actually obtaining the data currently is in the range of a few seconds and therefore somewhat too large. This is mainly due to slow camera action and data transfer, partly also caused by the *Canon Powershot* cameras still using the old USB1.1 standard. Switching to newer and higher resolution cameras is therefore a high priority on our list of improvements.

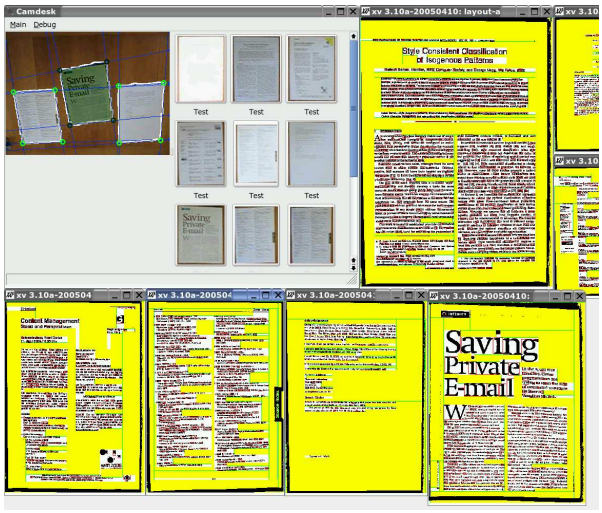


Figure 9. Screen-shot of a debugging output showing various captured documents that have undergone a first layout analysis step, i.e. the detection of white space.

On the software side, the current surveillance module should be made aware of the document content instead of only the shape of its outline. That way, a document that has already been captured would not have to be captured again, and a new document that is found at exactly the same position as the previous one (e.g. different pages of a book) will be detected more reliably.

We also plan to address another use of the document database that results from the captured images, i.e. content-based access. We will integrate both image-based and text-based retrieval of similar documents in addition to the keyword search already possible.

The field of document image dewarping is a very active field of research, and we plan to improve our algorithms in that field as well. Our special aim is to make the approach using only one camera better suitable for complex documents that contain more than just text. This will in particular require a module for separating text from images and for layout analysis, both of which are currently under development. Figure 9 shows a screen-shot of the results of a first layout analysis step, the detection of white space in the document images.

Acknowledgments

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project IPeT (01 IW D03).

References

[1] American National Standards Institute. Electronic still picture imaging - Picture Transfer Protocol (PTP) for Digital Still Photography Devices. ANSI/PIMA 15740:2000.

[2] H. S. Baird. Document Image Defect Models. *Document Image Analyis*, pages 315–325, 1995.

[3] T. Braun. Camdesk - Towards Easy and Portable Document Capture. Technical Report, Robotic Systems Group / IUPR, Technical University Kaiserslautern, Germany, 2005.

[4] T. M. Breuel. Robust Least Square Baseline Finding using a Branch and Bound Algorithm. In *Proc. of the SPIE – The Int. Society for Optical Engineering*, pages 20–27, 2002.

[5] M. S. Brown and W. B. Seales. Document Restoration Using 3D Shape: A General Deskewing Algorithm for Arbitrarily Warped Documents. In *Int. Conf. on Computer Vision (ICCV01)*, volume 2, pages 367–374, Vancouver, Canada, July 2001.

[6] H. Cao, X. Ding, and C. Liu. Rectifying the Bound Document Image Captured by the Camera: A Model Based Approach. In *7th Int. Conf. on Document Analysis and Recognition (ICDAR2003)*, pages 71–75, Edinburgh, UK, August 2003.

[7] D. H. Douglas and T. P. Peucker. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Caricature. *The Canadian Cartographer*, 10(2), 1973.

[8] T. Kawashima, T. Nakagawa, T. Miyazaki, and Y. Aoki. Desktop Scene Analysis for Document Management System. In *10th Int. Workshop on Database & Expert Systems Applications (DEXA '99)*, pages 544–548, Washington, DC, 1999.

[9] J. Liang, D. DeMenthon, and D. Doermann. Flattening Curved Documents in Images. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR2005)*, volume II, pages 338–345, June 2005.

[10] W. Newman, C. Dance, A. Taylor, S. Taylor, M. Taylor, and T. Aldhous. CamWorks: A Video-based Tool for Efficient Capture from Paper Source Documents. In *IEEE Int. Conf. on Multimedia Computing and Systems*, volume 2, pages 647–653, June 1999.

[11] M. Pilu. Undoing Page Curl Distortion Using Applicable Surfaces. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR2001)*, pages 67–72, Kauai, HI, December 2001.

[12] P. Simard. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. In *7th Int. Conf. Document Analysis and Recognition*, pages 958–962, Edinburgh, UK, August 2003.

[13] S. Suzuki and K. Abe. Topological Structural Analysis of Digital Binary Images by Border Following. *Computer Vision, Graphics, and Image Processing (CVGIP)*, 30(1):32–46, 1985.

[14] A. Ulges. Indexing and Recognition of Documents Captured with a Handheld Camera. Master's thesis, Technical University Kaiserslautern, Germany, 2005.

[15] A. Ulges, C. H. Lampert, and T. M. Breuel. Document Capture using Stereo Vision. In *Proc. of the ACM Symposium on Document Engineering*, pages 198–200, Milwaukee, WI, 2004.

[16] A. Ulges, C. H. Lampert, and T. M. Breuel. Document Image Dewarping using Robust Estimation of Curled Text Lines. In *8th Int. Conf. on Computer Vision and Pattern Recognition (ICDAR2005)*, Seoul, South Korea, August 2005. In press.

[17] P. Wellner. Interacting with Paper on the DigitalDesk. *Commun. ACM*, 36(7):87–96, 1993.

[18] C. Wu and G. Agam. Document Image Dewarping for Text/Graphics Recognition. In *SPR 2002, Int. Workshop on Statistical and Structural Pattern Recognition*, volume 2396 of *Lecture Notes in Computer Science*, pages 348–357, Windsor, Ontario, Canada, August 2002.

[19] A. Yamashita, A. Kawarago, T. Kaneko, and K. T. Miura. Shape Reconstruction and Image Restoration for Non-Flat Surfaces of Documents with a Stereo Vision System. In *17th Int. Conf. on Pattern Recognition (ICPR2004)*, volume 1, pages 482–485, Cambridge, UK, 2004.

Hashing with Local Combinations of Feature Points and Its Application to Camera-Based Document Image Retrieval

— Retrieval in 0.14 Second from 10,000 Pages —

Tomohiro Nakai, Koichi Kise, Masakazu Iwamura
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka, 599-8531 Japan
nakai@m.cs.osakafu-u.ac.jp, {kise, masa}@cs.osakafu-u.ac.jp

Abstract

This paper presents a new method of indexing and retrieval of planar objects based on feature points and its application to document image retrieval using cameras. As the indexing method we propose a method based on local combinations of projective invariants calculated from feature points. As the retrieval method we employ a voting technique for efficiency and robustness against erasure of feature points. Experimental results on 10,000 images with 50 queries show that the method is effective (98% accuracy; the remaining query was ranked at the 5th position among 10,000) and efficient (0.14 second per query).

1. Introduction

Document image retrieval is a task of searching document images relevant to a user's query. For meeting diverse needs from users, a wide variety of queries have been employed [1]. With document images as queries, the task of finding similar or equivalent document images has been considered. For scanned documents it is called "document image matching" or "duplicate detection" [2, 3]. This paper concerns a kind of document image matching with camera captured documents as queries. We call this task "camera-based document image retrieval".

In order to deal with camera captured images, various kind of problems including perspective distortion, uneven lighting and focusing should be solved [4, 5]. We are concerned here with the problem of perspective distortion. An ordinary way of dealing with the distortion is to normalize the image by estimating parameters of projective transformation. In this paper we employ a different approach to this problem with the help of *invariants* and *hashing*.

In the field of computer vision, a method called geometric hashing [6] is well-known as an effective way of indexing and retrieval of images. In geometric hashing, images are represented as a collection of points, and images in the database or *models* are indexed with invariants calculated from their points. The voting technique is employed for distinguishing models based on a query image. It is difficult, however, to apply geometric hashing to camera-based document image retrieval since ordinary geometric hashing can deal with similarity or affine transformation; it cannot handle perspective distortion in an efficient way.

To solve this problem, this paper presents a new method of indexing and retrieval for images of planar objects using techniques of hashing and voting for feature points of images. As the feature points we utilize centroids of word regions. The main contribution of this paper is the proposal of a new hash key that is effective and efficient even under perspective distortion as well as erasure of some feature points. A projective invariant called "cross-ratio" is employed for the robustness to perspective distortion. The hash key is defined based on *local combinations* of feature points. The locality allows us to make the method insensitive to point erasure. The discriminability of the hash key is boosted by combining the feature points. From the experimental results on 10,000 document images, it is shown that the method can achieve almost perfect retrieval (only 1 of 50 queries is missed) within a short period of time (0.14 second per query in average).

2. Proposed method

2.1. Fundamental ideas

There are some problems to be solved for achieving camera-based document image retrieval: images captured by cameras can be projectively transformed, images may not include whole text regions, and resolution and illumination of images may be different from those in the database. Basic ideas for solving these problems are as follows:

(1) Hash based indexing and retrieval as voting

In order to make retrieval computationally feasible, we employ hashing and voting for documents. In the proposed method, a document image with the largest number of votes is selected as the result.

(2) Invariant-based hash key

In order to make the hash keys of document images projectively invariant, we calculate them using cross-ratios. As feature points from which cross-ratios are calculated, centroids of word regions are utilized, since they are robust to projective transformation and noises.

(3) Local combinations of feature points

In order to make the hash key insensitive to point erasure as well as to improve its discriminability, we locally combine feature points.

2.2. Cross-Ratio

The cross-ratio is known as an invariant of projective transformation. It is calculated using coordinates of five coplanar points on an image. For five points ABCDE, the cross-ratio is calculated as

$$\frac{P(A,B,C)P(A,D,E)}{P(A,B,D)P(A,C,E)} \quad (1)$$

where P(A,B,C) is the area of a triangle with apexes A, B, and C [7]. Since the cross-ratio is a projective invariant, its value keeps unchanged even if coordinates of points ABCDE change by perspective distortion.

Although the values of cross-ratios obtained from feature points are continuous, they must be converted to k discrete values in order to be used as indices. Values should be discretized by taking into account their frequency: the discretization step should be finer for values occurring more frequently. In the proposed method, discrete values are assigned in proportion to the frequency of values of cross-ratios using a histogram of values of cross-ratios obtained in a preliminary experiment.

2.3. Overview of processing

Figure 1 shows the overview of processing. At the step of feature point extraction, a document image is transformed into a set of feature points. Then feature points are inputted into the registration step or the retrieval step. These steps share the step of calculation of indices. In the registration step, every feature point in the image is registered into the document image database using its index. In the retrieval step, the document image database is accessed with indices to retrieve images by voting. We explain each step in the following.

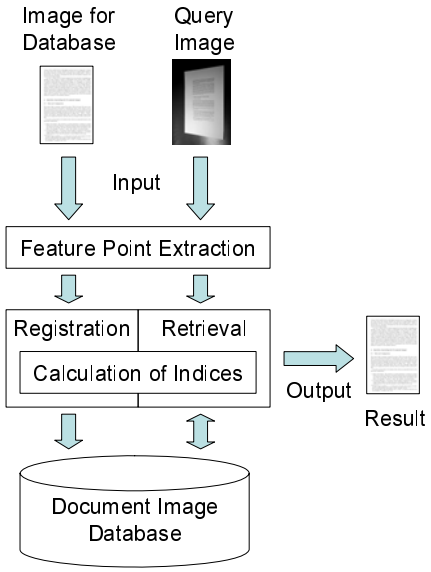
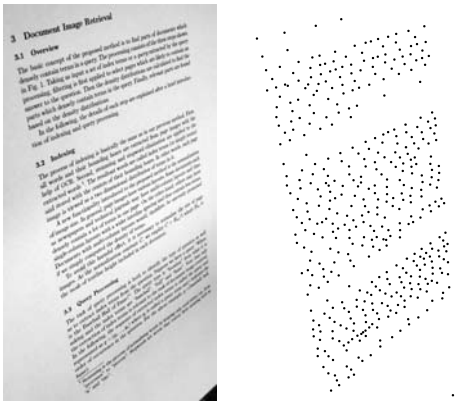


Figure 1. Overview of processing.



(a) input image (b) feature points

Figure 2. Feature point extraction.

2.4. Feature point extraction

Feature points should be obtained identically even under the perspective distortion, noise and low resolution. We employ centroids of word regions as feature points because they almost satisfy this requirement. First, input images (Fig. 2(a)) are adaptively thresholded into binary images. Next, the binary images are blurred using the Gaussian filter whose parameters are determined based on an estimated character size (the square root of a mode value of areas of connected components). Then, the blurred images are adaptively thresholded again. Finally, centroids of word regions (Fig. 2(b)) are extracted as feature points.

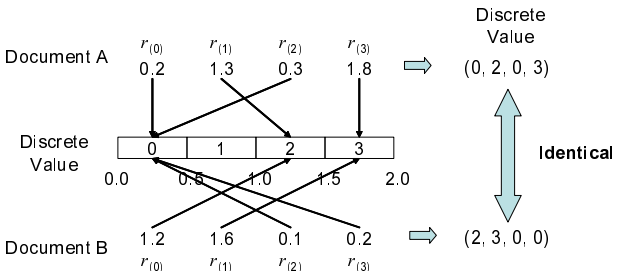


Figure 3. Discriminability of cross-ratios.

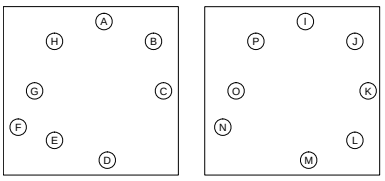


Figure 4. n points.

2.5. Calculation of indices

In the proposed method, each feature point is characterized by cross-ratios. Although it seems reasonable to calculate a cross-ratio for each feature point based on its five nearest feature points, it is not appropriate since in general the nearest points vary due to the projective distortion.

Another important problem is the discriminability of cross-ratios as illustrated in Fig. 3 that represents a case with quantization level $k = 4$. Suppose we have cross-ratios $r_{(0)}$,$\cdots$,$r_{(3)}$ for documents A and B. Although the real values are different, their discrete versions are identical. Moreover, it is impossible to distinguish the documents A and B by counting the votes for each discrete value (0:twice, 2 and 3: once). Although the discriminability could be improved by increasing the level of quantization k, it sacrifices the robustness to noise.

In the proposed method, we attempt to solve the first problem using local combinations of feature points. The index of a feature point is calculated not just from the five nearest points but from the n nearest points. It is often the case that $m(< n)$ points in the n points are kept unchanged under ordinary perspective distortion.

Let us explain in more details with Fig. 4 which represents the $n(= 8)$ nearest feature points for a feature point in a document image and those for the corresponding feature point in a query image. In this figure, $m(= 7)$ points ABCDFGH and IJKMNOP are common. Thus the common combination of feature points can be obtained by examining all possible ${}_nC_m$ combinations. From the same combination of m points, the common cross-ratios are obtained

1: **for each** $p \in$ {All feature points in a database image} **do**
2: $P_n \leftarrow$ The nearest n points of p (clockwise)
3: **for each** $P_m \in$ { All m points combinations from P_n } **do**
4: **for each** $P_5 \in$ { All 5 points combinations from P_m } **do**
5: $r_{(i)} \leftarrow$ The cross-ratio calculated with P_5
6: **end for**
7: $H_{\rm index} \leftarrow$ The hash index calculated by Eq. (2).
8: Register the item (document ID, point ID, $r_{(0)}$,$\cdots$,$r_{({}_mC_5-1)}$) using $H_{\rm index}$
9: **end for**
10: **end for**

Figure 5. Registration algorithm.

by combining all possible ${}_mC_5$ points for calculating cross-ratios from points such as ABCDF and IJKMN, ABCDG and IJKMO.

The second problem of discriminability is solved by taking into account the order of cross-ratios. In the case of Fig. 3, the cross-ratios are different if we consider them as the sequences (0,2,0,3) and (2,3,0,0). Note that if a feature point in a database image corresponds to that in a query image, the sequence should be identical. Consider again the case in Fig. 4. A sequence of cross-ratios are calculated for every m points. Let a series of letters such as ABCDF represent the cross-ratio defined by these points. If the points correspond with each other, the sequence of cross-ratios from m points (ABCDF, ABCDG, ABCDH, BCDFG, BCDFH, ...) and its corresponding sequence (IJKMN, IJKMO, IJKMP, JKMNO, JKMNP, ...) become identical.

The following is the summary of calculation of indices. For each feature point, its n nearest points are obtained. Then all possible ${}_nC_m$ combinations of m points are generated from n points. Indices are defined as ordered cross-ratios by taking ${}_mC_5$ combinations from m points in the fixed order.

2.6. Registration

Let us turn to the registration step. Figure 5 shows the algorithm of registration of document images to the database. In this algorithm, the document ID is the identification number of a document, and the point ID is that of a point.

Next, the index of the hash table $H_{\rm index}$ is calculated by the following hash function:

$$H_{\rm index} = \left(\sum_{i=0}^{{}_mC_5-1} r_{(i)} k^i \right) \bmod H_{\rm size} \qquad (2)$$

where $r_{(i)}$ is the discrete value of the cross-ratio, k is the level of quantization of cross-ratios and $H_{\rm size}$ is the size of the hash table.

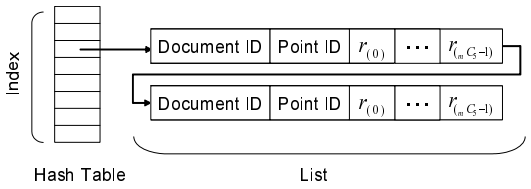


Figure 6. Configuration of the hash table.

1: **for each** $p \in$ {All feature points in a query image} **do**
2: $P_n \leftarrow$ The nearest n points of p (clockwise)
3: **for each** $P_m \in$ { All m points combinations from P_n } **do**
4: **for each** $P'_m \in$ { Cyclic permutations of P_m} **do**
5: **for each** $P_5 \in$ { All 5 points combinations from P'_m} **do**
6: $r_{(i)} \leftarrow$ The cross-ratio calculated with P_5
7: **end for**
8: $H_{\mathrm{index}} \leftarrow$ The hash index calculated by Eq. (2).
9: Look up the hash table using H_{index} and obtain the list.
10: **for each** Item of the list **do**
11: **if** Conditions 1 to 3 are satisfied **then**
12: Vote for the document ID in the voting table.
13: **end if**
14: **end for**
15: **end for**
16: **end for**
17: **end for**
18: Calculate the score based on the votes.
19: Return the document image with the maximum score.

Figure 7. Retrieval algorithm.

The item (document ID, point ID, $r_{(0)}, \cdots, r_{({}_mC_5-1)}$) is registered into the hash table as shown in Fig. 6 where chaining is employed to collision resolution.

2.7. Retrieval

The retrieval algorithm is shown in Fig. 7. In the proposed method, retrieval results are determined by voting on documents represented as cells in the voting table.

First, the hash index is calculated at the lines 5 to 8 in the same way as in the registration step. At the line 9, the list shown in Fig. 6 is obtained by looking up the hash table. For each item of the list, a cell of the corresponding document ID in the voting table is incremented if the following conditions are satisfied.

Condition 1: All values of $r_{(0)} \cdots r_{({}_mC_5-1)}$ in the item are equal to those calculated at the lines 5 to 7 for P'_m.

Condition 2: It is the first time to vote for the document ID with the point p.

Condition 3: It is the first time to vote for the point ID of the document ID.

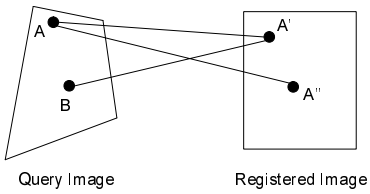


Figure 8. Incorrect correspondence.

The condition 1 aims to remove items with different sequences of cross-ratios. Note that a sequence of cross-ratios $r_{(0)} \cdots r_{({}_mC_5-1)}$ is not necessarily identical for items with the same value of the hash function H_{index}.

The conditions 2 and 3 aim to limit votes caused by inconsistent correspondences. In the algorithm in Fig. 7 voting is to seek points which correspond to the point p. If only the condition 1 is employed, we face the following two types of inconsistency shown in Fig. 8: (Type 1) A point (A) in the query image corresponds to more than one point (A' and A") in a registered image. (Type 2) A point (A') in a registered image corresponds to more than one point (A and B) in the query image. In order to avoid such inconsistent correspondences, the conditions 2 and 3, which are for the types 1 and 2, respectively, are employed.

After repeating these steps for every point, the voting table with votes on every registered document is obtained. In spite of the above conditions 2 and 3, votes caused by incorrect point correspondences are generally obtained. The number of such incorrect votes is approximately in proportion to the number of feature points in a registered image. Hence registered images with a larger number of feature points tend to have unfairly larger votes. In order to compensate for the number of unfair votes, the following score $S(d_i)$ for a document d_i is calculated based on the numbers of votes $V(d_i)$ and feature points $N(d_i)$:

$$S(d_i) = V(d_i) - p_n \cdot N(d_i) \tag{3}$$

where p_n is the proportionality constant of the number of feature points to those of incorrect votes, which is determined by a preliminary experiment. Finally, the document with the maximum score is determined as the result.

3. Experimental results

3.1. Overview

In order to examine effectiveness of the proposed method, we measured accuracy and processing time. Query images were captured from a skew angle using the digital camera CANON EOS Kiss Digital (also known as EOS-300D; 6.3 million pixels) with EF-S 18-55mm USM. The

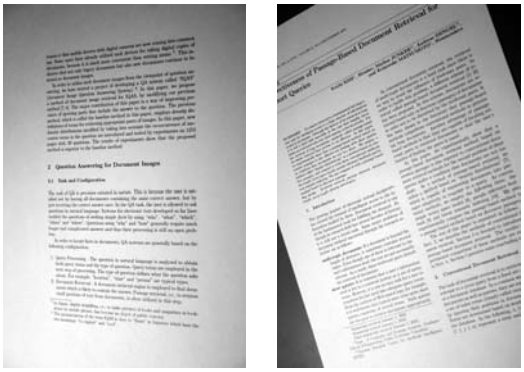


Figure 9. Examples of query images.

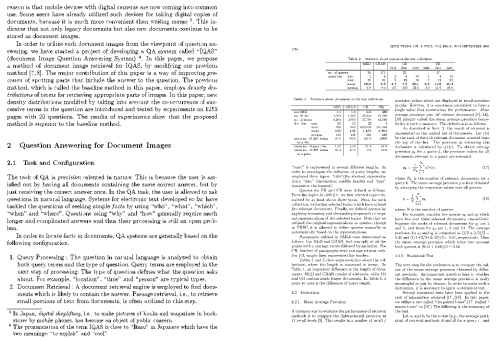


Figure 10. Examples of images in database.

Table 1. Contents of the database.

Title	Registered pages
CVPR 2001	1630
CVPR 1999	1211
ICCV 1999	1170
IDCAR 1997	609
ICPR 2002	2426
ICPR 2004	2724
IWFHR 2004	65
IEEE Transactions on Multimedia 1999	144
Others	21

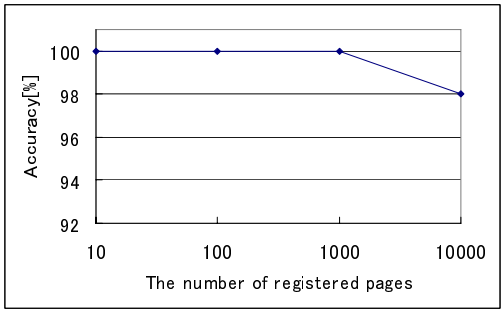


Figure 11. Accuracy of retrieval.

number of query images was 50. Figure 9 shows examples of query images whose size is 2,048 $\times$ 3,072. As documents in the database we employed 10,000 page images converted with 200 dpi from PDF files of single- and double-column English papers collected from CD-ROM proceedings shown in Table 1. Figure 10 shows examples of images in the database whose size is about 1,700 $\times$ 2,200. Note that the pages in the database look quite similar because all pages are from scientific papers. Experiments were performed on a workstation with AMD Opteron 1.8GHz CPUs and 4GB memory. Parameters described in Sect. 2 were set to $n = 8, m = 7, k = 10, H_{\text{size}} = 128M, p_n = 0.022$.

3.2. Accuracy of retrieval

We first analyzed the relationship between the size of the database (the number of registered pages) and the accuracy of retrieval (the rate that the correct page receives the maximum score). The results are shown in Fig. 11: the accuracy of 100% was obtained for the sizes of 10 to 1,000 pages, and 98% for 10,000 pages. Figure 12 to 14 show some examples of a query image and 1st to 5th ranked images.

The query image that caused failure for the case with 10,000 pages is shown in Fig. 14(a): the correct page was ranked in 5th position. We consider that the reason of failure on this image is its narrow text region; it becomes more difficult to obtain correct correspondences between points if text regions are smaller since the number of feature points is small. Figure 15 shows successful and erroneous cases of point correspondences. As illustrated in Fig. 15(b), narrow text regions limit correct correspondences.

Figure 16 shows the relationship between the number of pages in the database and the average ratio of scores for the first to second ranked pages. As the size of the database grows, the difference between scores for the first and second ranked pages decreases. This is because expansion of the database increases the chance of having similar configuration of points.

3.3. Processing time

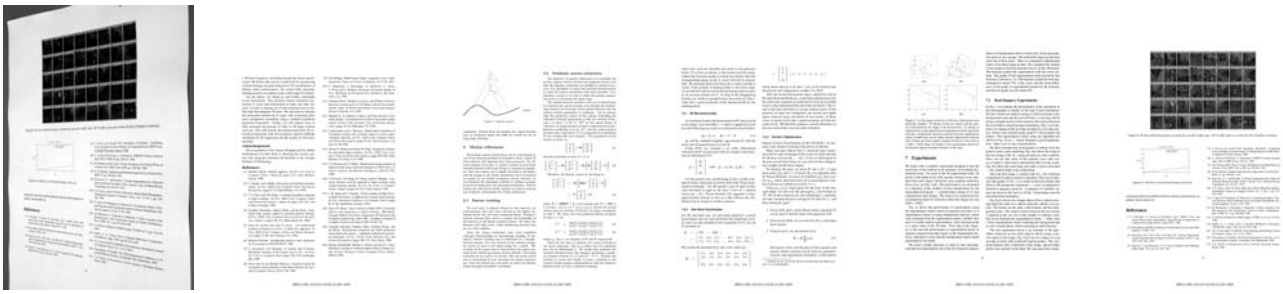
Next, we analyzed how the database size affects processing time. Figure 17 shows the results. The growth of the number of pages accompanies the increase of processing time. This figure also shows the average length of lists in the hash. The average list length is the average number of length of lists with at least one entry. The average list length, which means the number of collisions, increases as the number of registered pages increases. That is the reason

(a) (b) (c) (d) (e) (f)

Figure 12. Successful case. (a) query image, (b) 1st, (c) 2nd, (d) 3rd, (e) 4th, and (f) 5th ranked image.

(a) (b) (c) (d) (e) (f)

Figure 13. Successful case. (a) query image, (b) 1st, (c) 2nd, (d) 3rd, (e) 4th, and (f) 5th ranked image.



(a) (b) (c) (d) (e) (f)

Figure 14. Failure case. (a) query image, (b) 1st, (c) 2nd, (d) 3rd, (e) 4th, and (f) 5th ranked image.

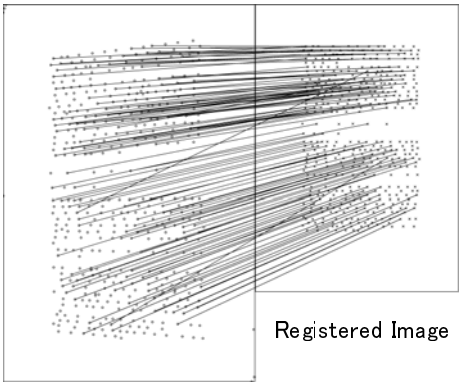
of the increase of processing time.

4. Related work

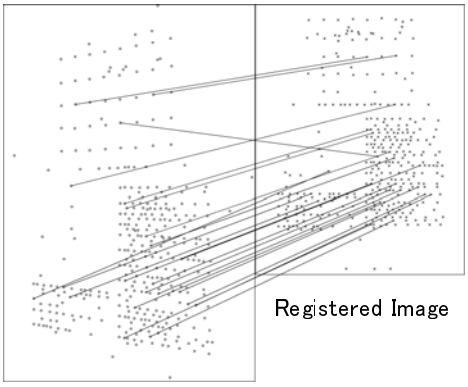
The proposed method can be said as a method for object recognition since it is for retrieval of the corresponding models to query images from a database. There have been many methods for object recognition which utilize invariants as the proposed method does. In this section, we describe similar methods and differences from them.

4.1. Geometric hashing

As mentioned above, the geometric hashing is a method of object recognition based on invariants. In the geometric hashing, all feature points of models are registered into a hash table using 2 to 4 selected points for defining a local coordinate basis. The number of points for the basis depends on the kind of invariance: 2 for similarity, 3 for



(a) successful case



(b) erroneous case

Figure 15. Point correspondences between a query and a database image.

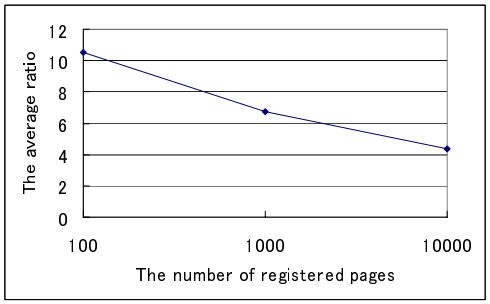


Figure 16. The average ratio of scores for the first to the second ranked pages.

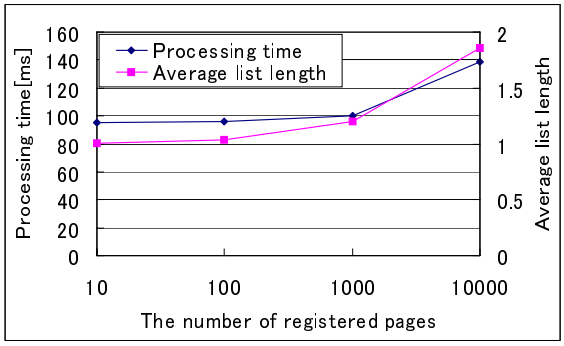


Figure 17. The relationship among the number of registered pages, processing time and the length of lists in the hash table.

affine, and 4 for projective transformation invariance. Registration is performed on every possible basis. Retrieval is performed by looking up from the hash table using an arbitrarily selected basis and voting. The geometric hashing is similar to the proposed method in the following points.

- Invariance for transformation
- Registration of each point
- Utilization of hashing

However, the proposed method is superior to the geometric hashing in terms of computational complexity. In the proposed method, features are calculated from limited neighboring points for each feature point in the registration and the retrieval processes. Hence the computational complexity of the proposed method is O(N) where N is the number of feature points in each model. On the other hand, in the geometric hashing each feature point is registered using every possible basis. Hence the computational complexity of registration is O(N^{b+1}) where b is the number of points for defining a basis. For example, for the case of projective invariants, the computational complexity of geometric hashing is O(N^5) since four points are necessary for the basis.

4.2. Other methods

Many invariant-based object recognition methods such as [8] and [9] have so far been proposed. However, improvement of discriminability by combinating invariants is not employed in these methods. For example, the feature is simply a cross-ratio of five connected line segments in [9]. It is difficult in our case to adopt such a simple indexing, because a huge number of points have similar cross-ratios.

In order to avoid the problem, the sequence of cross-ratios is employed in the proposed method; this high-dimensional feature realizes high accuracy and computational efficiency.

5. Conclusion

We have proposed a method of indexing and retrieval of planar objects based on feature points and its application to camera-based document image retrieval. The method is characterized by the hash key calculated from local combinations of projective invariants. High accuracy and efficiency of the proposed method were shown by the experimental results. Future work includes experiments with more queries and an extension of the method to object retrieval in scene images.

References

[1] D. Doermann: "The Indexing and Retrieval of Document Images: A Survey", Computer Vision and Image Understanding, **70**, 3, pp.287–298 (1998).

[2] J. J. Hull : "Document image matching and retrieval with multiple distortion-invariant descriptors", Document Analysis Systems, pp.379–396 (1995).

[3] D. Doermann, H. Li and O. Kia: "The detection of duplicates in document image databases", Proc. ICDAR'97, pp.314–318 (1997).

[4] D. Doermann, J. Liang and H. Li: "Progress in camera-based document image analysis", Proc. ICDAR'03, pp. 606–616 (2003).

[5] P. Clark and M. Mirmehdi: "Recognising text in real scenes", IJDAR, **4**, pp. 243–257 (2002).

[6] H. J. Wolfson and I. Rigoutsos: "Geometric hashing: an overview", IEEE Computational Science & Engineering, Vol. 4, No. 4, pp.10–21 (1997).

[7] T. Suk and J. Flusser : "Point-based projective invariants", Pattern Recognition 33, pp.251–261 (2000).

[8] B. Huet and E. R. Hancock: "Cartographic indexing into a database of remotely sensed images", WACV96, pp.8–14(1996).

[9] C. A. Rothwell, A. Zisserman, D. A. Fosyth and J. L. Mundy: "Using projective invariants for constant time library indexing in model based vision", Proc. BMVC, pp.62–70(1991).

Experiments in Video-Based Whiteboard Reading

Gernot A. Fink Markus Wienecke[1] Gerhard Sagerer
Bielefeld University, Faculty of Technology
33594 Bielefeld, Germany
gernot@techfak.uni-bielefeld.de

Abstract

With the increasing computational support for collaborative work-environments electronically enhanced whiteboards have been developed to serve as automatic meeting assistants. The most flexible of these systems use cameras to observe the whiteboard, and, therefore, do not require the use of special pens or erasers. However, currently these systems are only capable to interpret some special graphical symbols and can not produce transcripts of the documents written on them. As a major advancement beyond the state-of-the-art we propose a system for automatic video-based reading of unconstrained handwritten text from a whiteboard. Text lines are extracted from the captured image sequence using an incremental processing strategy. The recognition results are then obtained from the text-line images by off-line techniques and a segmentation-free statistical recognizer. We will present results on a writer independent unconstrained handwriting recognition task showing that handwriting recognition can successfully be applied to automatically reading texts from whiteboards.

1. Introduction

Whiteboards are very popular tools not only for presentations and educational purposes but also in meeting rooms for the exchange of ideas during group discussions, for project planning, system design, etc.

In order to make use of whiteboards as user interfaces for human computer interaction in such collaborative working environments, systems based on electronic whiteboards have been developed. Similar to digitizing tablets these systems employ electronic pens and erasers allowing their positions in the plane to be sensed and tracked during the writing process. This data can then be used to construct an electronic version of the document-image on the whiteboard. Additionally, the pen trajectory can be interpreted by an on-line recognition module to automatically recognize what was written on the board.

However, electronic whiteboards exhibit some disadvantages. As special pens and erasers are necessary, the natural interaction is restricted. Therefore, a promising alternative is to retain ordinary whiteboards and pens, and to observe the writing process using a video camera.

In order to cover a large area of the whiteboard, the preferable position of the camera is several meters in front of the board, either mounted to the ceiling or fixed on a tripod. However, in such a setup the writer will usually stand in front of the board while writing. Therefore, the pen and portions of text are frequently occluded by the user. In order to circumvent this drawback, a kind of activity analysis could be employed to decide whether the captured image is suitable for further processing. An alternative method is to extract only the visible portions of the handwritten text and to incrementally integrate the partial transcriptions into the overall recognition result.

In our previous work we proposed a system for automatic video-based whiteboard reading [14]. In contrast to the approaches presented in [9, 10], which only permit the recognition of a limited set of symbols, our system is designed for recognizing unconstrained handwritten text. As the pen is rarely visible in the image and thus on-line recognition based on the pen trajectory is not feasible, an incremental off-line recognition approach is applied. In this paper we will present results of a thorough evaluation of our system on a writer independent unconstrained handwriting recognition task that clearly demonstrate that handwriting recognition can successfully be applied to automatically reading texts from whiteboards.

In the following section we will briefly review some relevant related work. In section 3 we will give an overview of the architecture of the proposed whiteboard reading system. The techniques applied for statistical modeling and recognition of unconstrained handwritten texts will be described in

1 Markus Wienecke now is with Siemens AG, Logistics and Assembly Systems, Postal Automation Division, Constance, Germany.

2 An extended version of this paper will appear in *Int. Journal on Document Analysis and Recognition* (IJDAR).

detail in section 4. Finally, the results of our extensive evaluation experiments will be presented in section 5.

2. Related Work

Electronically enhanced whiteboards that do not make use of specialized hardware for pen tracking, observe the board with cameras. In these systems pen movements or relevant image regions have to be extracted from the captured image sequences.

One approach is to use a special marker for writing that has a distinctive color. By tracking that pen a temporal trajectory is obtained that can be recognized using on-line methods. In [1] such a system is described, which allows the user to control a computer with simple gestures produced by a special marker pen. A video-based system which is capable to track an ordinary pen in image sequences was proposed by Munich & Perona [8]. In [4] the trajectories generated by this approach were used for on-line handwriting recognition.

On-line type systems as these can be successfully employed in scenarios where the pen is always visible in the image. However, they can hardly be applied for whiteboard reading where the pen is very often occluded by the writer.

Therefore, a contrary approach for video-based whiteboard reading is to extract and analyze the relevant image regions after the writing process has finished. For example, the video-based *BrightBoard* system described in [10] continuously observes the whiteboard and grabs a suitable image when the motion of the writer stops. This image is analyzed in order to find and recognize graphical marks that correspond to control commands. A similar camera-based whiteboard scanner is the so-called *ZombieBoard* system proposed in [9], which applies a mosaicing algorithm to enable high-resolution imaging. The system monitors activity in front of the board and detects the drawing of graphical marks indicating commands and associated parameters.

As the system presented in this paper is not restricted to a small set of commands but is designed for recognizing unconstrained handwritten text the approach is also closely related to the task of off-line handwriting recognition (cf. e.g. [11]). In contrast to isolated word recognition the task of recognizing unconstrained handwritten texts using a large or even unlimited vocabulary is much more difficult. This is mainly caused by the absence of context knowledge and word segmentation information. Most current systems, therefore, rely on segmentation-free methods in order to avoid errors introduced by segmenting the text into words or even characters at an early stage. Especially, Hidden-Markov Models (HMMs) were successfully applied and gained growing interest in the research community. Advanced systems for writer-independent unconstrained text recognition are described, e.g., in [5, 6, 12] or [13].

3. System Overview

Our system for automatic whiteboard reading applies an incremental processing strategy. The writing process is continuously observed and the recognition process is activated as soon a handwritten text region is visible in the image. Thus, the text regions are transcribed in their order of appearance and integrated into the overall recognition result.

The writing process is observed with a video camera positioned approximately 3 m from the whiteboard. After grabbing an image that shows an area of approximately 70×50 cm all text regions currently visible are extracted. In order to avoid recognizing the same text region multiple times in the image sequence, we employ a region memory containing all the different text regions extracted so far.

If a new, not yet memorized text region is found, several pre-processing steps are applied to compensate for the highly varying background intensity and to normalize the handwriting. First, the region image is binarized using an adaptive threshold that depends on the intensity distribution in a local neighborhood. Then the the vertical position, skew, and slant of each text region are corrected locally. Especially for the baseline estimation a local procedure is absolutely crucial as in texts written on a whiteboard frequently baseline drifts can be observed (see figure 1). In order to make the subsequent feature extraction process more robust, finally, the size of the handwriting is normalized. This is achieved by rescaling the line images such that the average distance between local extrema of the text contour matches a predefined distance.

From the pre-processed line images a set of 9 geometric features is extracted in a sliding window technique similar to the approach described in [6]. For considering a wider context, we additionally compute an approximate horizontal derivative for each component of the feature vector, so that an 18 dimensional feature vector is obtained. The details of the text extraction, preprocessing, and feature extraction methods applied can be found in [14].

4. Statistical Modeling & Recognition

A successful statistical recognition system for handwriting or spoken language consists of two modeling components, one that describes the realization of individual segments, e.g. words or characters, and another that describes the restrictions on the expected segment sequences. The first component is usually realized by HMMs that model the probability density $p(\boldsymbol{x}|\boldsymbol{w})$ of observing a certain sequence of feature vectors $\boldsymbol{x}$ given a sequence of words or characters $\boldsymbol{w}$. The restriction of these sequences to plausible ones is achieved by defining a probability distribution $P(\boldsymbol{w})$ for all possible sequences, which can be realized by a Markov-chain or n-gram model. The goal of the recogni-

	Source	Type	Categories	Documents	Writers	Lines	Words	Characters
Training	IAM-DB	scanned document	A – D	492	>200	4222	36582	189852
		text prompt	A – D	492	–	–	37273	–
Cross-validation	IAM-DB	scanned document	E – F	129	≈50	1081	9612	49002
Test	whiteboard	video document	F01	20	10	173	1171	6171

Table 1. Corpora of handwritten & text data: word counts include punctuation and word fragments resulting from hyphenation; character counts include approximately 20% of white space.

tion process is then to find the word or character sequence $\hat{\boldsymbol{w}}$ that maximizes the probability of the combined statistical model given the observed data $\boldsymbol{x}$ according to:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, p(\boldsymbol{x}|\boldsymbol{w})P(\boldsymbol{w})$$

In analogy to the terminology used in spoken language processing the HMM $p(\boldsymbol{x}|\boldsymbol{w})$ could be termed the *writing model* and the n-gram model $P(\boldsymbol{w})$ is equivalent to the so-called *language model*.

4.1. Corpora

For the design of statistical recognition systems the availability of a sufficiently large database of training samples is an important prerequisite. Ideally, for a video-based system it would be desirable to obtain a large amount of image data recorded while observing a subject writing on the whiteboard. However, recording and labeling of such video data requires a substantial manual effort. Therefore, we decided to use the IAM-database of scanned documents [7] for training and cross-validation. The database provides a large amount of handwritten text documents that were produced by several hundred subjects. The documents are divided into categories according to the different topics covered.

Unfortunately, the IAM-database does not contain writer IDs for the handwritten samples. However, writers never provided samples for different categories. Therefore, we defined the training data to comprise categories A to D and the cross-validation data categories E & F. This partitioning corresponds to the training and test sets used in [13] and ensures all experiments to be truly writer independent.

The test data was collected in our lab by recording image sequences of texts written on a whiteboard. In order to be able to compare the performance of the video-based system with our off-line recognizer [13], we asked ten subjects to write portions from the off-line cross-validation texts on the whiteboard, namely from category F01. No constraints with respect to the writing style were given. In contrast to the training patterns resulting from scanned forms, where rulers on a second sheet put below were used to align the baseline horizontally, the video-based data often shows baseline drifts and variations of the corpus height.

A summary of the relevant characteristics of the corpora used is given in table 1. Figure 1 shows examples of a scanned document used for training and the final version of a video document from the test data. Additionally, the results of the incremental text detection are shown, which, for the example given, produces the lines in a different order than found in the final document.

4.2. Writing Model

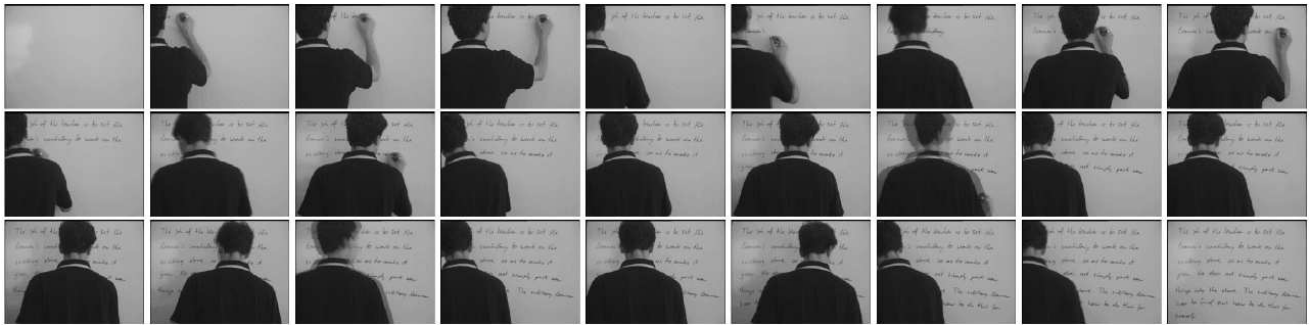
The configuration and parameter estimation for the HMMs defining the writing model as well as for the language models used is carried out in the framework of the ESMERALDA development environment [3].

As general setup we use semi-continuous HMMs with a shared codebook of approximately 2000 Gaussian mixtures with diagonal covariance matrices. A total of 75 HMMs are created for modeling 52 letters, ten numbers, twelve punctuation marks and brackets, and white space. The latter consists of three variants accounting for different lengths in blank space between words or characters. All these models use the *Bakis*-type topology, i.e. they are basically linear models which in addition to loops and forward state-transitions permit the skipping of states in the sequence. Thus, the models can cope with a wider range of lengths in the character patterns described.

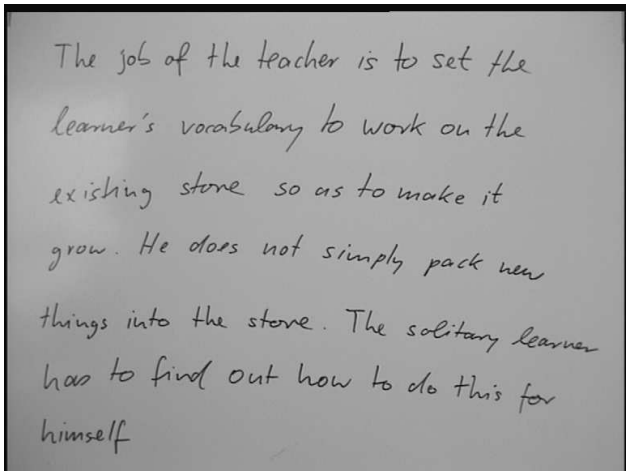
The shared codebook is initialized in un-supervised mode by applying the k-means algorithm to the training data. Then the initial HMM parameters can be determined on labeled initialization data. Afterwards, we apply several iterations of the Baum-Welch parameter re-estimation to the models. From the context-independent character model set thus obtained, models for arbitrary words of some given lexicon can be constructed easily by concatenating the appropriate character models.

4.3. Language Model

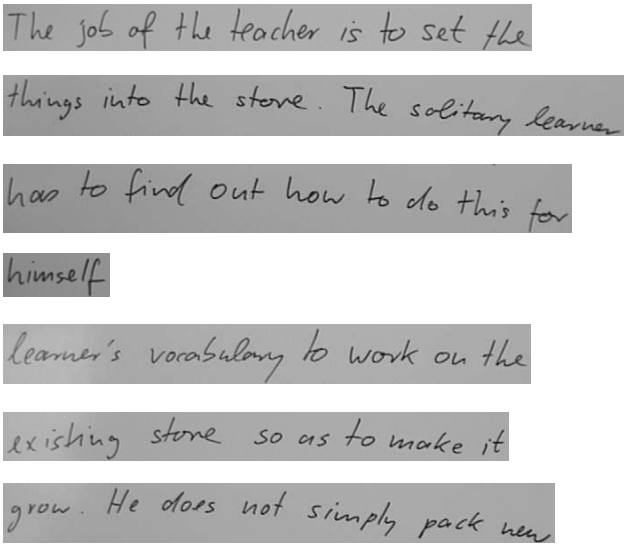
For estimating character-based language models the transcriptions of the training data and for word-based mod-



(a)



(b)



(c)

Figure 1. Examples of whiteboard data collected: (a) image sequence captured (summarized), (b) resulting final whiteboard document, and (c) text lines extracted during observation of the writing process (in order of extraction).

els the original text prompts were used. The raw n-gram probability distributions were smoothed by applying absolute discounting and backing-off (cf. e.g. [2]).

A major limitation for the performance of a word-based language model in our configuration of training and test data arises from the fact that the texts belong to different categories covering widely differing topics. From the total of 2534 word forms appearing in the text prompts of the cross-validation data (categories E & F) more than 48% never appeared in the training texts (categories A – D). Additionally, writers sometimes used varying hyphenation which introduces unseen word fragments. In the whiteboard data 316 different word forms are used, more than 26% of which are not covered by the training set. Therefore, we decided to include in addition to the lexicon of the training data all those word forms in the overall recognition lexicon that are necessary to describe the text prompts from which cross-validation and test set were generated. From this word list a small number of entries was eliminated, which contained characters not present in the training material. The resulting recognition lexicon consists of 7485 entries including punctuation and word fragments resulting from hyphenation. The percentage of out-of-vocabulary words for both cross-validation and test data is approximately 0.5%.

5. Results & Discussion

In order to evaluate the proposed methods for video-based whiteboard reading we carried out several experiments on the test set described in section 4.1. Whenever possible the results obtained are compared to those achieved by an off-line recognition system on the cross-validation data. A comparison of those figures with results on data from the IAM-database reported in the literature [5, 6, 12]

also clearly shows the excellent quality achieved in modeling and decoding. Though restricting the possible occurrence of upper-case characters to word-initial positions Kavallieratou and colleagues only achieved character error rates slightly below 30% on the IAM-database [5]. With a 7k vocabulary and a bi-gram language model Marti *et. al* achieve a word error rate of approximately 40% [6]. Vinciarelli *et. al* report error rates between 57% and 55% for different language models ranging from uni- to tri-gram when using a 10k lexicon [12].

5.1. Text Detection

The precondition for whiteboard reading is to robustly detect the image regions of the handwriting. Therefore, we first investigated the effectiveness of the method for text detection. Using the 20 image sequences for testing consisting of 152 handwritten lines of text, it turned out that a total of 188 image regions have been detected. 173 of these regions are correctly detected text regions. In only 15 cases errors occurred due to noise or line segmentation errors caused by touching or heavily overlapping lines. The discrepancy of the total number of originally written lines (152) and the overall number of correctly detected text regions (173) is caused by the incremental processing strategy. Thus, we observed that in 21 cases portions of text lines have been detected repeatedly. Additionally, we investigated whether the sequence of detected regions corresponds to the chronological order in which the text lines were written on the board. From the overall number of 173 text regions the chronological order was not correct in 9 cases (see e.g. figure 1).

5.2. Lexicon-based Recognition

For lexicon-based recognition of whiteboard texts we used a lexicon containing 7485 word forms (see section 4.3). The results achieved are summarized in table 2. Without the use of any restrictions on the possible word sequences we obtain a word error rate of 47.8%. Clearly, such a figure would not be acceptable for an automatic transcription system. However, with some limited knowledge about the expected texts represented as a bi-gram language model this figure could be improved to 28.9%. This corresponds to a reduction of the error rate of approximately 40%. Due to the widely differing lexicons of training and test data the bi-gram model has a very high perplexity on both test and cross-validation set. For a well trained language model that could be estimated on text data *matching* the topics of the final application a substantially lower perplexity can be expected[1]. Therefore, word-based recognition results on white-board data could easily be improved further for better matching training and test conditions.

	% WER / perplexity	
	none	2-gram
Cross-validation	43.9 / (7485)	28.3 / 757
Test (whiteboard)	47.8 / (7485)	28.9 / 645

Table 2. Word error rates (WER) achieved for a 7485-word lexicon with and without using a bi-gram language model.

5.3. Lexicon-free Recognition

Ultimately, any handwriting recognition system should be able to recognize text independently from a predefined list of possible words. For such lexicon-free recognition at least some expectation on the possible sequence of characters is required.

Therefore, we estimated character-based language models with n-gram lengths ranging from two to five (see section 4.3). These models were then used in conjunction with the context-independent character HMMs during the recognition process. The results obtained are shown in table 3. Without the restriction of a language model a character error rate of 31.0% is obtained, i.e. roughly every third character – including white space – is misrecognized. However, when using the statistical restrictions on possible character sequences as represented by the character based language models this figure can be improved significantly. With a 5-gram model a character error rate as low as 19.0% can be achieved on the whiteboard data.

Though the mismatch of lexicons between training and test data is a severe limitation for word-based recognition it has an advantage for the judgment of the lexicon-free results. In principle long-span n-gram models could learn the training lexicon and, therefore, results obtained with such a model might not be truly lexicon-free. In our configuration, however, learning of the word forms found in the training texts has very limited effect on the cross-validation and test data (see also section 4.3). Therefore, the low character error rates achieved impressively demonstrate the capability of the n-gram models to capture more general characteristics of the character sequences.

5.4. Video vs. Off-line Recognition

The comparison of the recognition results obtained on the whiteboard data and on the scanned documents used for cross-validation clearly shows better performance on the

1 Despite a lower perplexity (≈ 400) when using a 10k lexicon and a bi-gram language model in [12] only a surprisingly high error rate of almost 60% is achieved on data from the IAM-database.

	% CER / perplexity				
	none	2	3	4	5
Cross-validation	29.2 / (75)	22.1 / 12.7	18.3 / 9.3	16.1 7.7	15.6 / 7.3
Test (whiteboard)	31.0 / (75)	25.9 / 12.0	22.0 / 8.5	20.1 / 6.9	19.0 / 6.5

Table 3. Character error rates (CER) achieved with different n-gram language models.

latter ones. However, the difference in recognition quality is relatively small when considering the widely different nature of the documents used. This evidence makes it obvious that the methods used for text-detection, preprocessing and feature extraction are capable of compensating for the majority of distortion effects found in the video data.

6. Conclusion

We presented a system for automatic whiteboard reading based on visual input. It is characterized by an incremental processing strategy, i.e. the text lines are extracted as soon as they are visible in the image. The pre-processing and feature extraction methods applied generate a data representation which is to a certain extent robust against variations concerning the writing style and the reduced quality of the video-based data. Evaluation results on a writer independent task were presented for both lexicon-based and lexicon-free recognition of unconstrained handwriting. When using a 7.5k lexicon and a bi-gram model a word error rate of only 28.9% could be achieved. Without an explicit lexicon and the use of only a character 5-gram model a character error rate as low as 19.0% was reached. These results clearly demonstrate the effectiveness of the proposed methods for text detection, preprocessing, feature extraction, and statistical modeling and recognition and their successful combination in a complete system for automatic video-based whiteboard reading.

Acknowledgments

This work was supported by the German Research Foundation (DFG) within project **Fi799/1**.

Additionally, we would like to thank the Institute of Informatics and Applied Mathematics, University of Bern, namely Horst Bunke and Urs-Viktor Marti, who allowed us to use the IAM database of handwritten forms [7] for our recognition experiments.

References

[1] M. J. Black and A. D. Jepson. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In H. Burkhardt and B. Neumann, editors, *European Conf. on Computer Vision*, pages 909–924, Freiburg, Germany, 1998.

[2] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13:359–394, 1999.

[3] G. A. Fink. Developing HMM-based recognizers with ESMERALDA. In V. Matoušek, P. Mautner, J. Ocelíková, and P. Sojka, editors, *Text, Speech and Dialogue*, volume 1692 of *Lecture Notes in Artificial Intelligence*, pages 229–234. Springer, Berlin Heidelberg, 1999.

[4] G. A. Fink, M. Wienecke, and G. Sagerer. Video-based online handwriting recognition. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 226–230, 2001.

[5] A. Kavallieratou, N. Fakotakis, and G. Kokkinakis. An unconstrained handrwiting recognition system. *Int. Journal on Document Analysis and Recognition*, 4:226–242, 2005.

[6] U.-V. Marti and H. Bunke. Handwritten sentence recognition. In *Proc. Int. Conf. on Pattern Recognition*, volume 3, pages 467–470, Barcelona, 2000.

[7] U.-V. Marti and H. Bunke. The IAM-database: An english sentence database for offline handwriting recognition. *Int. Journal on Document Analysis and Recognition*, 5(1):39–46, 2002.

[8] M. E. Munich and P. Perona. Visual input for pen-based computers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):313–328, 2002.

[9] E. Saund. Bringing the marks on a whiteboard to electronic life. In *Proc. 2nd Int. Workshop on Cooperative Buildings, CoBuild'99*, pages 69–78, Pittsburgh, 1999. Springer.

[10] Q. Stafford-Fraser and P. Robinson. Brightboard: A video-augmented environment. In *Proc. Conf. on Human Factors and Computing Systems*, pages 134–141, Vancouver, BC, Canada, 1996.

[11] T. Steinherz, E. Rivlin, and N. Intrator. Offline cursive script word recognition – A survey. *Int. Journal on Document Analysis and Recognition*, 2(2):90–110, 1999.

[12] A. Vinciarelli, S. Bengio, and H. Bunke. Offline recognition of unconstrainded handwritten texts using HMMs and statistical language models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(6):709–720, 2004.

[13] M. Wienecke, G. A. Fink, and G. Sagerer. Experiments in unconstrained offline handwritten text recognition. In *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, Niagara on the Lake, Canada, August 2002.

[14] M. Wienecke, G. A. Fink, and G. Sagerer. Towards automatic video-based whiteboard reading. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 87–91, Edinburgh, 2003.

Web-Based Deployment of Text Locating Algorithms

Simon M. Lucas and Carlos R. Jaimez González
Computer Science Department
University of Essex, UK
{sml,crjaim}@essex.ac.uk,

Abstract

This paper describes a simple yet novel approach to the web-based deployment and evaluation of text locating algorithms.

Web-based deployment allows algorithms to be evaluated by end users or researchers, without the need to install the algorithm. This is a major advantage both for the end user, and for the algorithm developer. The end user is protected from lengthy installation procedures, which may also leave one's machine in a corrupted state. The algorithm developer is protected from theft of software or intellectual property.

Our system provides access to a deployed algorithm in two ways: interactive mode via a web browser, and program access mode via a special kind of web service architecture. The system is demonstrated with the deployment and testing of one of the entries for the ICDAR 2005 text locating competition.

1 Introduction

The Web has already dramatically improved the efficiency of the research process, offering searchable access to a vast number of papers, on-line articles, and discussion forums. For those engaged in empirical computer science research, however, the best may be yet to come. Currently, the *modus operandi* in fields such as computer vision is for researchers to evaluate their own algorithms on public datasets, and then publish the results in a paper, which is subject to a delay of at best several months, but at worst two years or more, before publication. Competitions have been associated with several research communities and conferences, and these help in establishing the state of the art in a particular field, but do to the effort of running them, and of participating in them, are usually run only annually or biennially. We argue that the software technology is now ready to enable researchers to deploy their algorithms as special kinds of web services as a matter of standard practice. Evaluation and comparison on a potentially vast number of datasets can then be an automatic, and on-going process.

Web-based deployment allows algorithms to be evaluated by end users or researchers, without the need to install the algorithm. This is a major advantage both for the end user, and for the algorithm developer. The end user is protected from lengthy installation procedures, which may also leave one's machine in a corrupted state. The algorithm developer is protected from theft of software or intellectual property.

In recent years there has been a great deal of interest in web services, and *Service Oriented Programming* [1] has been proposed as a new programming paradigm. However, so far, the reality has not lived up to the hype, and in most web services perform only simple functions, such as retrieving the current price of a specified stock, for example.

On the other hand, pattern recognition researchers have long recognized the benefits of offering web-based demonstrations of their software. A very relevant example of this is the Carnegie Mellon University Robotics Institute (CMU-RI) Face Detection demonstration[1]. The demonstration system works as follows. Users upload images to the system using a simple two-field form, specifying their email address in one field, and the URI of the image in the other field. This latter detail means that users must be able to upload an image to a web server before interacting with the application. The demonstrator then downloads the image from the URI, informs the user of this, and specifies that an email containing the results will be sent the next day.

We tested the system, and the email did indeed arrive the next day, with a hyperlink to the result image, as shown in Figure 1. Note the hyperlink to the numerical results. An example of these numerical results is shown in Table 1. While this data is useful, and could be used by an evaluation algorithm, it would be even better if it published in XML, instead of *ad hoc* plain text. The problem with plain text output is that it requires manual effort to write parsers for it, and is extremely version sensitive i.e. if a decision is made to change the format of the output or to add a new attribute (e.g. such as the time taken to process the image), then any programs used to read such data must be modified accordingly, which can be tedious, and failure to make such

[1] http://vasc.ri.cmu.edu/cgi-bin/demos/findface.cgi

modifications could lead to either to a program that fails to run, or worse still, or one that appears to run but produces incorrect results.

There are many positive aspects of this CMU-RI Demo. It has been up and running for many months, and has successfully processed hundreds of images. It is simple to operate, though rather slow, (this may have been done deliberately to reduce server load). The slow turnaround allows the possibility of human intervention in the generated results, a possibility that can in practical terms be avoided with an immediate turn-around.

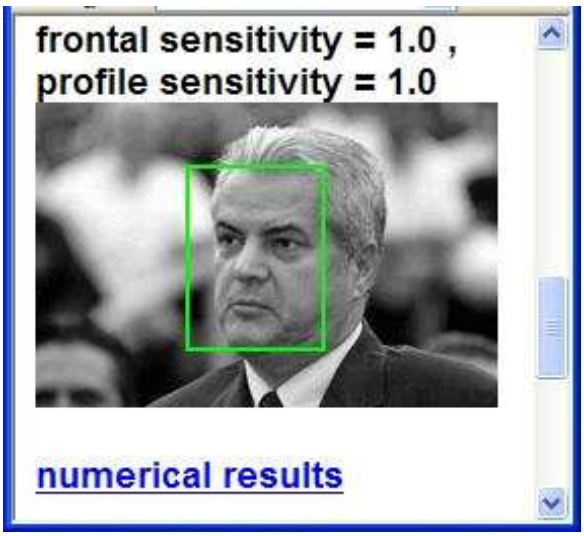


Figure 1. A sample gallery image from the CMU-RI web-based face detector demonstration.

```
1 -- number of faces

 1. Frontal Face
   107    76      -- Position
     6          -- size
    0.44       -- confidence
```

Table 1. Sample Numerical Output from the CMU-RI Face Detector Demonstration

Our system provides access to a deployed algorithm in two ways: interactive mode via a web browser, and program access mode via a special kind of web service architecture. Both deployment modes aim to offer immediate results, subject to server load. Web-browser mode is useful for users wishing to casually test systems on selected images. Program mode, on the other hand, can be used to test deployed algorithms on hundreds of test-cases, or even to build complete systems, where each component is a special kind of web service. The work reported here builds on previous efforts to deploy and evaluate pattern recognition algorithms using the Internet [7] or the Web (meaning HTTP over the Internet) [6]. Compared to previous efforts, the current work differs in two main ways. Firstly, the system is object-based rather than service based. This overcomes limitations in offering parameterized systems as services, since in our system, objects can be directly constructed using the same complex sets of parameters as for local objects. Secondly, the use of URIs to refer to all objects means that the results of all method invocations can be serialized to an XML file and stored on a web-site for future use. In addition to offering web-based access to client programs, we also think that it is important where possible to offer interactive access to deployed programs via a standard web browser.

The first author is currently in the process of evaluating the software submitted to the ICDAR 2005 text locating competition. Of the five entries received, only one of them worked first time. Others failed to operate due to missing library files, or compatibility problems with different versions of Java. While all the problems were eventually overcome, this costs effort both for the competition entrant and for the organizer. The web-based evaluation mode puts the responsibility for making the program run firmly with the entrant. Due to the effort involved in running competitions, for a given field these only tend to happen once every year, or perhaps once every two years (such as the highly successful Fingerprint Verification Contests [9]). The web-based deployment mode makes it practical for algorithms to be continuously deployed and evaluated. Web-based deployment has the potential to turbo-charge research in this area, by allowing good ideas that are well implemented to gain immediate acceptance by the research community.

The main contribution of this paper is to describe a relatively simple yet powerful system that we have developed for distributed object programming over the web. The system is called WOX, which stands for Web Objects in XML. The rest of the paper describes why existing distributed programming protocols are not yet ideal for our purposes, the ideas behind WOX, and how we have already used it to deploy a text locating algorithm as a web service. Note that the problem of finding all the text regions in an image is also referred to as *text detection*, but we prefer the term text locating, as detection could mean simply indicating whether or not an image contained some text.

2 Distributed Programming and Web Services

There are two goals to the work reported here: to make text-locating algorithms available to end users via web-browsers, and also to make them available to end-user client programs.

The latter mode requires some form of distributed programming, whereby a program on the client machine makes a remote method call or a remote procedure call to a remote machine, where for the current exercise, the remote machine can be situated anywhere on the Internet. Actually, we make a further requirement. Due to security measures, many institutions only allow calls to port 80 or 8080 through their firewall, typically to machines running web servers (such as Apache HTTPD), or web application servers (such as Tomcat).

While there are many remote procedure call mechanisms, such as PVM, MPI, RMI, etc. none of these are ideally suited to the task at hand.

The requirement that all traffic be routed through a web server or web application server is quite restrictive, and restricts us to using HTTP. While it is possible to tunnel CORBA or Java RMI calls through HTTP, this exercise may be technically quite demanding, and also rather opaque: we have developed a simpler, and more transparent system.

To gain a degree of language independence and HTTP compliance, we have settled on using XML to encode the procedure call and return messages. This also has the advantage of being human readable, which can aid debugging.

XML-RPC offers a simple mechanism for making remote procedure calls through the web, but only allows for a fixed set of data types to be used. This means that any application specific data types must be translated into the pre-defined types before transmission, which incurs an unnecessary overhead in deploying a new application in this way.

Simple Object Access Protocol (SOAP) might seem like the obvious choice. The acronym is something of a misnomer, however, since it is not simple, and it does not provide access to objects. In other words, there is no SOAP supported way for a client to instantiate an object on a remote server, and then subsequently refer to it just as they would a local object, although of course it is possible to program one's way around this if necessary. Having direct access to objects is important when dealing with stateful programs, but is not an issue when dealing with stateless ones. The text locating programs we have in mind for this application are expected to be stateless, since the result of processing an image is expected to be independent of the previous images seen.

If we were evaluating trainable text locaters, however, then maintaining the state of a text locater (i.e. what it had learnt during training) would be a vital consideration.

Unlike XML-RPC, SOAP does allow for user-defined (application specific) data-types. This is done by installing specific serializers for specific data-types. However, the default encodings used by SOAP for arrays of primitive data-types are extremely inefficient, and hence it is questionable as to whether SOAP would provide any benefits for this kind of application.

By default, SOAP uses an XML element for each element of an array of primitive elements (such as `int`). This means that SOAP-encoded arrays can be over 40 times the size of their binary encoding. The exception to this are byte arrays, that are encoded efficiently using base-64 (which WOX also uses for byte arrays). Given the speed of modern computers, and the fact that many of us have access to high bandwidth Internet connections, this difference in encoding efficiency might seem unimportant. However, table 2 emphasizes how significant this difference is, both in time and space usage. For arrays of more than $30,000$ `int`, the SOAP server crashed with an out of memory error.

method	Time (ms)	Size (kb)
WOX	80	106
SOAP	3,300	4,200

Table 2. Time and space usage for passing an array of $20,000$ `int` using WOX, and SOAP.

2.1 Representational State Transfer (REST)

Many analysts and developers have become extremely frustrated with SOAP's shortcomings, and there is significant interest in an alternative paradigm called REST, which stands for Representational State Transfer [4], [5]. Proponents of REST argue that what made the Web really take off was HTTP, and the notion of a URI - a Uniform Resource Indicator. This gave a standard way to refer to any item of information.

SOAP hides a set of services at a site behind a single URI endpoint used for remote procedure calls, and the details of the service required are encoded in the message instead of the URI.

The idea behind REST is that URIs should be used to name service, data structures and objects directly, in order to exploit the full power of the web. However, REST is an architectural style rather than a concrete protocol.

2.2 Web Objects in XML (WOX)

We have designed a new protocol for making remote method invocations over the web, and we call this WOX, for Web Objects in XML. WOX is based largely on the principles of REST, and each object on a server can be uniquely identified with a URI.

While SOAP does not allow references to objects (i.e. all parameters must be passed by value), WOX allows call by value or call by reference. Call by reference can in some cases make huge savings on network traffic.

The basic ideas behind WOX are as follows. A remote method call to a WOX server specifies the URI of the object (which could be local to that server or remote from it), the name of the method to invoke, and the parameters to that method. Each parameter can be passed by value, or by reference - again for references, URIs are used). The XML

encoding of the object specifies the object's class. This can then be used by the WOX server to load the appropriate class, deserialize the object of that class, and invoke the method on it. When the method invocation has finished, the WOX Server will then serialize the result to XML, which can either be sent directly back to the client, or be saved on the server, and it's URI sent back to the client.

The WOX system currently exists as an operational prototype, which has been implemented only in Java, though we expect that all the concepts used could be applied to any object oriented language.

3 Requirements

In designing our system we began with a set of requirements functional and non-functional requirements for our system to deploy text locating algorithms:

3.1 Functional

The functional requirements are as follows:

Interactive Testing The system must support an interactive mode of usage.

Client Program Access The system must support access to a text locating algorithm to a remote client program.

Graphical Results : the system must provide graphical results of running a text locater on an image.

XML results the system must provide detailed results in XML, which are stored on the web server for future reference and processing.

3.2 Non-Functional Requirements

Simplicity the system should be simple to deploy and use, requiring no expert knowledge of web services.

Efficiency The system should not impose any unnecessary CPU or network bandwidth overheads

Generality The system should be able to deploy algorithm implementations written in any language

Free / Open Source The system should not rely on any commercial tools - it must be free to deploy, and open source to enable others to extend it freely.

Platform Independent The system must be easily runnable on a variety of platforms. For example, a system implemented using Microsoft .Net would be unacceptable to the community.

Although still in prototype, the system already meets its main requirements. Both the web browser interface[2] and the WOX Server[3] are currently running and freely available to interact with as a service, and to download and install as one's own service.

4 The Text Locating Interface

We define the interface to a text locating algorithm in as simple a way as possible, defining data structures where necessary, but avoiding the use of Collection classes as far as possible, as these may be complex to serialize.

We use Java to define the interface, as this is the language that we work with most often, and the one we always use for prototype development. The fact the Java has an `interface` keyword makes the syntax especially appropriate. This does not mean, however, that service implementations are in any way restricted to be in Java, and non-Java programs can either by invoked by starting a new process from within Java, or by using the Java Native Interface.

Table 3 shows the Java interface for a text locating service. Note that this interface assumes that the text locater implementation is pre-trained. An interesting alternative would be to allow for a trainable interface, whereby images tagged with ground-truth rectangles could be uploaded to the service in order to train it. This would lead to two interesting possibilities: either that a common text locater object could be shared among all users, or that an individual text locater be made available to each user (or indeed many per user). The former allows for a community-wide training process reminiscent of the *OpenMind* concept [10], while the latter allows text-locaters to be trained and tested for individual needs.

```
public interface TextLocater {
   public Rect[] locateText(byte[] encodedBytes);
}
```

Table 3. Text Locater Interface.

The only class that the interface depends on is the `Rect` class, shown in Table 4. This simply codes the coordinates of a single rectangle. The interface specifies that a Text Locater should return an array of such rectangles. The actual `Rect` class used by client and server may differ, but providing that they use the same fields, they will be serialized and de-serialized in the same way by WOX, and hence be compatible.

5 The WOX Client-Server Architecture

Figure 2 shows the WOX system architecture, and how it relates to the text locating application. To interact with a

[2] http://algoval.essex.ac.uk:8080/textloc/

[3] http://algoval.essex.ac.uk:8080/WOXServer.jsp

```
public class Rect {
   int x, y;
   int w, h;
}
```

Table 4. The `Rect` class used to store rectangle coordinates.

text locating WOX web service, the client application program first instantiates an object of the appropriate class on the server. To do this, the client passes the class name of the object, together with any parameters, to the WOX server. The WOX Server then attempts to instantiate an object of that class, and then returns either a copy of the instantiated object, or a reference to the instantiated object. In our case, we wish to make all subsequent calls to the object on the server, so we specify that a reference should be returned.

When the WOX client receives the URI of the newly instantiated object, it instantiates a special proxy object, that implements the text locater interface, and will send all client-side method invocations on to the server object for processing. In the case of the rectangle results, in our client program we now wish to receive all the encoded rectangles directly (instead of just receiving URI references to them), so the WOX method call policy is specified accordingly.

Now the client has a reference to the server side text locater implementation object, it can call the `locate` method to find the rectangles in an image. When making this call, we suppose that the image is stored on the client as an array of byte. This is then encoded using the WOX serializer, and send to the WOX server as part of a WOX method call. Note that the client program does all the work through a proxy: it just sees the `textLocater` interface.

The WOX server then deserializes the XML method call to a set of Java objects, attempts to find the object to invoke the method on, makes the invocation, and in this case returns an array of `Rect`, serialized in XML. To get the required logging behaviour that would allow researchers with the aid of a web browser to interrogate the results of any text locating method invocations, we had to install an adapter for this service. The reason for using an adapter is that we wanted all the rectangles found by each text locater to be overlaid on a copy of each original image uploaded to the WOX server. This is not part of the normal WOX method invocation logging process: hence the need for an adapter, as shown in Figure 2.

5.1 XML Result File

Table 5 shows a sample XML result file (Figure 3 shows a different image with the rectangles overlaid on it). There are a number of issues that arise when choosing a format for the XML results file. Currently, our main emphasis is on rapid prototyping, so we are using the default WOX object serialization format to produce all our XML. This eliminates the necessity to produce helper classes to read and write XML from objects, and avoids the need to design any XML representations. The other issue that arises when producing the result images with the rectangles overlaid on them, is when to do this. Here, there are two choices. One option is to store only the original images, then dynamically overlay the rectangles on them for each request. This costs more time, and is potentially brittle, since the image results now requires special software in order to be properly interpreted. Here we chose the less space efficient but simpler option of saving the overlaid images immediately when the text locater has found the rectangles.

```
<object type="problems.roi.TextLocResult" id="0">
  <field name="elapsedTime" type="int"
         value="1063"/>
  <field name="rectangles">
    <array type="problems.roi.Rect" length="6"
           id="1">
      <object type="problems.roi.Rect" id="2">
        <field name="x" type="int" value="102"/>
        <field name="y" type="int" value="142"/>
        <field name="w" type="int" value="182"/>
        <field name="h" type="int" value="158"/>
      </object>
      <object type="problems.roi.Rect" id="3">
        <field name="x" type="int" value="101"/>
        <field name="y" type="int" value="98"/>
        <field name="w" type="int" value="226"/>
        <field name="h" type="int" value="114"/>
      </object>
      <!-- rest of array omitted -->
      </array>
    </field>
</object>
```

Table 5. An example XML results file created by the WOX Text Locating Service Adapter.

5.2 Deployment

We believe that a strong feature of WOX is the simplicity with which algorithms can be deployed. In order to deploy a Java algorithm as a text locater, for example (assuming that the WOX server has already been installed on a web application server such as Tomcat), one has only to copy the Java classes or `jar` file to the appropriate directory on the server (web-based upload is also possible, but we've not yet enabled this, as it as potential security risks). If the algorithm is not implementation in Java, then a Java wrapper can be used to interact with the implementation, either using the Java Native Interface, or by reading/writing files.

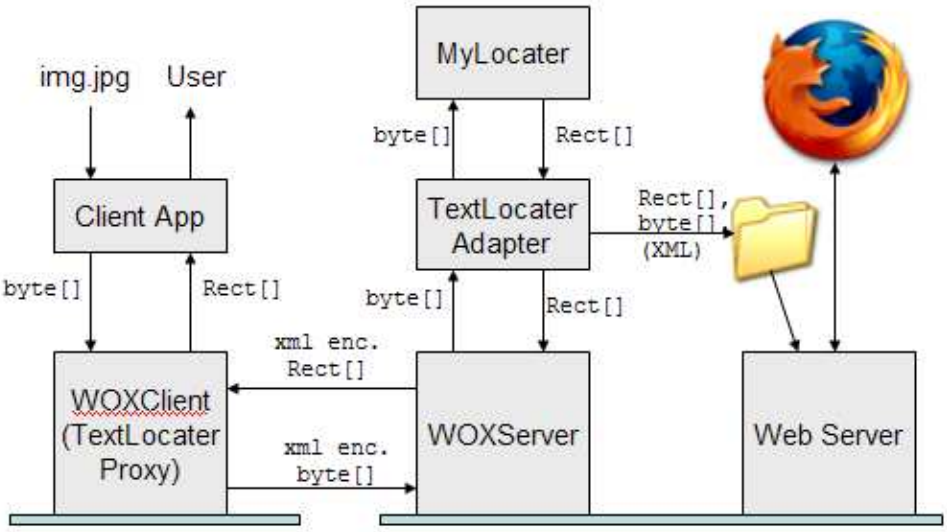


Figure 2. The WOX Client-Server Architecture.

6 Test Results

We tested the system first of all with some trivial text locaters, that would simply decompress an uploaded JPEG image. When the system passed this test, we then installed a non-trivial text locating algorithm with reasonably high performance. With the permission of Alex Chen, we installed his submission to the ICDAR 2005 text locating contest [3] [2]. Of all the submissions, this one is relatively fast, and had proven to be trouble-free in operation. As we only had access to the executable file, we used a file-based mechanism to interact with it. The `TextLocater` object that is exposed on the WOX server is a simple wrapper that takes in the submitted image, saves it to a file, invokes the Chen algorithm, then reads the results of the text detection (which are in XML), and returns the result to the adapter. Recall that the adapter then overlays the detected rectangles on the image, saves that image to a file (together with the XML), and returns the XML-encoded rectangles to the client program.

6.1 Evaluating Text Locaters

The system described so far caters for the web-based deployment and usage of algorithms, in this case, text locating algorithms. So far, we have made no mention of how the accuracy of these algorithms should be evaluated. Our proposal is to keep the usage and evaluation of these algorithms entirely separate. The way this works in our prototype is that the WOX text locating client uses the service to locate the text rectangles in a set of images. In order to assess the accuracy of the service, it is necessary to have access to the ground-truth data. If the client does have access to this, then it may run the necessary evaluation algorithms to score the algorithm. The separation of the running of the algorithm from its evaluation is a good idea, since it allows any number of evaluation algorithms to be run on the detected rectangles. This is especially important for text locating, where the community has yet to agree on the most appropriate measures. The write-up of the ICDAR 2003 text locating competition discussed two alternative metrics, for example [8].

7 Conclusions

We have described a relatively simple method for deploying text locating algorithms as interactive web applications, and as REST-based web services. We believe that web-based deployment of these algorithms is of great importance in order to speed up the way that new ideas are propagated and assimilated in the research community. It offers faster and easier access to newly developed programs than would be possible with other means.

The system is reasonably efficient, and stores all the results in XML files, and as JPEG images with the rectangles overlaid. The XML results can later be processed by a variety of evaluation algorithms, and the overlaid images can be visually inspected using a web browser.

The system is freely available, and we encourage researchers to use it to make their algorithms accessible in this way. Not only would this lead to more rapid dissemination and evaluation of new algorithms and their implementations, but it would also enable new systems to be constructed from a range of web-based components, by merely specifying the connections between the URIs of those components.

Figure 3. Sample image of shop front that has been uploaded to the Alex Chen text locating algorithm. Only a cropped version of the image is shown to save space, but the retrieval rate is very good, though there are a few false positives (which will lower the precision).

Acknowledgments

We thank Alex Chen for his cooperation in allowing us to install his text locating algorithm in our text locating web service.

References

[1] G. Bieber and J. Carpenter. Introduction to service-oriented programming (rev 2.1). `http://www.openwings.org/download/specs/ServiceOrientedIntroduction.pdf` .

[2] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages II:366 – II:373, 2004.

[3] X. Chen and A. L. Yuille. A time efficient cascade for real-time object detection: with applications for the visually impaired. In *Proceedings of the CVAVI05, IEEE Conference on Computer Vision and Pattern Recognition Workshop*, page to appear, 2005.

[4] R. L. Costello. Building web services the rest way, Accessed May 2005. http://www.xfront.com/REST-Web-Services.html.

[5] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, 2000. University of California at Irvine.

[6] S. Lucas. Web-based evaluation and deployment of pattern recognizers. *Proceedings of International Conference on Pattern Recognition*, pages 419 –422, 2002.

[7] S. Lucas and K. Sarampalis. Automatic evaluation of algorithms over the internet. *Proceedings of International Conference on Pattern Recognition*, 4:434–437, 2000.

[8] S. M. Lucas and et al. Icdar 2003 robust reading competitions: Entries, results, and future directions. *International Journal of Document Analysis and Recognition*, page to appear, 2005.

[9] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain. Fvc2000: Fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:402 – 412, (2002).

[10] D. Stork. The open mind initiative. `http://www.openmind.org`.

Poster Papers

Pattern Classification Using Weighted Average Patterns of Categorical k-Nearest Neighbors

Yu Takigawa, Seiji Hotta, Senya Kiyasu, and Sueharu Miyahara
Department of Computer and Information Sciences, Nagasaki University, Japan
{hotta}@cis.nagasaki-u.ac.jp

Abstract

The recognition rate of the typical nonparametric method "k-nearest neighbor rule (kNN)" is degraded when the dimensionality of feature vectors is large. For reducing this difficulty, Mitani and Hamamoto have proposed a simple and strong classifier that outputs the class of a test sample by measuring the distance between the test sample and the average patterns, which are calculated using k-nearest neighbors belonging to each class. On the other hand, it is well known that distance-weighted kNN can improve its error rate due to robustness against outliers. Hence we propose a distance-weighted Mitani's classifier for improving error rates. In addition, we show how to apply kernel methods to our method. The performance of those methods is verified by experiments with handwritten digit patterns and binary classification problems.

1 Introduction

The nonparametric method of pattern recognition named k-nearest neighbor rule (kNN) is implemented on many pattern recognition systems because of its good performance and simple algorithm. The kNN rule determines the class of a test sample by voting of k-closest training samples. The main drawback in kNN is that recognition rates deteriorate when the dimensionality of feature vectors is large [1]. For overcoming this drawback, Mitani and Hamamoto have proposed the classifier that outputs the class of a test sample by measuring the distance between the test sample and the average patterns, which are calculated using k-nearest neighbors belonging to individual classes [2]. Hotta *et al*. have verified by experiments that this classifier in many cases outperformed other classifiers such as kNN and linear subspace methods [3]. In this paper, we term the Mitani's method a *CAP* classifier (classification using Categorical Average Patterns).

On the other hand, Dudani has presented the outline of distance-weighted kNN (WkNN). That is, training samples with smaller distance from a test sample are voted more heavily than ones with larger distance [4]. The WkNN rule can in some cases outperform an unweighted kNN rule when the size of training sets is finite [5]. According to this fact, it is expected that we can improve the recognition rates of CAP if we apply weighting-functions to CAP.

This paper presents a classifier that outputs the class of a test sample by measuring the distance between the test sample and the weighted average patterns, which are calculated using the categorical k-nearest neighbors and their distance values. In addition, we show how to apply kernel methods to the proposed classifier. The performance of those methods is verified by experiments with handwritten digit patterns and binary classification problems.

2 Classification Using Weighted Categorical Average Patterns

Before presenting the proposed method, we review a CAP classifier (i.e., Mitani's classifier) [2, 3] and WkNN [4].

2.1 CAP classifier

First, the procedure of CAP is explained intuitively. Figure 1 illustrates a test sample and its five nearest training samples of each class (only classes 3, 5 and 8 are shown). The CAP classifier computes the average patterns of the k-nearest neighbors for each class (see the rightmost in Figure 1). In this case, all weights of k-nearest neighbors are the same value $1/k = 1/5$. As shown in the rightmost in Figure 1, it seems that the

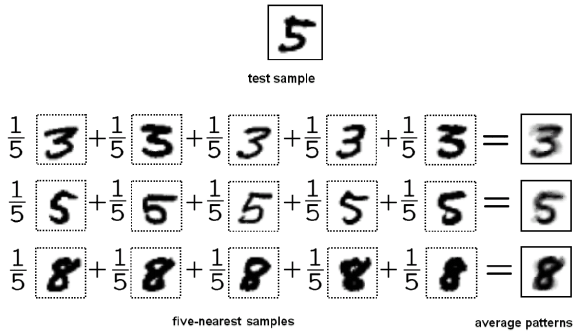


Figure 1. Outline of a CAP classifier: The top pattern is a test sample and its five nearest training samples are shown from the second row to bottom (only classes 3, 5 and 8 are shown). At the rightmost column are the average patterns of each class.

average pattern of class 5 is similar to the test sample, but other average patterns are not. In fact, if k was more than 1 the distance value between the test sample and the average pattern of class 5 was smallest, and the distance values of classes 3 and 8 were never less than that of class 5 [3]. Consequently, the CAP classifier exploits distance between a test sample and average patterns of each class for classification.

Next, the CAP classifier is formulated. Let $\boldsymbol{x}_i^j = [x_{i1}^j, ..., x_{id}^j]^T (i = 1, ..., N_j)$ be the d-dimensional training sample belonging to a class j, where N_j is the number of training samples belonging to a class j. When a test sample $\boldsymbol{q} = [q_1, ..., q_d]^T$ is given, the class of the test sample (denoted by ω) is determined by

$$\omega = \arg\min_j \left\{ \left\| \frac{1}{k} \sum_{i \in X_j} \boldsymbol{x}_i^j - \boldsymbol{q} \right\|^2 \right\}, \qquad (1)$$

where X_j is the set of the k-nearest training samples which belong to a class j. The following relationship is established between the individual samples of X_j:

$$\|\boldsymbol{x}_1^j - \boldsymbol{q}\| \leq \|\boldsymbol{x}_2^j - \boldsymbol{q}\| \leq ... \leq \|\boldsymbol{x}_k^j - \boldsymbol{q}\|. \qquad (2)$$

In short, the class that minimizes the distance between its average pattern ($\sum_{i \in X_j} \boldsymbol{x}_i^j / k$) and the test sample $\boldsymbol{q}$ is outputted as the class of the test sample. Note that CAP coincides with a nearest neighbor rule and a minimum distance method when $k = 1$ and $k = N_j$, respectively. The main shortcoming of CAP is that neighbors with large distance values deteriorate the average patterns when the value of k is large. This is attributed to the fact that all weights of k-nearest samples are equal to $1/k$ independently of their distance values. Actually, the error rates of CAP increase when the value of k is larger than the optimal one [3].

2.2 Distance-Weighted kNN rule

Next, we explain about the outline of the Distance-Weighted kNN rule (WkNN). Let w_i be the weight of the ith nearest samples. The WkNN rule determines the weight w_i by using a function of distance between the test sample and the ith nearest neighbor i.e., samples with smaller distance are weighted more heavily than ones with larger distance. Dudani has proposed the following simple function that scales weights linearly [4]:

$$w_i = \begin{cases} 1 & \text{if } d_k = d_1 \\ \dfrac{d_k - d_i}{d_k - d_1} & \text{if } d_k \neq d_1 \end{cases} \qquad (3)$$

where d_i is the distance to the test sample of the ith nearest neighbor, and d_1 and d_k indicate the distance of the nearest neighbor and the farthest (kth) neighbor respectively. Dudani has further proposed an *inverse distance weighting* function

$$w_i = \begin{cases} \dfrac{1}{d_i} & \text{if } d_i \neq 0 \end{cases} \qquad (4)$$

and a *rank weighting* function

$$w_i = k - i + 1. \qquad (5)$$

These weights are used as the value of one vote in the WkNN rule [4], but we use these functions for computing the weights of k-nearest samples belonging to individual classes.

3 Distance-Weighted CAP classifier

For overcoming the difficulty found on a CAP classifier, we employ the idea of WkNN to modify the weights of k-nearest samples. Let w_i^j be a weight of the ith nearest samples of a class j. First, the k-nearest samples of a test sample $\boldsymbol{q}$ are extracted from each

class by using $d_i^j = \|\boldsymbol{x}_i^j - \boldsymbol{q}\|$. Next, the weight w_i^j is calculated by one of the above weighting-functions (e.g. $w_i^j = 1/d_i^j$). Then the weight is normalized by $w_i^j = w_i^j / \sum_{l=1}^{k} w_l^j$ for computing the weighted average pattern of a class j. Finally, the proposed method determines the class of a test sample by the following classification rule:

$$\omega = \arg\min_j \left\{ \left\| \sum_{i \in X_j} w_i^j \boldsymbol{x}_i^j - \boldsymbol{q} \right\|^2 \right\}. \quad (6)$$

In this paper, we term this method *WCAP* (classification using Weighted Categorical Average Patterns).

3.1 Kernel WCAP

In recent years much research has been conducted on a kernel method (e.g. [6, 7]), to which WCAP described above can be applied. If we define appropriate kernel functions between structured data such as a tree or a graph, WCAP can be applied to such structured data. In addition, if we can use kernelized WCAP, recognition rates will be improved by using an appropriate kernel function for a specific problem. Therefore, kernelization is necessary for general use of classifiers. When a test sample $\boldsymbol{q}$ is given, the kernelized WCAP rule determines the class of the test sample by

$$\omega = \arg\min_j \left\{ \left\| \sum_{i \in X_j} w_i^j \Phi(\boldsymbol{x}_i^j) - \Phi(\boldsymbol{q}) \right\|^2 \right\}, \quad (7)$$

where $\Phi(\cdot)$ is a mapping function that maps samples from an input space to a high-dimensional space, and X_j is the set of the k-nearest training samples in high-dimensional space that belong to a class j. We can represent an inner product in the high-dimensional space $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle$ by an appropriate Mercer kernel $K(\boldsymbol{x}, \boldsymbol{y})$, so the squared Euclidean distance between the test sample $\boldsymbol{q}$ and the training sample $\boldsymbol{x}_i^j$ in the high-dimensional space is written as

$$\begin{aligned} d_i^j &= \|\Phi(\boldsymbol{x}_i^j) - \Phi(\boldsymbol{q})\| \\ &= \sqrt{\langle \Phi(\boldsymbol{x}_i^j), \Phi(\boldsymbol{x}_i^j) \rangle - 2\langle \Phi(\boldsymbol{x}_i^j), \Phi(\boldsymbol{q}) \rangle + \langle \Phi(\boldsymbol{q}), \Phi(\boldsymbol{q}) \rangle} \\ &= \sqrt{K(\boldsymbol{x}_i^j, \boldsymbol{x}_i^j) - 2K(\boldsymbol{x}_i^j, \boldsymbol{q}) + K(\boldsymbol{q}, \boldsymbol{q})}. \end{aligned} \quad (8)$$

In the same way, Equation (7) can be expanded as

$$\begin{aligned} &\left\| \sum_{i \in X_j} w_i^j \Phi(\boldsymbol{x}_i^j) - \Phi(\boldsymbol{q}) \right\|^2 \\ &= \sum_{l,m \in X_j} w_l^j w_m^j K(\boldsymbol{x}_l^j, \boldsymbol{x}_m^j) - 2 \sum_{i \in X_j} w_i^j K(\boldsymbol{x}_i^j, \boldsymbol{q}) \\ &+ K(\boldsymbol{q}, \boldsymbol{q}). \end{aligned} \quad (9)$$

Hence, the class that minimizes the above equation value is outputted as the class of the test sample. In this paper, we term this method *KWCAP* (Kernel WCAP).

4 Experiments

4.1 Experimental results on handwritten digit data

We first have done experiments with the handwritten digit datasets MNIST [8] and USPS [9]. The MNIST dataset consists of 60,000 training and 10,000 test images. The USPS dataset consists of 7,291 training and 2,007 test images. It is well known that the USPS dataset is more difficult to recognize than MNIST because USPS consists of little and mislabeled training samples. For feature extraction, we extracted *local stroke direction* feature [10]. The local stroke direction technique divides each digit pattern into a 8×8 grid and assigns each pixel the direction (vertical, horizontal, diagonal right, and diagonal left) of the vector that covers the maximum number of consecutive black pixels. The numbers of pixels in each grid cell that are assigned each direction are output. This feature set represents each digit pattern as a 256 dimensional vector. We show an example of local stroke feature description in Figure 2. In addition, we use the Gaussian kernel as a kernel function in experiments:

$$K(\boldsymbol{x}_i^j, \boldsymbol{q}) = e^{-\alpha \|\boldsymbol{x}_i^j - \boldsymbol{q}\|^2}. \quad (10)$$

4.1.1 Influence of weighting-functions on error rates

First, the influence of weighting-functions on error rates of WCAP was examined by using the MNIST dataset. Figure 3 and Figure 4 show the results of test and training error rates on each weighting-function, respectively. The results of KWCAP are not included in

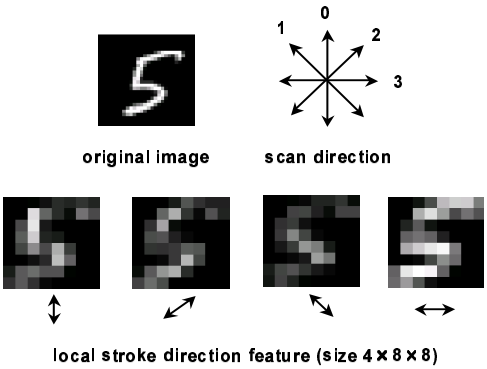


Figure 2. Example of local stroke direction feature.

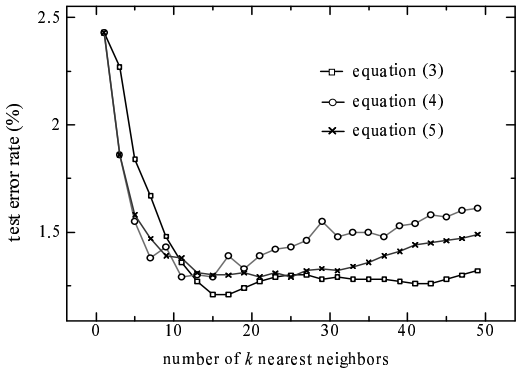


Figure 3. The test error rates of WCAP on each weighting-function.

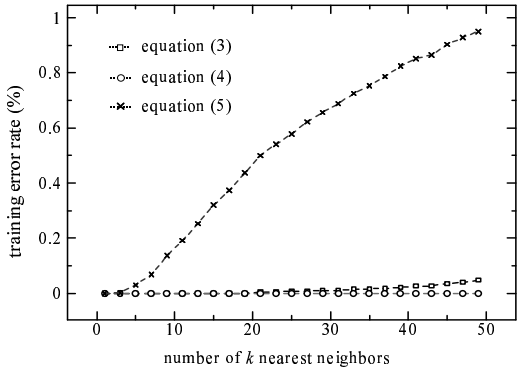


Figure 4. The training error rates of WCAP on each weighting-function.

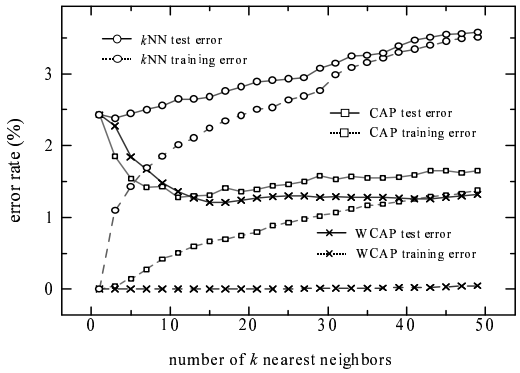


Figure 5. Relationship between k and error rates.

these figures, because they were almost same as those of WCAP. As shown in Figure 3, the test error rate obtained by Equation (3) was lower than those of Equation (4) and Equation (5) in most range of k. On the other hand, as shown in Figure 4, the training error rates obtained by Equation (3) and Equation (4) were equal to zero in most range of k. Hence we exploit Equation (3) for computing the weights of nearest samples in future experiments, which gives the lowest error rates on both test and training samples.

4.1.2 Influence of parameter k on error rates

Next, we investigated the relationship between parameter k and error rates by using the MNIST dataset. Figure 5 shows the results of kNN, CAP and WCAP. The result of KWCAP is not included in this figure, because it was almost same as that of WCAP. As shown in this figure, the error rates of kNN against test and training samples increased as k increased. In contrast, the test errors of CAP and WCAP decreased while k was less than or equal to about 15. Note that the test error rate of WCAP was almost flat while k was more than 15. In contrast, the test error rate of CAP slightly increased after $k = 15$. In addition, the training error rate of WCAP was equal to zero in most range of k, but that of CAP increased as k increased.

On the other hand, Figure 6 shows the cross-validation (leave-one-out method) curve of CAP and WCAP for the MNIST dataset. As shown in this figure, the estimated error rates of WCAP was almost flat while k was greater than 15. In contrast, the estimated

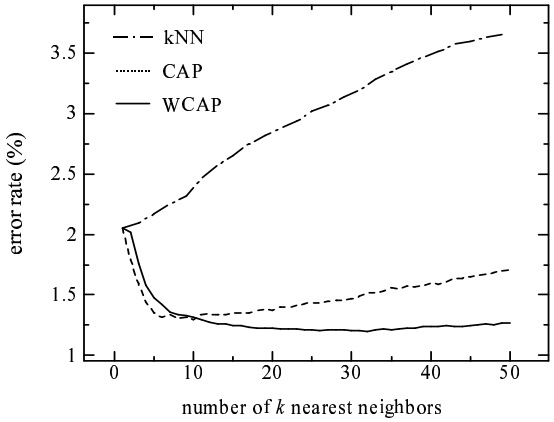


Figure 6. Cross-validation curves of CAP and WCAP on the MNIST dataset.

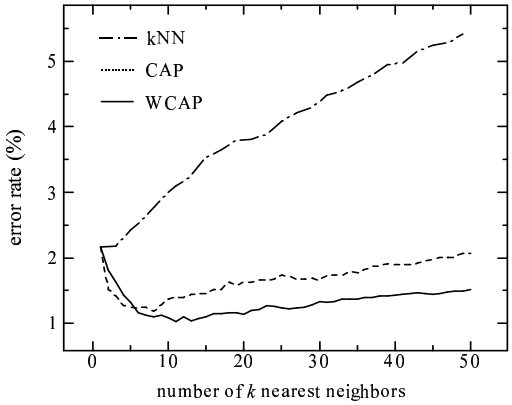


Figure 7. Cross-validation curves of CAP and WCAP on the USPS dataset.

Table 1. Error Rates on MNIST

method	test [%]	training [%]
kNN ($k = 5$)	2.39	1.43
WkNN ($k = 11$)	2.12	0.12
CLAFIC ($k = 30$)	3.68	3.87
SVM ($\alpha = 180$, $C = 10$)	0.92	0.01
CAP ($k = 11$)	1.28	0.50
KCAP ($k = 11$, $\alpha = 70$)	1.27	0.37
WCAP ($k = 15$)	1.21	0
KWCAP ($k = 15, \alpha = 70$)	1.21	0

Table 2. Error Rates on USPS

method	test [%]	training [%]
kNN ($k = 1$)	5.2	0
WkNN ($k = 5$)	4.73	0
CLAFIC ($k = 15$)	4.73	1.71
SVM ($\alpha = 260$, $C = 3$)	3.69	0.03
CAP ($k = 12$)	3.54	0.59
KCAP ($k = 12$, $\alpha = 130$)	3.44	0.43
WCAP ($k = 14$)	3.44	0.03
KWCAP ($k = 16$, $\alpha = 130$)	3.34	0.03

error rate of CAP increased after k=10. Similarly, the estimated error rate of kNN increased drastically after k=1. Hence selection of k on WCAP is easier than those on kNN and CAP. According to these results, it is thought that advantages of WCAP are obtained by using weighting-functions that are for robustness against outliers.

4.2 Experimental results on benchmark datasets

4.2.1 Comparison with other classifiers

Table 1 shows the lowest error rates on MNIST with parameter values of *each classifier*: kNN, WkNN with Equation (3), the basic linear subspace method *CLAFIC* [11], Support Vector Machine (SVM) with the Gaussian kernel, CAP, Kernel CAP (KCAP) [3], WCAP and KWCAP. The parameter k in CLAFIC indicates the dimensionality of subspaces. The parameter α indicates the scale parameter of the Gaussian kernel (see Equation (10)). The parameter C in SVM indicates the soft margin constant. For SVM, we used the SVM package, *LIBSVM* [12]. As shown in Table 1, SVM outperformed all the other investigated techniques, and the test error rates of WCAP and KWCAP were lower than those of kNN, WkNN, CLAFIC, CAP and KCAP.

Table 2 shows the lowest error rates on USPS with parameter values of each classifier. As shown in this table, the proposed methods WCAP and KWCAP outperformed the other classifiers even if the number of training samples is small and training set contains mislabeled samples. Figure 7 shows the cross-validation (leave-one-out method) curve of CAP and WCAP for

Table 3. Confusion matrix of SVM on USPS.

	class 0	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9
class 0	352	0	2	0	2	1	1	0	1	0
class 1	0	257	0	0	5	0	2	0	0	0
class 2	3	0	190	1	2	0	0	1	1	0
class 3	0	0	3	153	1	8	0	0	1	0
class 4	0	1	2	0	192	0	1	1	0	3
class 5	2	1	1	1	0	154	0	0	0	1
class 6	1	1	1	0	3	0	164	0	0	0
class 7	0	1	1	0	6	0	0	138	1	0
class 8	2	1	1	0	1	1	0	0	158	2
class 9	0	0	0	0	1	0	0	0	1	175

Table 4. Confusion matrix of WCAP on USPS.

	class 0	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9
class 0	355	0	2	0	0	1	1	0	0	0
class 1	0	257	0	0	5	0	2	0	0	0
class 2	0	0	193	3	0	0	0	1	1	0
class 3	2	0	2	148	0	13	0	0	1	0
class 4	0	2	3	0	190	1	0	0	0	4
class 5	2	1	1	1	0	152	0	0	2	1
class 6	1	0	0	0	1	0	168	0	0	0
class 7	0	1	1	0	3	0	0	141	1	0
class 8	2	2	1	1	0	2	0	0	158	0
class 9	0	0	0	0	0	0	0	1	1	175

the USPS dataset. As shown in this figure, the parameter k that obtained minimum estimated error rates of WCAP was almost equal to the optimal value on test samples. In contrast, the parameter k that obtained the minimum estimated error rate of CAP was smaller than the optimal one. Hence selection of k on WCAP is easier than those on kNN and CAP even if the number of training samples is small.

4.2.2 Confusion matrix on USPS

Next, we show confusion matrices of SVM and WCAP for visualization performance of each classifier. Table 3 and Table 4 present confusion matrices of SVM and WCAP on USPS, respectively. Each column of matrices represents predicted classes, while each row represents actual classes. As shown these tables, the number of combination of mislabeling of SVM is 42, but that of WCAP is 35 even if the difference of the number of misclassified patterns is 4. In other words, WCAP is confusing two classes less than SVM. This property is desired one for improving accuracy of classifiers.

4.2.3 Experimental results on binary classification problems

Finally we tested the proposed method on 13 benchmark datasets of binary classification problems [13, 14]. Each benchmark consists of 100 (or 20) random partitions of data for form test and training sets. Table 5 (see the last page) shows the lowest average test error rates and its standard deviations of *each classifier*: kNN, WkNN, CAP, KCAP, WCAP and KWCAP. Table 6 shows parameters of each classifier optimized

on test error rates[1]. For comparison with other classifiers, see [13, 14]. The lowest average error rates are shown in boldface type, and the second ones are shown in italic type. As shown in this table, the results of CAP and WCAP were better than those of kNN and WkNN in some cases. In addition, the results of KCAP and KWCAP were better than those of CAP and WCAP in many cases. That is, the use of kernel method helped improve performance of CAP and WCAP.

5 Conclusions

This paper has presented the algorithm of Weighted-CAP (WCAP) that outputs the class of a test sample by measuring the distance between the test sample and the weighted average patterns, which are calculated using categorical k-nearest neighbors and their distance values. The weighting-functions that were proposed by Dudani were used for reducing the influence of samples with large distance on average patterns. In addition, we showed how to apply kernel methods to WCAP for improving the recognition performance. It was verified by experiments that WCAP was often superior to un-weighted CAP and kernel methods helped improve the recognition performance of WCAP.

In short, the proposed method includes the following advantages: 1) The proposed methods can achieve lower error rates than other nonparametric methods such as kNN, subspace methods and a un-weighted CAP classifier. 2) The proposed method can achieve low error rates even if the dimensionality of feature vectors is large. Hence, it is possible to improve recognition rates by employing kernel methods to WCAP. 3) We can implement CAP and KCAP easily because of its simple algorithms. 4) There is no need to reconstruct systems when samples are added. Future work will include speeding up and theoretical explanation of the proposed method.

Acknowledgments This research was supported by the Ministry of Education, Culture, Sports, Science and Technology in Japan under a Grant-in-Aid for Scientific Research C (No. 17500115).

[1]These parameters should be estimated by using a statistic estimator such as a cross-validation method. In this paper, however, their values were determined simply by using a rough searching.

References

[1] K. Fukunaga. Bias of nearest neighbor error estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(1):103–112, 1987.

[2] Y. Mitani and Y. Hamamoto. Classifier design based on the use of nearest neighbor samples. *Proc. of 15th Int. Conf. Patt. Recog.*, 2:773–776, 2000.

[3] S. Hotta, S. Kiyasu, and S. Miyahara. Pattern recognition using average patterns of categorical k-nearest neighbors. *Proc. of 17th Int. Conf. Patt. Recog.*, 4:412–415, 2004.

[4] S.A. Dudani. The distance-weighted k-nearest neighbor rule. *IEEE trans. on systems, man and cybernetics*, 6(4):325–327, 1976.

[5] J.E.S. Macleod, A.Luk, and D.M. Titterington. A re-examination of the distance-weighted k-nearest neighbor classification rule. *IEEE trans. on systems, man and cybernetics*, 17(4):689–696, 1987.

[6] V. Vapnik. *Statistical learning theory*. John Wiley and Sons, 1998.

[7] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12(2):181–201, Mar. 2001.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Intell. Signal Process.*, 306–351, 2001.

[9] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[10] T. Akiyama and N. Hagita. Automated entry system for printed documents. *Patt. Recog.*, 23(11):1141–1154, 1990.

[11] S. Watanabe, and N. Pakvasa. Subspace method in pattern recognition. *Proc. 1st Int. Joint Conf. on Patt. Recog.*, Washington DC, 2–32, 1973.

[12] C.C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. 2001, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

[13] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing*, 36(10):41–48, 1999.

[14] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001.

Table 5. Error rates [%] and its standard deviations

dataset	# of dimension	kNN	WkNN	CAP	WCAP	KCAP	KWCAP
Banana	2	11.3 ± 0.6	11.0 ± 0.5	11.8 ± 0.5	11.4 ± 0.6	$\mathit{10.7 \pm 0.5}$	$\mathbf{10.6 \pm 0.4}$
B. Cancer	9	$\mathit{25.3 \pm 4.0}$	$\mathbf{25.0 \pm 3.9}$	26.5 ± 4.5	26.0 ± 4.5	25.9 ± 4.4	25.9 ± 4.5
Diabetes	8	25.1 ± 1.7	25.1 ± 1.8	24.5 ± 1.8	24.5 ± 1.7	$\mathbf{23.7 \pm 1.9}$	$\mathit{24.3 \pm 1.6}$
German	20	25.2 ± 2.3	24.8 ± 2.5	24.6 ± 2.3	$\mathbf{24.3 \pm 2.1}$	24.4 ± 2.5	$\mathbf{24.2 \pm 2.0}$
Heart	13	$\mathbf{15.7 \pm 3.3}$	16.1 ± 3.4	$\mathit{15.9 \pm 3.4}$	17.3 ± 3.3	16.1 ± 3.5	17.3 ± 3.3
Image	18	3.4 ± 0.5	3.4 ± 0.5	3.3 ± 0.6	$\mathbf{3.1 \pm 0.5}$	3.3 ± 0.6	$\mathbf{3.1 \pm 0.5}$
Ringnorm	20	35.0 ± 1.4	35.0 ± 1.4	12.0 ± 0.8	12.4 ± 1.0	$\mathbf{1.4 \pm 0.1}$	$\mathit{1.6 \pm 0.1}$
F. Sonar	9	34.8 ± 1.9	34.8 ± 1.8	$\mathbf{34.4 \pm 1.7}$	34.8 ± 1.6	$\mathbf{34.4 \pm 1.7}$	34.8 ± 1.6
Splice	60	26.2 ± 2.1	24.9 ± 2.3	13.5 ± 0.8	$\mathit{12.6 \pm 0.8}$	12.9 ± 0.7	$\mathbf{12.5 \pm 0.9}$
Thyroid	5	4.4 ± 2.2	4.4 ± 2.2	4.4 ± 2.2	4.4 ± 2.2	$\mathit{4.2 \pm 2.1}$	$\mathbf{4.0 \pm 2.3}$
Titanic	3	22.8 ± 1.1	22.8 ± 0.7	23.1 ± 1.9	$\mathit{22.7 \pm 1.3}$	22.8 ± 1.5	$\mathbf{22.5 \pm 1.7}$
Twonorm	20	2.5 ± 0.2	2.5 ± 0.2	$\mathbf{2.4 \pm 0.1}$	$\mathbf{2.4 \pm 0.1}$	$\mathbf{2.4 \pm 0.1}$	$\mathbf{2.4 \pm 0.1}$
Waveform	21	10.7 ± 1.0	10.4 ± 1.1	$\mathit{10.2 \pm 0.5}$	10.3 ± 0.5	$\mathbf{9.9 \pm 0.6}$	10.3 ± 0.4
# of boldface		1	1	2	3	5	7
# of Italics		1	0	2	3	2	2

Table 6. Parameter values on each classifier.

	kNN	WkNN	CAP	WCAP	KCAP		KWCAP	
dataset	k	k	k	k	k	α	k	α
Banana	11	23	8	14	15	1.4	16	0.8
B. Cancer	17	39	27	50	27	0.09	50	0.007
Diabetes	35	40	102	62	102	0.5	104	0.05
German	13	33	44	48	42	0.05	59	0.005
Heart	35	47	68	60	67	10^{-4}	60	10^{-4}
Image	1	1	2	3	2	10^{-4}	3	10^{-4}
Ringnorm	1	1	4	7	93	0.1	15	1.4
F. Sonar	37	269	254	205	246	10^{-3}	205	10^{-3}
Splice	9	20	92	193	92	0.01	202	10^{-3}
Thyroid	1	1	1	1	9	1	4	0.25
Titanic	25	24	22	29	22	0.5	29	0.05
Twonorm	95	167	114	128	177	0.1	177	0.1
Waveform	41	83	36	50	40	0.05	41	0.05

A Recursive Approach For Bleed-Through Removal

F. DRIRA H. EMPTOZ

LIRIS - INSA de LYON - Bât Jules Verne
20 avenue Albert Einstein 69621 Villeurbanne Cedex France
{fdrira, hubert.emptoz}@liris.cnrs.fr

Abstract

Historical documents are valuable resources worth to be preserved in order to support our cultural and social knowledge. Unfortunately, these supports based on fragile materials are often affected by several types of degradations. Applying restoration techniques on degraded captured digital images of historical documents may be a quick and efficient way to preserve the document and avoid the loss in its content.

This paper presents a new method to restore a particular type of degradation which is referred to as "bleed-through". This degradation is caused by the interference of characters from the reverse side with the text to be read. Our proposed method is based on a recursive approach that relies on two types of analysis: the Principal Component Analysis and the k-means clustering algorithm. The aim here is to extract clear textural images from these interfering and overlapping areas of text. Our restoration method analyses the front side image alone and corrects the unneeded image components. This paper concludes with some experimental results that demonstrate the effectiveness of our proposed method.

1. Introduction

The advance of digital technologies and techniques makes it possible to preserve heritage documents for a longer period. This digital mean of preservation would also enable a widespread diffusion of these documents. Indeed, recent techniques help in producing digital copies of original heritage documents. However, the quality of these digital copies depends greatly on the quality of the original ancient documents. These are often affected by several types of degradations. Baird [1] suggests the following definition of the term degradation (or defect): "every sort of less-than ideal properties of real document images". In fact, old documents, supported by fragile materials, are easily affected by bad environmental conditions. Manipulations, humidity and unfitted storage for many years affect heritage documents and make them difficult to read. Moreover, the digitizing techniques used in image scanning inevitably further degrade the quality of the document images. Indeed, degradations affect ancient documents and make them difficult to read. Resorting to restoration techniques for these deteriorated old documents becomes an increasingly urgent need. Restoration refers to the treatment of a low quality historical document. Restoration techniques can improve the quality of the digital copy of the originally degraded document, thus improving human readability and allowing further application of image processing techniques such as segmentation and character recognition. A large number of algorithms have been developed by the community. However, each of these methods depends on a certain context of use and is intended to process a precise type of defects.

In this study, we will focus on a particular type of degradation, which is referred to as "bleed-through". This degradation is due to ink's seeping through the pages of documents after long periods of storage. The result is that characters from the reverse side appear as noise on the front side. This can deteriorate the legibility of the document if the interference acts in a significant way. An overview of some restoration techniques tackling this kind of degradation is presented in the first section. In the second section, we propose a new algorithm trying to restore such kind of degraded documents by extracting clear textual images from interfering and overlapping areas. The main idea behind our algorithm is to classify each pixel of the page to be processed in one of the following three classes: (1) background, (2) original text, or (3) interfering text. Our problem is therefore close to a segmentation problem. We propose to perform, recursively, a k-means algorithm on the dimensionally reduced image data. This dimension reduction is done through Principal Component Analysis (PCA). Our recursive restoration method does not require specific input devices or the digital processing of the backside to be input. It is able to correct unneeded image components through analysis of the front side image

alone. The third section shows experimental results that verify the effectiveness of our proposed method.

2. Brief review of bleed-through restoration techniques

Thresholding techniques are a simple possibility but remain insufficient for too degraded documents. For instance, Leedham and al. [2] compared several thresholding techniques for separating text and background in degraded, historical documents. The results prove that neither global nor local thresholding techniques perform satisfactorily. Indeed, many restoration approaches dealing with "bleed-through" removal were proposed. Some of them have successfully resolved this problem but under specific conditions. These methods can be divided into two classes according to the presence of the verso side page document: non-blind ones—treating this interference problem using both sides of the document— and blind ones treating this problem without the verso side.

2.1. Non-blind restoration techniques

The main idea of non-blind approaches is mainly based on the comparison between the front and back page, which requires a registration of two sides of the document in order to identify the interfering strokes to be eliminated. Techniques of this type are reported in [3, 4, 5] for scanned documents. Sharma's approach [3] simplifies the physical model of these effects to derive a linear mathematical model and then defines an adaptive linear-filtering scheme. Another approach proposed by Dubois and Pathak [4] is mainly based on processing both sides of a gray-level manuscript simultaneously using a six-parameter affine transformation to register the two sides. Once the two sides have been correctly registered, areas consisting primarily of "bleed-through" are identified using a thresholding technique and replaced by the background color or intensity. In [5], a wavelet reconstruction process is applied to iteratively enhance the foreground strokes and smear the interfering strokes. Doing so strengthens the discriminating capability of an improved Canny edge detector against the interfering strokes.

All these different non-blind restoration techniques dealt successfully with "bleed-through" removal. Nevertheless, a registration process of both sides of the document is required. Perfect registration, however, is difficult to achieve. This is due to (1) different document skews and (2) different resolutions during image capture of both sides, (3) non-availability of the reverse side and (4) warped pages resulting from the scanning of thick documents. The main drawback of this approach is therefore its dependency on both sides of the documents that must been processed together. Resorting to a blind restoration method, i.e. removing the bleed through without the need of the both sides of the document is often a more interesting solution.

2.2. Blind restoration techniques

A variety of techniques have been proposed in this regard. An interesting approach successfully used is based on steered filters. This approach is especially designed for old handwritten documents. A restoration method [6] proposed by Tan and al. consists in adopting an edge detection algorithm together with an orientation filter to extract the foreground edges and remove the reverse side edges. This algorithm performs well and improves greatly the appearance of the original documents. However, one problem with this method is mainly obtained when the interference is so serious that the edges of the interfering strokes are even stronger than that of the foreground edges. As a result, the edges of the interfering strokes would remain in the resultant text image. Another approach proposed by Wang et al. [7] uses directional wavelets to remove images of interfering strokes. The writing style of the document determines the differences between the orientations of the foreground and the interfering strokes. Basically, the foreground and the interfering strokes are slanting along the directions of 45° and 135° respectively. The directional aspect of the transform is capable of distinguishing the foreground and reverse side strokes and effectively removing the appearing interference. This approach produces very interesting results but it remains applicable only to particular cases of character orientation (45° and 135°). All the techniques cited above treat a particular case of degraded document, where foreground and interfering strokes characters are oriented differently, which is not always the case. Other more flexible techniques exist, among which, we can cite techniques based on Independent Component Analysis [8], adaptive binarization [9], self-organizing maps [10], color analysis [11].

So far, we presented a classification of some methodologies proposed to tackle the "bleed-through" degradation. After this short outline, our choice will directed to a blind restoration method as the verso side is not necessary available.

3. The proposed method

3.1. Justification

As already said, the scanned image of a document, which has been subject to "bleed-through" degradation, contains the content of the original side combined with the content of the reverse side. A representative part of such a degraded document (provided by Chatillon-Chalaronne) is shown in Figure 1. Our problem, illustrated in this figure, is to extract clear text strings of the front side from this noisy background. We propose to proceed with a segmentation approach. In fact, the main idea behind our algorithm is to classify the pixels of the page into three classes: (1) background, (2) original text, or (3) interfering text. This last class must be removed from the original page and replaced by the background color (the average of the detected background pixels for example).

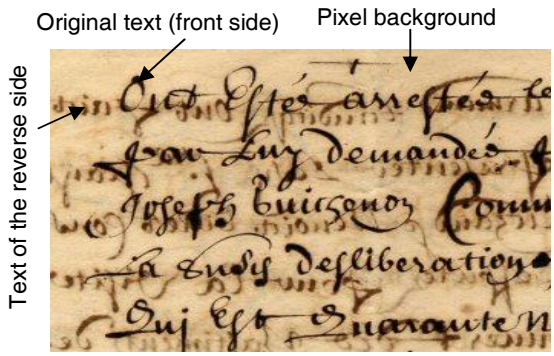


Figure 1: An extract of a degraded document image

"Bleed-through" removal is thus a three-class segmentation problem. Nevertheless, a single clustering step is not sufficient to correctly extract the text of the front side (Figure 2).

Figure 2: Application's Results of the 3-means classification algorithm on a degraded image

Thus, we propose to apply a recursive segmentation method on the reduced data with PCA. To simplify the analysis and reduce its computational complexity, we will restrict ourselves to the case of a two-class problem: original text or not. The proposed method is built then via recursively dividing the test image into two subsets of classes.

The following paragraph will briefly (1) introduce k-means, (2) introduce PCA, and (3) explain the importance of applying k-means on PCA. Our method will be explained in greater detail in the following section (section 4).

(1) k-means is an algorithm [12] using prototypes (centroids) to represent clusters by optimizing the squared error function. The prototypes are initially randomly assigned to a cluster. The k-means clustering proceeds by repeated application of a two-step process where the mean vector for all prototypes in each cluster is computed and then prototypes are reassigned to the cluster whose centre is closest to the prototype. The data points are thus decomposed into disjoint groups such that those belonging to same cluster are similar while others belonging to different clusters are dissimilar.

(2) PCA is an example of eigenvector-based technique which is commonly used for dimensionality reduction and feature extraction of an embedded data. The main justification of dimension reduction is that PCA uses singular value decomposition (SVD) which gives the best low rank approximation to original data. Indeed, PCA can reduce the correlation between the different components where coherent patterns can be detected more clearly.

(3) Applying k-means on PCA: we propose here to apply K-means (K=2) clustering in the Principal Component Analysis (PCA) subspace. Pioneering work [13] has shown that PCA dimension reduction is particularly beneficial for K-means clustering. It was also proven that the continuous solutions of the discrete K-means clustering membership indicators are the data projections on the principal directions (principal eigenvectors of the covariance matrix). More precisely, we decided to apply the segmentation algorithm on image data decorrelated using a PCA. The PCA is computed on the RGB color space. It improves the quality of classification because of its properties which reduce data space and eliminate associations between data. In representing the document image in a convenient vector space, we will succeed to improve the gathering of elements with approximately similar values in order to make them converging to significant classes.

3.2. Description of the method

A new framework based on a recursive approach is presented here, which relies on two types of analysis:

the Principal Component Analysis and the k-means algorithm applied recursively on each generated data image. A scheme of our approach is given in Figure 3. The following steps are performed recursively:

(1) The dimension of an image is reduced and its data is decorrelated using Principal Component Analysis.

(2) The k-means algorithm is applied with parameter k=2, resulting in two classes of image pixels.

(3) The pixels of each class back-projected into the original color space. Each generated class is recursively used as input to the same algorithm beginning with step 1.

The dimension reduction step projects the document image from the original vector space to another reduced subspace generated via PCA. The RGB color space, where each color is represented by a triplet red, green and blue intensity, is used as input.

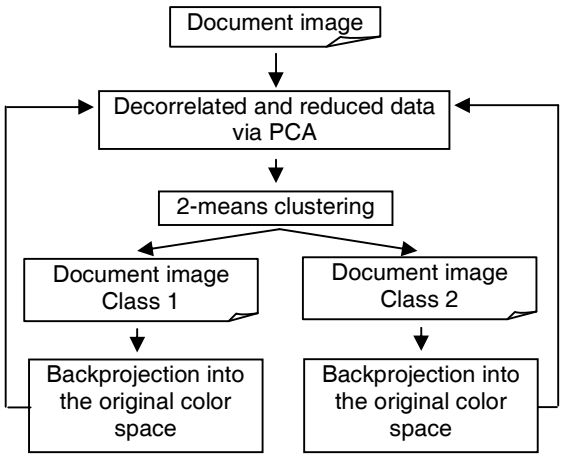


Figure 3: The flowchart of the proposed method

As shown in Figure 4, the first principal component (its eigenvalue represents 99.02% of the total eigenvalues variance) gives a good approximation of the image compared to the other principal components. For instance, when we project onto the directions with biggest variance, we can not only retain as much information as possible but also we can deliberately drop out directions with small variance.

Indeed, selecting the most significant principal components as input to the k-means clustering algorithm reduces the data enough in order to make the problem manageable while at the same time retaining enough information to perform a successful separation. The Figure 5 shows more clearly the difference between the luminance axis and the first component of the PCA. This analysis can better maximize pixel separation projected on its first component.

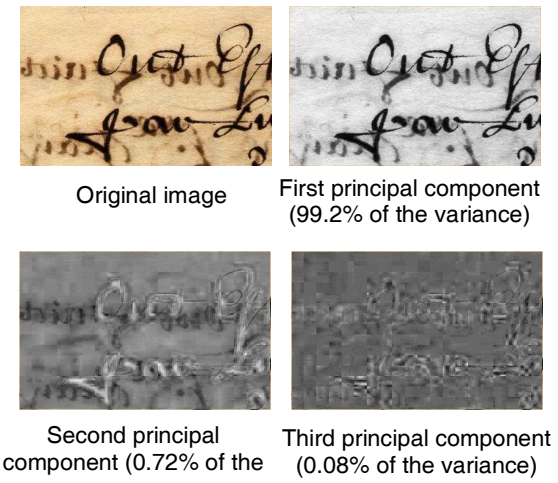


Figure 4: Results of PCA projection

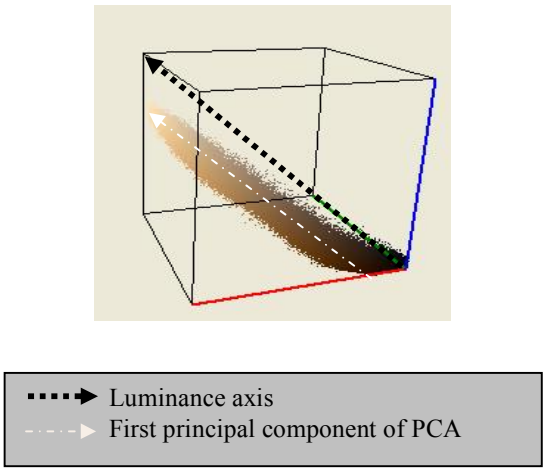


Figure 5: Color distribution of Figure 1 in the RGB cube

This method starts with the whole image set as a single cluster. Then, it is partitioned into disjoint subsets a1 and a2, where the inter-cluster distance is maximized. The subsets a1 and a2 are further subdivided into a11 and a12 and a21 and a22, and so on. The process thus generates a binary tree. One of its leaves represents the expected image which contains the original text. Figure 6 represents an extract of this tree. Only subsets leading to the expected image are represented. As shown in Figure 6, image (a122) is the expected result. The number of iterations in our method has been determined empirically and set to a fixed number of iterations (=3). The result of the algorithm is a set of classes (the leaves of the tree of recursive function calls), where one class represents the pixels of the original handwriting. We can so notice that the

segmentation of the data in a recursive way allows us to refine the final restoration result as soon as we traverse down the binary tree. Our method outperforms other methods that involve a global classification in K classes applied to the entire image (Figure 2). It converges more correctly to the final result.
For the moment, the class containing the original recto side is chosen iteratively by an operator. We are currently working on an automatic detection of this class. While the manual intervention choosing the final result would appear as a weak point in our method, we consider this intervention crucial to preserve the originality of the restored document.

Figure 6: An extract of the generated tree with our proposed method applied on a test image

4. Experimental results and discussion

Experiments were carried out to evaluate the performance of our approach. We used some samples of degraded image documents. An example of a restored image resulting from the application of our method on a degraded document image (Figure 1) is given in Figure 7. This figure shows one of the subsets generated after three iterations of the method. Moreover, this subset represents the front side text and we clearly notice, compared with the test image (Figure 1) that the interfering text has been successfully removed and replaced by the average of the detected background pixels.

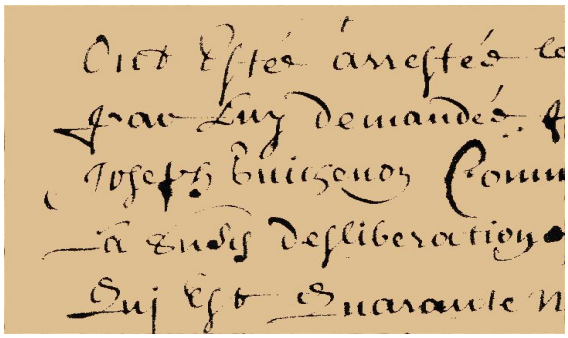


Figure 7: The obtained image with our method applied on figure 1 (after 3 iterations)

We have also validated our method on degraded document images containing black and red text. An example of such document image (provided by the French Institute of Research on Text History (IRHT)) is illustrated in Figure 8. Figures 9 and 10 show the results of the experiment performed on this document image. Those images could be combined to create a restored version (Figure 11) of the degraded image. Other degraded documents (Figure 12, Figure 14 and Figure 16) and their respectively restored versions (Figure 13, Figure 15 and Figure 17) are given below.

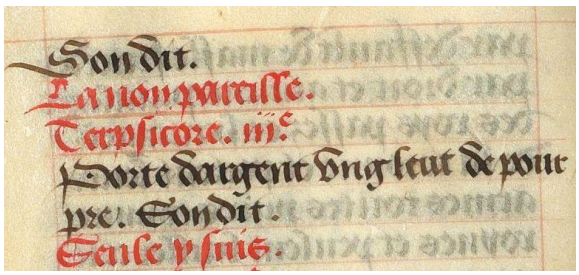


Figure 8: A degraded document image

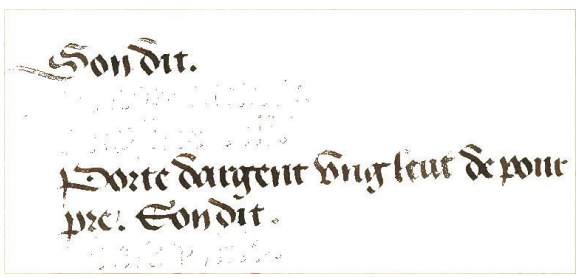


Figure 9: Black text extracted by our method

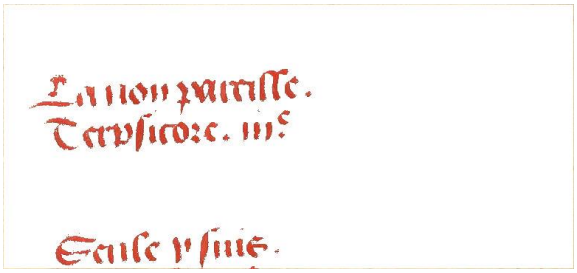


Figure 10: Red text extracted by our method

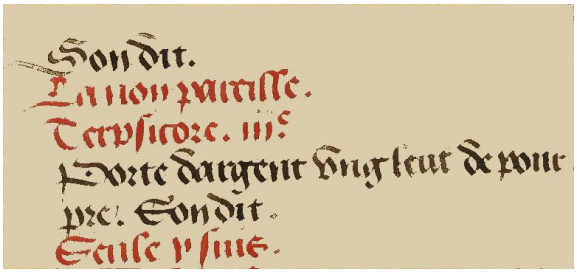


Figure 11: Restored image

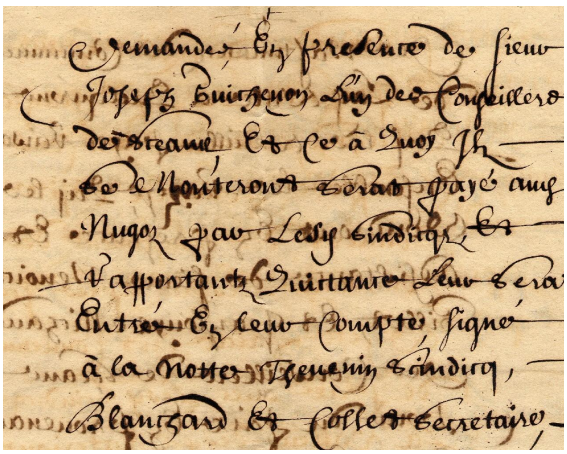


Figure 12: A degraded document image

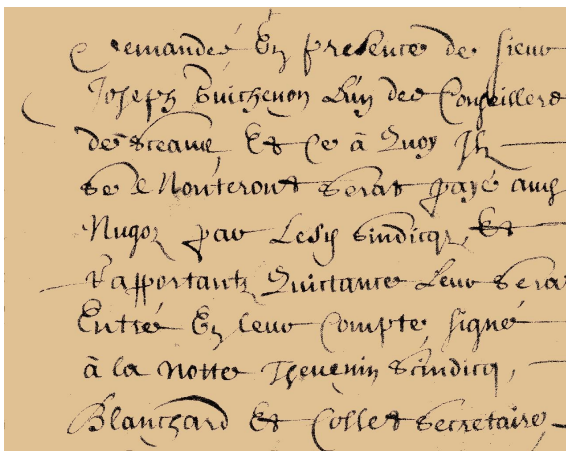


Figure 13: Restored image

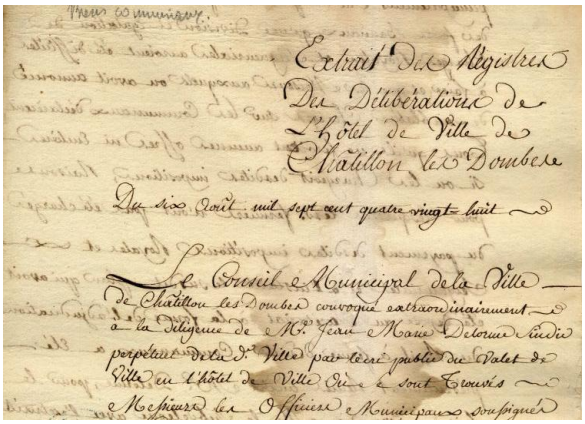


Figure 14: A degraded document image

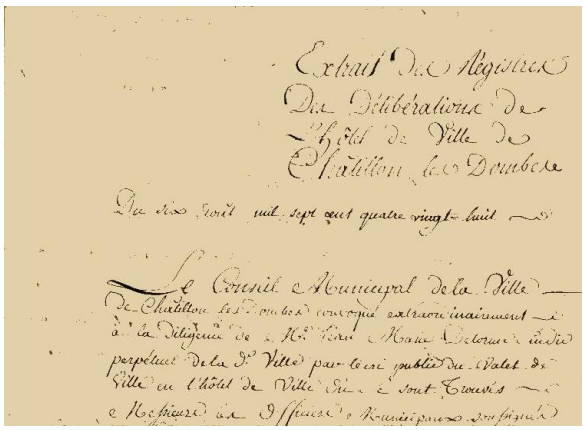


Figure 15: Restored image

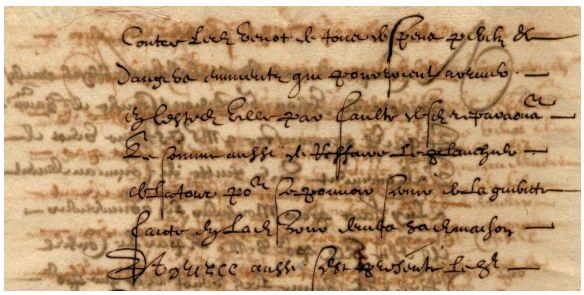


Figure 16: A degraded document image

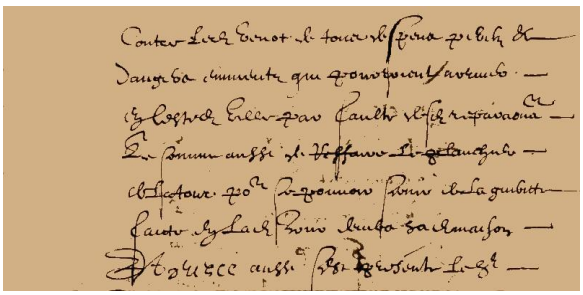


Figure 17: Restored image

Experimental results illustrate the significant performance of this recursive approach, which produces similar results as those obtained with the approach [11] (Figures 18 and 19 are the restored version of an extract of Figure 1). However, it is not hampered by the same restrictions. The approach [11] represents an adaptive segmentation algorithm suited for color document images analysis. It is based on the serialization of the k-means algorithm.

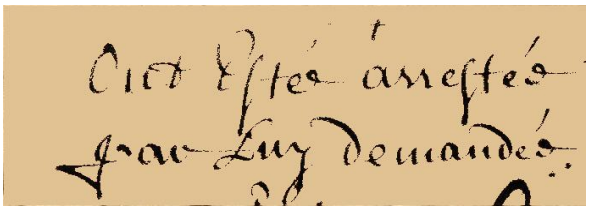


Figure 18: The obtained image with our method

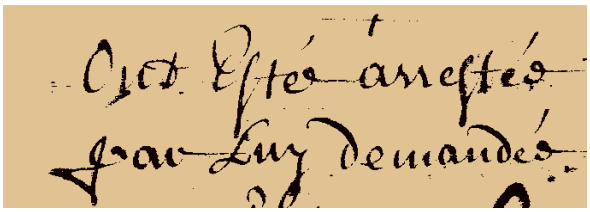


Figure 19: The obtained image with the approach [11]

Compared to other existing methods, our method
(1) does not require specific input devices or the processing of the reverse side of the document to be input. It is able to correct unneeded image components through the analysis of the front side image alone. Our approach can be classified among "blind bleed-through" removal techniques.
(2) does not require any specific learning process such as the case of the self-organizing Maps based approach [10] where a learning process must be performed on each chosen image.
(3) does not require any input parameters as in the case of the serialized k-means based approach. Certainly, this approach gives good results but it is an unsupervised one as the choice of some parameters such as the number of clusters and the color samples for each class are not done automatically.

Nevertheless, one of major weakness of our algorithm resides in its computationally complexity especially for high dimension document images. Indeed, one of our future aims is to obtain a correct front page image with efficient computation and reduced memory demands. This can be done by controlling the recursive decomposition of our original image. In other word, the scope is to allow decomposition only of leaves images that can lead to the corrected expected image. If we suppose that the original text is the darkest one, the analysis of the image histogram values could be a solution. By doing so, we can reach an automatic final class image detection

Moreover, other extensions and refinements of our work will be directed towards the mixture of the obtained image results with further image processing techniques. These techniques could improve the obtained results and help to produce a more readable and visible text. For instance, removal of the interference text may damage the touching characters in the overlapping area. This is due to the fact that some parts of the removed segments possibly belong to both original and interfering text. Broken characters could also been justified by the irregularity of ink color and also the variability of the ink layer's depth over the different characters. This distinguishes handwritten documents from printed ones and makes their treatment more complex. The aim here is to recover broken edges of the words or characters on the front side. Further research will concentrate on these ideas.

5. Conclusion

Both Principal Component Analysis (PCA) and K-means can be combined to make together a powerful tool for image processing tasks. In this paper, they are applied recursively to separate original text from interfering and overlapping areas of text. Experimental results illustrate visual improvement results of digital degraded document images. Certainly, PCA used as a space reduction technique has proven to be powerful as a pre-processing step for the k-means classification algorithm. Nevertheless, the linearity of this transform could limit its application since this transform could not detect at all times the different structures in a given image. Resorting to a suitable nonlinear transform could give better results or decrease the iteration number of the process. Moreover, the choice of the k-means and the PCA, widely used techniques in the literature, represents a first step for testing its relevance. Our future research will investigate other techniques and compare the results with those obtained here to evaluate performances.

6. Bibliography

[1] H. S. Baird, *State of the Art of Document Image Degradation Modelling*, invited talk, IAPR 2000 Workshop on Document Analysis Systems, Brazil, December 2000.

[2] G. Leedham, S. Varma, A. Patankar, V. Govindaraju, *Separating text and background in degraded document images – a comparison of global thresholding techniques for multi-stage thresholding*. In: Proceedings of the 8th international workshop on frontiers in handwriting recognition, pp 244–249, Canada, August 2002,

[3] G. SHARMA, *Cancellation of show-through in duplex scanning*, International Conference on Image Processing (ICIP), vol. 2, pp. 609-612, September 2000.

[4] E. Dubois, A. Pathak, *Reduction of bleed-through in scanned manuscripts documents*, In: Proceedings of the IS&T conference on image processing, image quality, image capture systems, Montreal, Canada, April 2001, pp 177–180

[5] C. L. Tan, R. Cao, P. Shen, *Restoration of Archival Documents Using a Wavelet Technique*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, 1399–1404, October 2002.

[6] C. L. Tan, R. Cao, P Shen, J Chee and J Chang, *Text extraction from historical handwritten documents by edge detection*, 6th International Conference on Control, Automation, Robotics and Vision, ICARCV2000, Singapore, December2000.

[7] Q. Wang, T. Xia, C. L. Tan, L. Li, «Directional Wavelet Approach to Remove Document Image Interference», ICDAR 2003: p736-740, Edinburgh, Scotland, August 2003.

[8] A. Tonazzini, E. Salerno, M. Mochi, L. Bedini, *Bleed-through removal from degraded documents using a color decorrelation method*, DAS 2004, pp 229-240, 2004.

[9] B. Gatos, I. Pratikakis, S. J. Perantonis, *An Adaptive Binarization Technique for Low Quality Historical Documents*, Document Analysis Systems VI, 6th international workshop, DAS2004, pp.102-113, Florence, ITALY, September 2004.

[10] E. Smigiel, A. belaid, H. Hamza, *Self-organizing Maps and Ancient Documents*, Document Analysis Systems VI, 6th international workshop, pp.125-134, Florence, ITALY, September 2004.

[11] Y. Leydier, F. LeBourgeois, H. Emptoz, *Serialized k-means for adaptative color image segmentation – application to document images and others*, DAS 2004, LNCS 3163, pp. 252-263, Florence, Italy, September 2004.

[12] J.A. Hartigan and M.A. Wang. A K-means clustering algorithm. Applied Statistics, 28:100{108, 1979.

[13] D. Chris and H. Xiaofeng. *K-means Clustering via Principal Component Analysis*. Proc. of Int'l Conf. Machine Learning (ICML 2004), Canada. July 2004.

Text Detection in Indoor/Outdoor Scene Images

B. Gatos, I. Pratikakis, K. Kepene and S.J. Perantonis

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

Abstract

In this paper, we propose a novel methodology for text detection in indoor/outdoor scene images. The proposed methodology is based on an efficient binarization and enhancement technique followed by a suitable connected component analysis procedure. Image binarization successfully process indoor/ outdoor scene images having shadows, non-uniform illumination, low contrast and large signal-depended noise. Connected component analysis is used to define the final binary images that mainly consist of text regions. The proposed methodology leads in increased success rates at commercial OCR engines. Experimental results based on the public database of the ICDAR2003 Robust Reading Competition prove the efficiency of the proposed approach.

1. Introduction

Indoor/outdoor scene images contain text information which is often required to be automatically recognized and processed. This paper strives toward a novel methodology that aids automatic detection, segmentation and recognition of visual text entities in complex indoor/outdoor scene images. Scene text may be any textual part of the scene images such as street signs, name plates or even text appearing on T-shirts. The research field of scene text recognition receives a growing attention due to the proliferation of digital cameras and the great variety of potential applications, as well. Such applications include robotic vision, image retrieval, intelligent navigation systems and applications to provide assistance to the visual impaired persons.

Indoor/outdoor scene images usually suffer from low resolution and low quality, perspective distortion and complex background [1]. Scene text is hard to detect, extract and recognize since it can appear with any slant, tilt, in any lighting, upon any surface and may be partially occluded. Many approaches for text detection from natural scene images have been proposed recently.

Ezaki *et al.* [2] propose four character extraction methods based on connected components. The performance of the different methods depends on character size. The most effective extraction method proves to be the sequence: Sobel edge detection, Otsu binarization, connected component extraction and rule-based connected component filtering. Yamaguchi *et al.* [3] propose a digits classification system to recognize telephone numbers written on signboards. Candidate regions of digits are extracted from an image through edge extraction, enhancement and labeling. Since the digits in the images often have skew and slant, the digits are recognized after the skew and slant correction. To correct the skew, Hough transform is used, and the slant is corrected using the method of circumscribing digits with tilted rectangles. In the work of Matsuo *et al.* [4] a method is proposed that extracts text from scene images after an identification stage of a local target area and adaptive thresholding. Yamaguchi and Maruyama [5] propose a method to extract character regions in natural scene images by hierarchical classifiers. The hierarchy consists of two types of classifiers: a histogram-based classifier and SVM. Finally, Yang *et al.* [6] have proposed a framework for automatic detection of signs from natural scenes. The framework considers critical challenges in sign extraction and can extract signs robustly under different conditions (image resolution, camera view angle, and lighting).

In this paper, we propose a novel methodology for text detection in indoor/outdoor scene images. The proposed methodology is based on an efficient binarization and enhancement technique followed by a suitable connected component analysis procedure. Image binarization successfully process indoor/ outdoor scene images having shadows, non-uniform illumination, low contrast and large signal-depended noise. Connected component analysis is used to define the final binary images that mainly consist of text regions. Experimental results show that by using the proposed method we achieve an improved recognition rate for indoor/outdoor scene images.

Our paper is structured as follows: Section 2 is dedicated to a detailed description of the proposed methodology. The experimental results are given in Section 3 while conclusions are drawn in Section 4.

2. Methodology

The proposed methodology for text detection in indoor/outdoor scene images is based on an efficient binarization and enhancement technique followed by a connected component analysis procedure. The flowchart of the proposed methodology is presented in Fig. 1. Starting from the scene image, we produce two images, gray level image O^1 and inverted gray level image O^{-1}. Then, we calculate the two corresponding binary images I^1 and I^{-1} using an adaptive binarization and image enhancement technique suitable for low resolution camera images. In the sequel, the proposed technique involves a decision function that indicates which image between binary images I^1 and I^{-1} contains text information. Finally, a procedure that detects connected components of text areas is applied. In the following sections, we will describe the main procedures of the proposed methodology.

2.1. Image Binarization and Enhancement

Binarization is the starting step of most document image analysis systems and refers to the conversion of the gray-scale or color image to a binary image. Since camera images are most of the times of low quality and low resolution, an image enhancement stage is also essential. The proposed scheme for image binarization and enhancement is mainly based on the work described in [7][8]. It does not require any parameter tuning by the user and can deal with degradations which occur due to shadows, non-uniform illumination, low contrast, large signal-dependent noise, etc. We follow several distinct steps:

Image preprocessing: For low resolution and poor quality scene images, a pre-processing stage of the grayscale source image is essential for the elimination of noisy areas, smoothing of background texture as well as contrast enhancement between background and text areas. The use of a low-pass Wiener filter [9] has proved efficient for the above goals. We should mention that we deal with both color and gray scale images. In the case of color images, we use only the luminance component.

Rough estimation of foreground regions: At this step, we obtain a rough estimation of foreground regions. Our intention is to proceed to an initial segmentation of foreground and background regions that will provide us a superset of the correct set of foreground pixels. This is refined at a later step. Sauvola's approach for adaptive thresholding [10] using $k = 0.2$, is suitable for this case. At this step, we process original image $O(x,y)$ in order to extract the binary image $S(x,y)$, where 1's correspond to the rough estimated foreground regions.

Background surface estimation: At this stage, we compute an approximate background surface $B(x,y)$ of

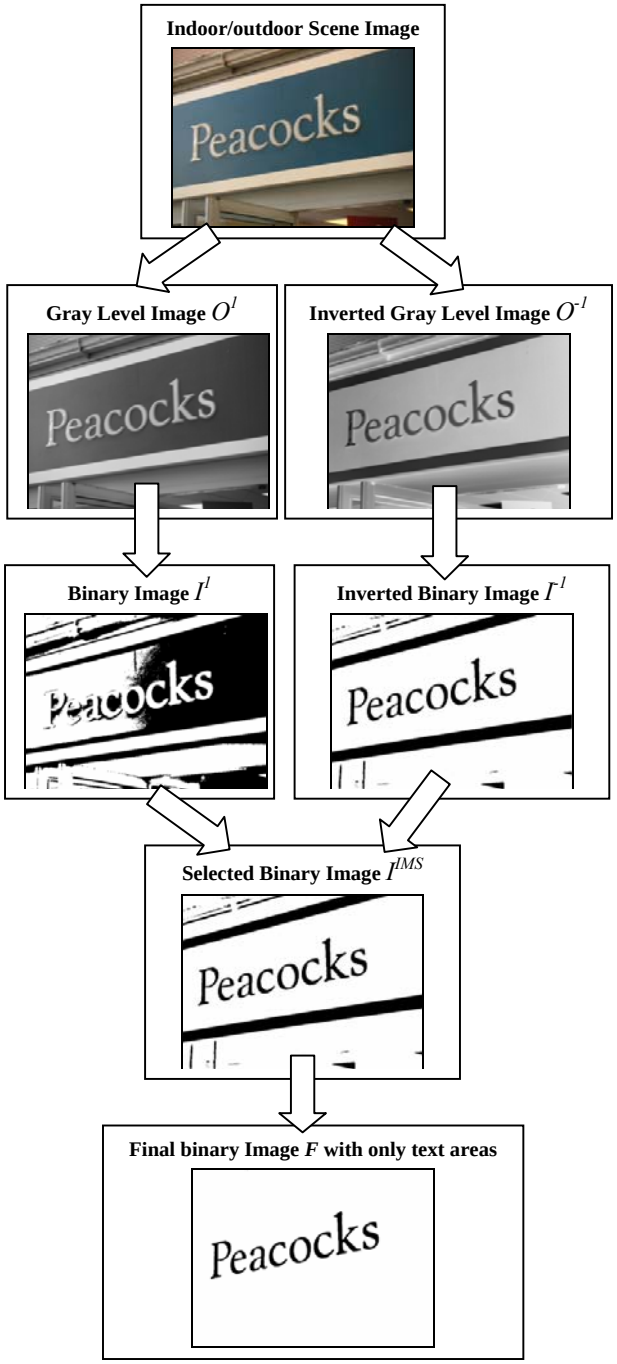


Figure 1. Flowchart of the proposed method for text detection in indoor / outdoor scene images.

the image $O(x,y)$. Background surface estimation is guided by the valuation of S(x,y) image. For pixels that correspond to 0's at image S(x,y), the corresponding value at $B(x,y)$ equals to O(x,y). For the remaining pixels, the valuation of $B(x,y)$ is computed by a neighboring pixel interpolation. At Fig. 2(b), an example of the estimated background surface of two outdoor scene images is given.

Final thresholding: In this step, we proceed to final thresholding by combining the calculated background surface *B(x,y)* with the original image *O(x,y)*. Text areas are detected if the distance of the preprocessed image *O(x,y)* with the calculated background *B(x,y)* exceeds a threshold ***d***. We suggest that the threshold ***d*** must change according to the gray-scale value of the background surface *B(x,y)* in order to preserve textual information even in very dark background areas. For this reason, we use a threshold ***d*** that has smaller values for darker regions [8].

Image up-sampling: In order to achieve a better quality binary image, we incorporate in the previous step an efficient up-sampling technique. Among available image up-sampling techniques, bi-cubic interpolation is the most common technique that provides satisfactory results [11]. It estimates the value at a pixel in the destination image by an average of 16 pixels surrounding the closest corresponding pixel in the source image.

Image post-processing: In the final step, we proceed to post-processing of the resulting binary image in order to eliminate noise, improve the quality of text regions and preserve stroke connectivity by isolated pixel removal and filling of possible breaks, gaps or holes. Our post-processing algorithm involves a successive application of shrink and swell filtering [12].

At Fig. 2(c), an example of the estimated binary images of two outdoor scene images is given. Since scene images may have dark text and light background or vice-versa, it is necessary to process both the original and the negative gray scale image in order to ensure that text information will occur in one of the two resulting binary images.

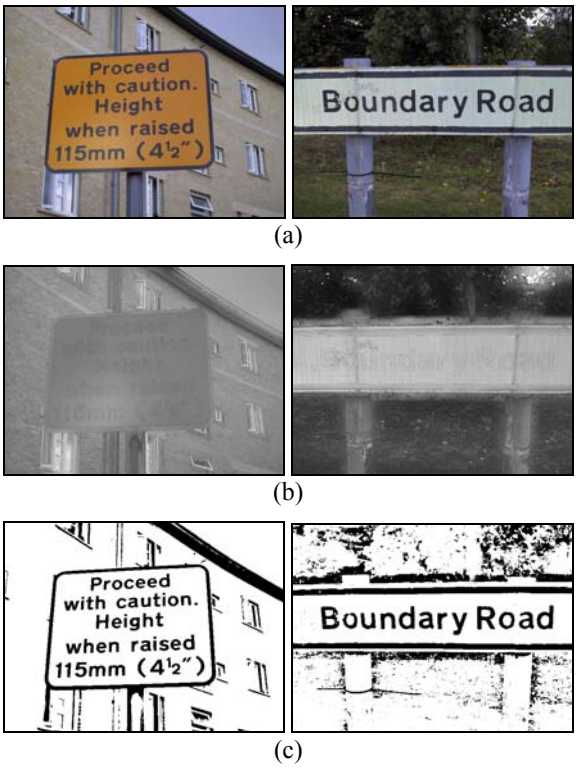


Figure 2. Image binarization and enhancement example. (a) Original outdoor scene image; (b) Estimated background surface; (c) Resulting image after image binarization and enhancement.

2.2. Text Areas Detection

After the binarization and enhancement process we get the binary images $I^1(x,y)$ and $I^{-1}(x,y)$, $x \in [1, x_{max}]$, $y \in [1, y_{max}]$. Image I^1 consists of CS^1 connected components C_i^1 with bounding boxes defined by coordinates $[C_{i,x^{TL}}^1, C_{i,y^{TL}}^1] - [C_{i,x^{BR}}^1, C_{i,y^{BR}}^1]$, $i \in [1, CS^1]$ while image I^{-1} consists of CS^{-1} connected components C_i^{-1} with bounding boxes defined by coordinates $[C_{i,x^{TL}}^{-1}, C_{i,y^{TL}}^{-1}] - [C_{i,x^{BR}}^{-1}, C_{i,y^{BR}}^{-1}]$, $i \in [1, CS^{-1}]$ (see Fig. 3). Function $CharOK(C_i^1)$ determines whether connected component C_i^1 is a character. It takes into account certain limits for the height and width of the connected component along with the appearance of neighboring connected components with almost the same height in the horizontal direction. Function $CharOK(C_i^1)$ is defined as follows:

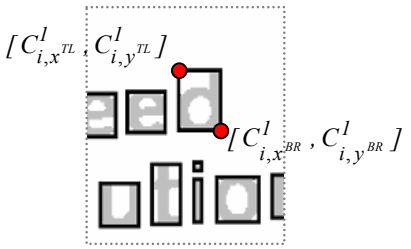


Figure 3. Connected components of the binary image.

$$CharOK(C_i^1)=\begin{cases}1, \text{ if } T1(C_i^1) \textbf{ AND} \\ \quad (\exists j:\ T2(C_i^1,C_j^1) \textbf{ AND } T3(C_i^1,C_j^1) \\ \quad\quad \textbf{AND } T4(C_i^1,C_j^1)) \\ 0, \text{ otherwise}\end{cases} \quad (1)$$

where

$$T1(C_i^1)=\begin{cases}\text{TRUE, if } (\frac{x_{max}}{MaxChars} < C_{i,x^{BR}}^1 - C_{i,x^{TL}}^1 < \frac{x_{max}}{MinChars}) \\ \quad \textbf{AND } (\frac{y_{max}}{MaxLines} < C_{i,y^{BR}}^1 - C_{i,y^{TL}}^1 < \frac{y_{max}}{MinLines}) \\ \text{FALSE, otherwise}\end{cases} \quad (2)$$

$$T2(C_i^1,C_j^1)=\begin{cases}\text{TRUE, if } (\left|C_{j,x^{TL}}^1 - C_{i,x^{BR}}^1\right| < 2(C_{i,y^{BR}}^1 - C_{i,y^{TL}}^1) \\ \quad \textbf{OR } \left|C_{j,x^{BR}}^1 - C_{i,x^{TL}}^1\right| < 2(C_{i,y^{BR}}^1 - C_{i,y^{TL}}^1) \\ \text{FALSE, otherwise}\end{cases} \quad (3)$$

$$T3(C_i^1,C_j^1)=\begin{cases}\text{TRUE, if } \left|C_{j,y^{TL}}^1 - C_{i,y^{TL}}^1\right| < C_{i,y^{BR}}^1 - C_{i,y^{TL}}^1 \\ \text{FALSE, otherwise}\end{cases} \quad (4)$$

$$T4(C_i^1,C_j^1)=\begin{cases}\text{TRUE, if } \frac{\left|(C_{i,y^{BR}}^1 - C_{i,y^{TL}}^1) - (C_{j,y^{BR}}^1 - C_{j,y^{TL}}^1)\right|}{C_{i,y^{BR}}^1 - C_{i,y^{TL}}^1} < 0.3 \\ \text{FALSE, otherwise}\end{cases} \quad (5)$$

where parameters *MaxChars* and *MinChars* correspond to the maximum and minimum number of expected characters in a text line, *MaxLines* and *MinLines* correspond to the maximum and minimum number of expected text lines and *e* is a small float. In our experiments, we used *MaxChars* = 100, *MinChars* = 5, *MaxLines* = 50, *MinLines* = 3.

$T1(C_i^1)$ is TRUE if the height and width of the connected component C_i^1 are in between certain limits, $T2(C_i^1, C_j^1)$ is TRUE if the connected components C_i^1 and C_j^1 are neighbors in horizontal direction, $T3(C_i^1, C_j^1)$ is TRUE if the connected components C_i^1 and C_j^1 belong to the same text line, and $T4(C_i^1, C_j^1)$ is TRUE if the connected components C_i^1 and C_j^1 have similar height.

Between the two images I^1 and I^{-1}, we select the one that has more connected components determined as characters. *IMS* denotes the selected image according to the formula:

$$IMS=\begin{cases}1, \text{if } \sum_{i=1}^{CS^1} CharOK(C_i^1) > \sum_{i=1}^{CS^{-1}} CharOK(C_i^{-1}) \\ -1, \text{otherwise}\end{cases} \quad (6)$$

The final binary image *F* consists of all the connected components of image I^{IMS} that are detected as characters, that is $CharOK(C_i^{IMS}) = 1$, as well as their adjacent ones. This is done, in order to include broken characters and character parts of small height. Binary image *F* is given by the following formula:

$$\begin{aligned}F = \cup C_i^{IMS}, i:\ & CharOK(C_i^{IMS}) = 1 \textbf{ OR} \\ & \exists j:\ (CharOK(C_j^{IMS}) = 1 \textbf{ AND} \\ & \sqrt{(C_{j,x^{TL}}^{IMS} - C_{i,x^{TL}}^{IMS})^2 - (C_{j,y^{TL}}^{IMS} - C_{i,y^{TL}}^{IMS})^2} + \\ & \sqrt{(C_{j,x^{BR}}^{IMS} - C_{i,x^{BR}}^{IMS})^2 - (C_{j,y^{BR}}^{IMS} - C_{i,y^{BR}}^{IMS})^2} < 4(C_{i,y^{BR}}^{IMS} - C_{i,y^{TL}}^{IMS}))\end{aligned} \quad (7)$$

In Fig. 4, a text detection example is demonstrated. In Fig. 4(b) and Fig. 4(c) the original and the inverted binary images as well as their connected components are shown, respectively. Between these two images we select image of Fig. 4(c) due to criterion in Eq. 6. The final binary image is shown in Fig. 4(d) and consists of the detected connected components as characters in the selected binary image.

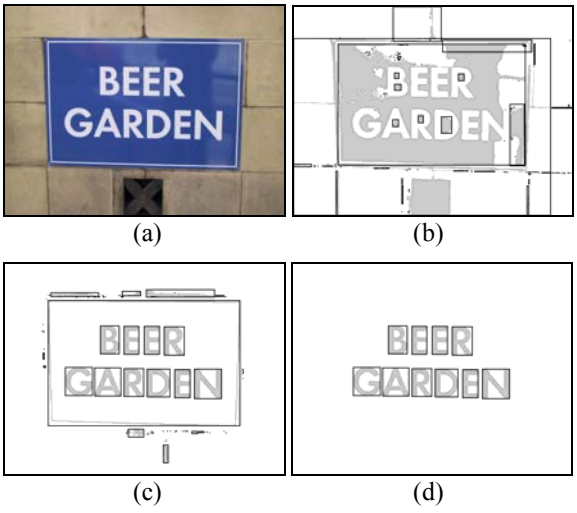


Figure 4. An example of text area detection. (a) Original scene image; (b) Resulting binary image I^1; (c) Resulting inverted binary image I^{-1}; (d) Resulting final binary image *F* with the detected text areas. In all cases, the surrounding boxes of the connected components are shown.

3. Experimental Results

The proposed algorithm was tested using the public database of the ICDAR2003 Robust Reading Competition [14]. We focused on the dataset for "Robust Reading and Text Locating" competition. A list of some representative results is presented in Table 1. In almost all the cases, the text areas are detected in the final binary images while the non-text areas are eliminated. The proposed method worked successfully even in cases with degradations, shadows, non-uniform illumination, low contrast and large signal-dependent noise. An experiment to automatically quantify the efficiency of the proposed text detection method was also performed. We compared the results obtained by the well-known OCR engine ABBYY FineReader_6 [15] with and without the incorporation of the proposed technique. To quantify the OCR results we calculated the Levenshtein distance [16] between the correct text (ground truth) and the resulting text for several scene images. As shown in the representative results at Table 2, the application of the proposed text detection technique has shown best performance with respect to the final OCR results. Total results show that by using the proposed method we achieve a more than 50% improvement of the FineReader_6 recognition rate for indoor/outdoor scene images.

4. Conclusion and Future Work

This paper strives toward a novel methodology that aids automatic detection, segmentation and recognition of visual text entities in complex indoor/outdoor scene images. The proposed methodology is based on an efficient binarization and enhancement technique followed by a suitable connected component analysis procedure. Image binarization successfully process indoor / outdoor scene images having shadows, non-uniform illumination, low contrast and large signal-depended noise. Connected component analysis is used to define the final binary images that mainly consist of text regions. Experimental results show that by using the proposed method we achieve an improved recognition rate for indoor / outdoor scene images.

In our future work, we plan to deal with the problem of text detection with dual profile (normal and invert) in the same image.

Table 1. Text detection representative results.

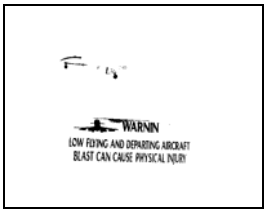
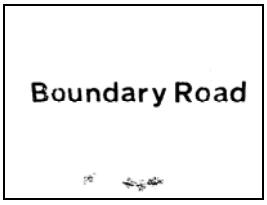
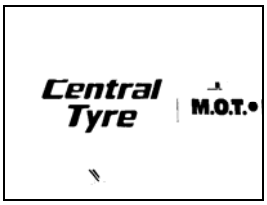
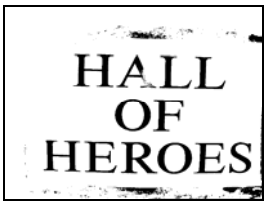
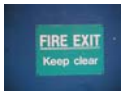
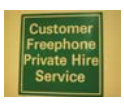
Indoor/outdoor scene image	Text detection result
	
	
	
	
	
	
	

Table 2. OCR results with and without the incorporation of the proposed method.

Indoor/outdoor scene image	Levenshtein Distance from the Ground truth		Indoor/outdoor scene image	Levenshtein Distance from the Ground truth	
	FineReade6	Proposed method+FineReade6		FineReade6	Proposed method+FineReade6
	21	0		1	1
	25	18		2	2
	5	4		32	3
	2	2		2	0
	3	3		39	18
	1	1		10	1
	0	0		10	10
	4	1		6	3
	0	0		38	16
			TOTAL	**201**	**83**

5. References

[1] David Doermann, Jian Liang, Huiping Li, "Progress in Camera-Based Document Image Analysis", In Proceedings of Seventh International Conference on Document Analysis and Recognition **(**ICDAR**'**03**)**, 2003, pp. 606-616**.**

[2] N. Ezaki, M. Bulacu and L. Schomaker, "Text detection from natural scene images:Towards a system for visually impaired persons", In Proceedings of the International Conference on Pattern Recognition (ICPR'04), 2004, pp. 683-686.

[3] T. Yamaguchi, Y. Nakano, M. Maruyama, H. Miyao, and T. Hananoi, "Digit classification on signboards for telephone number recognition", In Proc. of 7th Int. Conf. on Document Analysis and Recognition (ICDAR 2003), volume I, Edinburgh, Scotland, 3-6 August 2003, pp. 359–363.

[4] K. Matsuo, K. Ueda, and U. Michio, "Extraction of character string from scene image by binarizing local target area", Transaction of The Institute of Electrical Engineers of Japan, 122-C(2), February 2002, pp. 232–241.

[5] T. Yamaguchi and M. Maruyama, **"**Character Extraction from Natural Scene Images by Hierarchical Classifiers**",** In Proceedings of the International Conference on Pattern Recognition (ICPR'04), 2004, pp. 687-690.

[6] J. Yang, J. Gao, Y. Zang, X. Chen, and A. Waibel, "An automatic sign recognition and translation system", In Proceedings of the Workshop on Perceptive User Interfaces (PUI'01), November 2001, pp, 1-8.

[7] B. Gatos, I. Pratikakis and S.J. Perantonis, "Locating text in historical collection manuscripts", Lecture Notes on AI, SETN, 2004, pp 476-485.

[8] B. Gatos, I. Pratikakis and S.J. Perantonis, "An adaptive binarisation technique for low quality historical documents", IARP Workshop on Document Analysis Systems (DAS2004), Lecture Notes in Computer Science (3163), September 2004, pp. 102-113.

[9] A. Jain, Fundamentals of Digital Image Processing, Prentice Hall, Englewood Cliffs, NJ (1989).

[10] J. Sauvola, M. Pietikainen, Adaptive Document Image Binarization, Pattern Recognition 33, 2000, pp. 225-236.

[11] H. Hsieh, H. Andrews, "Cubic splines for image interpolation and digital filtering", IEEE Transactions on Acoustics, Speech, and Signal Processing, 26(6), 1978, pp. 508-517.

[12] R.J. Schilling, *Fundamentals of Robotics Analysis and Control*, Prentice-Hall, Englewood Cliffs, NJ (1990)

[13] Robust Reading Competition Database, http://algoval.essex.ac.uk/icdar/Datasets.html, 2005.

[14] ABBYY, www.finereader.com, 2005.

[15] V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals", Sov. Phys. Dokl., 6. 1966, pp. 707-710.

A Text Detection Technique Applied in the Framework of a Mobile Camera-Based Application

Silvio Ferreira Ph.D. Student
Vincent Garin M.S. Student
Bernard Gosselin Ph. D., Assistant Professor
Faculté Polytechnique de Mons, TCTS Labs, Belgium,
silvio.ferreira@tcts.fpms.ac.be
bernard.gosselin@fpms.ac.be

Abstract

Recent advances in mobile devices allow us to address many new challenging problems. One of them is automatic text recognition for embedded platform. This paper describes an innovative text detection system in the context of an embedded camera-based application. We propose a method to identify text regions inside an image, to correct orientation problems and to analyze document layout. Text areas are isolated with a texture segmentation approach. Due to mobile conditions, text orientation and perspective must be corrected. First a fuzzy estimation of text orientation is computed quickly. If text is too much distorted, the text perspective is corrected by using a line segmentation method in two steps. Finally the layout of the document is computed in order to deliver the reading order of the document. This language-free system has been developed with special attention to computational performances. The experimental results have proven that the method is effective and realistic.

1. Introduction

One of the most fascinating frontier projects in the field of artificial intelligence is machine understanding of text. Commercial solutions combining a scanner and a computer currently exist and have proved to be efficient. But through the recent developments in the segment of mobile devices like personal digital assistants (PDA) or smartphones a broad range of new applications are emerging. These mature hardware technologies introduce new opportunities to automatically recognize document images taken in mobile conditions with a camera-based system instead of a scanner. These new devices could be helpful for specific user groups such as blind and visually impaired people or tourists in foreign countries. Textual information is everywhere in our daily life and having access to it is essential for them to improve their autonomy and their integration. Our application aims at overcoming these barriers by offering to blind and visually impaired people a mobile access to textual information (signs, books, menus, etc.)

In our application, a blind user first takes a picture with a mobile device such as a smartphone or a PDA. Then, our system automatically detects text areas in the picture and delivers the layout of the document. Finally, text can be transformed into speech signal.

But the specific conditions of this application imply several major constraints:

- *Text image deterioration*: a document image acquired on a camera brings text detection and characters segmentation problems. Solutions need to be found to take care of the poor quality of image sensors (up to now available with this kind of devices), image stabilization and unknown lighting conditions.
- *Low computational resources*: the use of a mobile device such as a PDA or a smartphone limits the CPU and the memory resources. These constraints force our algorithmic techniques to be efficient and well optimized in order to achieve an acceptable execution time.

Three key software technologies are required for this system: text detection, optical character recognition (OCR) and text to speech synthesis (TTS). Figure 1 illustrates several examples of the realistic images database we have created in close collaboration with a group of blind users.

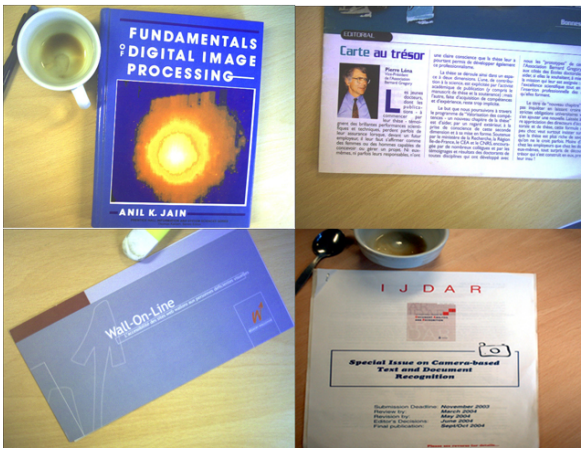


Figure 1. Samples of our images database

In this paper, we describe the current state of our text detection system, from capture of images to document layout analysis necessary in our framework for the determination of the reading order. For each step, we have either adapted traditional techniques of the text recognition field or developed new ones. This paper is organized as followed: section 2 provides an overview of the text regions detection system and the approach we follow. Considering that this part has already been detailed in a previous publication [1], this work is briefly described. This paper mainly details the methodology to deal with orientation and perspective problems in section 3 and the document layout analysis in section 4. These automatic sub-systems are especially dedicated to mobile camera-based applications. Finally section 5 concludes the paper.

2. Text detection system

Traditionally, document images are scanned with a flatbed, sheet-fed or mounted imaging device. However digital cameras have shown their potential as an alternative imaging device. But camera-based images require specific processing. The first step is detection and localization of the text regions. The idea is to locate the text elements without necessarily recognizing them, cut them out of the image, determine the reading order and eventually correct their perspective problems. Our system captures images with a resolution of 1280 * 1000 pixels and a focus fixed at a distance of 30 cm in order to be able to enclose an A4-sized document. Figure 1 illustrates the main steps of our text detection process.

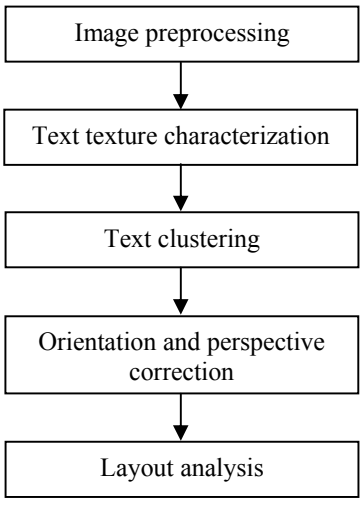


Figure 2. Overview of text detection system

Most of the previous researches about text detection focus on extracting text from video. Techniques applied to images or video key frames can broadly be classified as edge ([2], [3], [4]), color ([5], [6]), or texture based ([7], [8], [9]). Each approach has its advantages/drawbacks concerning accuracy, efficiency and computational requirements.

Our text detection technique is based on a texture segmentation approach. Text regions inside the image are considered as a textured region to isolate. Non-text contents in the image, such as blanks, pictures, graphics and other objects in the image must be considered as regions with different textures. The human vision can quickly identify text regions without having to recognize individual characters because text has textural properties that differentiate it from the rest of a scene.

Before characterization, images require pre-processing operations. Firstly original images are downsampled for the whole text detection process (due to computational restrictions). This reduction of pixels is obligatory mainly to reduce later the execution time of k-means clustering. A contrast adjustment is then operated in order to normalize global lighting conditions.

Our method for text texture characterization is based on Gabor filters which have been used earlier for a variety of texture classification and segmentation tasks [9]. The features are designed to identify text paragraphs. None of them will uniquely identify text regions. Each individual feature will still confuse text with non-text regions but a bank of filters will complement each other and allow identifying text unambiguously. We associate to the bank of filters a partially redundant feature, a local edge density measure based on Sobel filters. This feature improves the accuracy and robustness of this method while reducing false detections.

We use a reduced K-means clustering algorithm to cluster feature vectors. In order to reduce computational time, we apply the standard K-means clustering to a reduced number of pixels and a minimum distance classification is used to categorize all surrounding non-clustered pixels. Empirically, the number of clusters (value of K) was set to three, value that works well with all test images. The cluster whose centre is closest to the origin of feature vector space is labelled as background while the furthest one is labelled as text. This is because of larger answers to high spatial frequency filters in the text areas. Several results of text clustering and text region isolation are shown on figure 3.

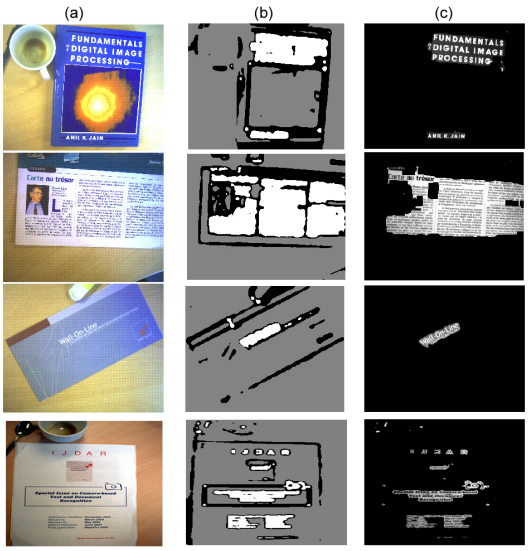


Figure 3. (a) Original images (b) Three-classes clustering (c) Text region

3. Orientation and perspective correction

3.1. Overview

At this step of the system, the image will be treated differently according to the estimation of the global orientation of the document. This fairly accurate estimation allows deciding if the perspective correction module must be applied. Indeed, due to computational efficiency, the whole perspective correction is applied only when this estimation of orientation is larger than an absolute tolerance margin. We have noticed in practise an average of 10% of non-horizontal text images taken by blind users. It confirms the need to apply perspective correction only when it is necessary to boost the average execution time. A margin of 5° is tolerated which ensures an efficient line segmentation and OCR without reorientation. This approximate orientation estimation must be computed quickly. Figure 4 schematizes the orientation and perspective correction system.

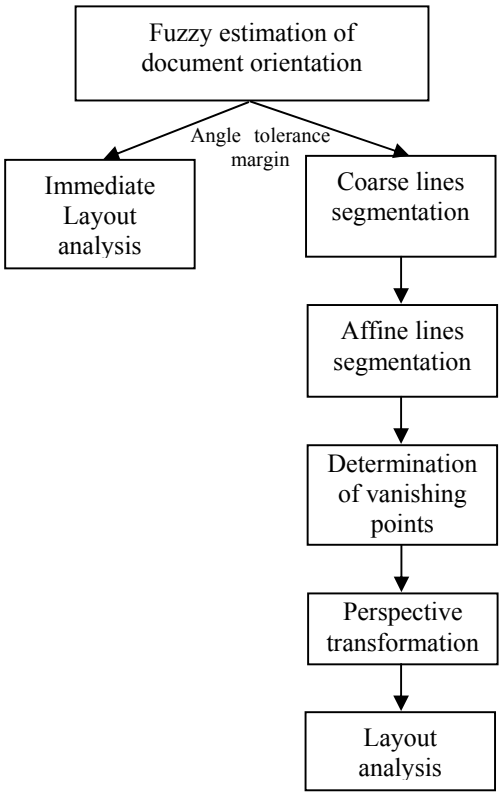


Figure 4. Scheme of perspective correction system

When the document is considered to have its perspective to be corrected, the main idea is to identify and segment text lines and then to determinate the horizontal vanishing point. The position of this point allows resolving partially unknown parameters of the perspective transform to apply. The vertical vanishing point is difficult to estimate accurately but correcting only the horizontal vanishing point already gives good results.

3.2. Approximate document orientation

Our approach is based on the theory of illusory clues [10]. When a document is captured by a camera at an unknown angle, it is of course impossible to establish a priori what is horizontal. However, given the usual layout of western-style writing, the horizontal direction is reflected in the image in the dominant direction of illusory lines. A preprocessing stage binarizes the input text areas computed during the text detection step, turning them into 'blobs' representing either single characters or (portion of) words or lines (cf. figure 5). An interesting advantage of this binarization is to analyse only pixels previously

classified as text and using an independent threshold for each text box. This method allows taking care of local gradient of luminosity in the image such as shadows, etc.

Finally the approximate global orientation is estimated with the orientation of the major axis of the blobs which are the most elongated and then the most representative. This fast estimation of orientation is performed in about 1 second with a classical PDA (CPU Intel XScale 400 MHz, 64 MB Ram). Figure 5 illustrates the fast binarization phase.

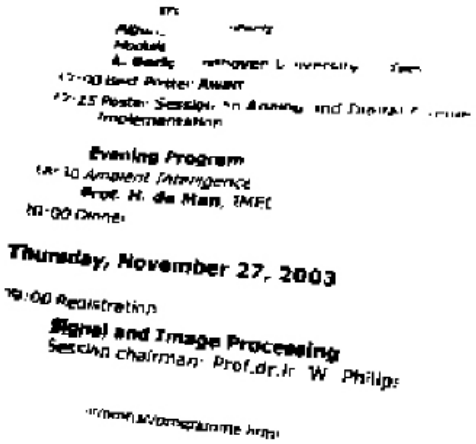


Figure 5. Binarization used for fuzzy estimation of document orientation using 'blobs' properties

3.3. Coarse and affine lines segmentation

As previously mentioned, the lines segmentation procedure is performed in two levels.

First, a reorientation is operated with the previous fuzzy estimation of text orientation. We can then compute the vertical profile of binarized text zones and operate a first coarse segmentation (cf. image (b) of figure 7) with the detection of gaps in the vertical profile.

This first lines segmentation is not accurate enough in all cases, especially when text suffers from a perspective problem. It is why a second local method is used to segment line after line. For each line previously detected with the vertical profile, a diffusion cone starts on the first blob detected. The line detection evolves incorporating the first blobs detected into the cone. When a new blob is added into the line, the properties of the diffusion cone (orientation and aperture) are adapted with the position, the size and the orientation of the last incorporated blob. This local segmentation method enables taking into account text lines with perspective problems. The algorithm is illustrated in figure 6 and in image (c) of figure 7. Figure 6 shows the diffusion of the cone across one line.

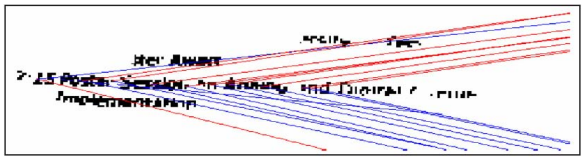
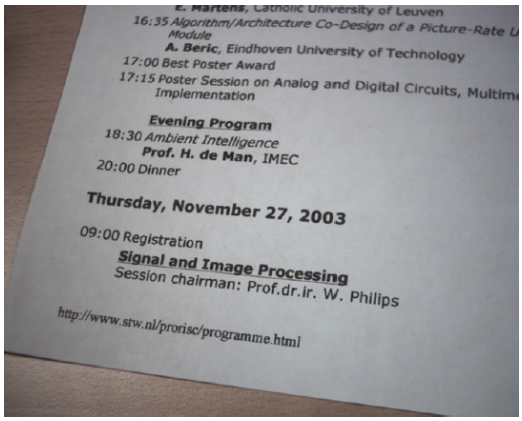


Figure 6. Affine lines segmentation

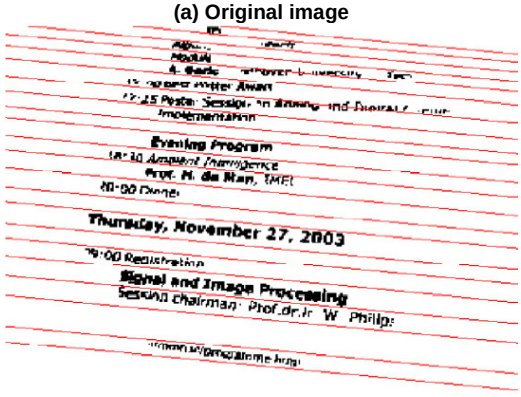
The orientation and perspective correction sub-system performs well in about 80% of cases of images with perspective problems.

3.4. Vanishing point determination and perspective transform

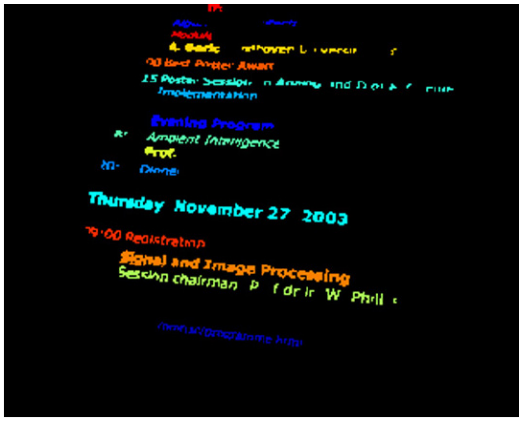
In non-horizontal images the text lines converge in a point called the horizontal vanishing point. We use a fast approximation method to estimate its position. We can then resolve partially the perspective transform. The mathematical development is detailed in [11]. The result is illustrated in image (d) of figure 7.



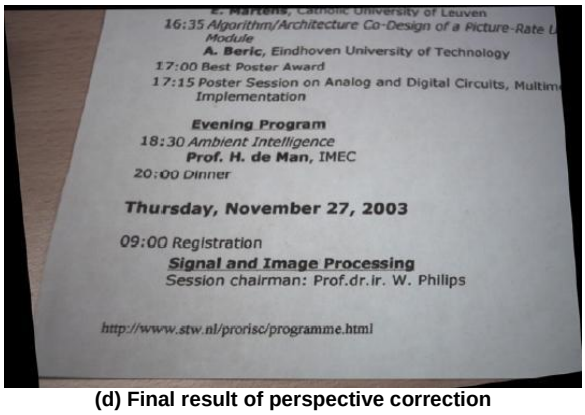
(a) Original image



(b) First coarse lines segmentation



(c) Second affine lines segmentation



(d) Final result of perspective correction

Figure 7. Main steps of perspective correction

Even without the determination of the vertical vanishing point, this method performs satisfying results in about 80% of non-horizontal document images but the entire algorithm requires an execution time of about 5 seconds with a classical PDA.

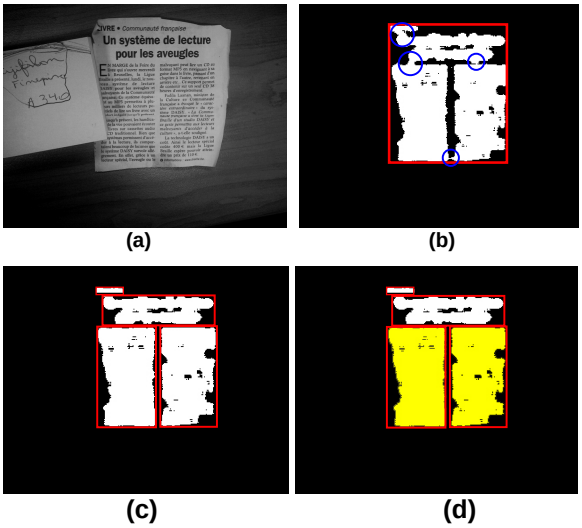
4. Document layout analysis

A document image contains important information in the geometric arrangement of the text zones on the page – the page layout. The layout of a document is the result of the application of complex, interactive rules about where to place text on the page. But almost all layouts tend to be composed of a number of recurring primitives, text lines, paragraphs and columns. We call the extraction of these primitives physical document layout analysis. The extraction of higher-level properties of a document like titles or authors is referred as logical document layout analysis [12].

In our framework we try to extract the physical layout analysis of the unknown document. The layout analysis is performed to organise text boxes for a logical reading order. Layout analysis provides in this manner a reading position to every text box. They are later processed independently by the the OCR system. We make assumptions of occidental writing systems. Text is read from top to bottom and from left to right. Another assumption consists in the major presence of document images composed of traditional class of layouts like Manhattan textual layouts which are fully described in [13]. Briefly the document page must be composed mainly of blocks of text lines and symbols and lines must share a common orientation.

Our document analysis sub-system is designed with special care to computational performances. It is why the algorithm uses previously computed information (results of text detection module). Text detection areas (binary images) will be transformed in text boxes and labelled in order.

Firstly an iterative procedure of columns and paragraphs separation is applied based on the morphological profiles of every binary text box. The separation can be achieved precisely with the detection of gaps in the vertical and the horizontal text class profiles of each textbox. These gaps correspond to undesirable 'bridges' which link columns and paragraphs together (cf. blue circles in image (b) of figure 8). This type of analysis allows fast computational performances due to the use of one-dimensional signals and binary sub-images.



(a) (b)

(c) (d)

Figure 8. (a) Original image (b) Illustration of text regions to separate (c) Results after iterative columns & paragraphs separation (d) Illustration of detection of columns

Now that paragraphs and columns are separated, we can detect the columns (cf. image (d) of figure 8). The method to identify two or more columns takes into account the ratios of vertical overlay between textboxes (to detect horizontal alignment) and their relative distance. Finally the reading order is decided

between the boxes from top to bottom taking into account detected columns. This method of layout analysis is performed on a PDA in about one second. One final result is shown on figure 9. The layout analysis sub-system has separated and given a reading order to four text boxes.

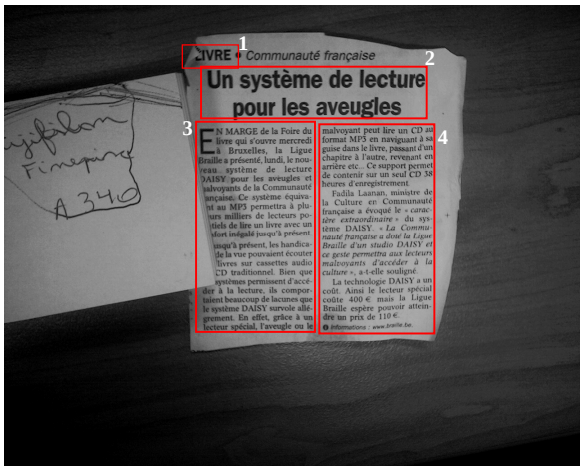


Figure 9. Layout analysis result

In our images database, the layout analysis sub-system has an estimated efficiency of about 90%.

5. Conclusion

This paper has described a text detection and document analysis system in the context of a camera-based text recognition application. The method has been designed in the context of providing mobile access to textual information for blind and visually impaired people.

The initial work has focused on an adapted technique for the separation of a document image into regions of text.

The text detection method based on texture segmentation has been tested with a variety of printed documents from different sources. Performances are acceptable although misclassifications occur occasionally. These misclassifications errors are often detected later during the optical character recognition step and then rejected.

The main contribution of this work consists in a new efficient approach to perform page deskewing and document analysis. The system performs orientation and perspective correction only when required after a first fast fuzzy estimation of text orientation. Results of this algorithm are promising but a reduction of its computational complexity need to be realised when the whole process of page deskewing is operated. Our document analysis system aims at extracting the physical layout of the document and deciding the logical reading order. Its results and computational performances are satisfactory.

The algorithms have been designed using a realistic images database (pictures were taken by blind users). Due to this methodology, further improvements would consist in the adaptation of this system to reference images databases (ICDAR images databases for example) in order to compare it performances to other techniques and obtain more quantitative results.

6. Acknowledgements

This project is called Sypole and is funded by Ministère de la Région wallonne in Belgium.

7. References

[1] S. Ferreira, C. Mancas-Thillou, B. Gosselin,"From Picture to speech: an innovative OCR application for embedded environment", Proc. of the 14th ProRISC workshop on Circuits, Systems and Signal Processing (ProRISC), 2003

[2] P. Clark and M. Mirmehdi, “Recognizing text in real scenes”, International Journal on Document Analysis and Recognition, Springer Berlin Heidelberg, August 2002, vol.4, pp.243-257

[3] J. Ohya, A. Shio and S. Akamatsu, “Recognizing characters in scene images”, IEEE Transactions on Pattern Analysis and Machine Intelligence, February 1994, vol. 16 n°2 pp.214-220

[4] M. Pietikäinen and O. Okun, “Text extraction from grey scale page images by simple edge detectors”, Proc. of the 12th Scandinavian Conference on Image Analysis, June 2001, pp.628-635

[5] W.-Y. Chen and S.-Y. Chen, “Adaptative page segmentation for color technical journals’cover images”, Image and Vision Computing, Elsevier Science, 1998, vol.16, pp.855-877

[6] Y. Zhong, K. Karu and A.K. Jain, “Locating text in complex color images”, Pattern Recognition, 1995, vol.28, n° 10, pp.1523-1535

[7] V. Wu, R. Manmatha and E.M. Riseman, “Textfinder: an automatic system to detect and recognize text in images”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Nov. 1999, vol.21, n° 11, pp.1224-1229

[8] H. Li, D. Doermann and O. Kia, “Automatic text detection and tracking in digital video”, IEEE Transactions on Image Processing, January 2000, vol.9, n°1, pp.147-156

[9] A.K. Jain and S. Bhattacharjee, “Text segmentation using Gabor filters for automatic document processing”, Machine Vision and Applications, Springer-Verlag, 1992, vol.5, pp.169-184

[10] Maurizio Pilu, “Extraction of illusory linear clues in perspectively skewed documents”, IEEE Computer Vision and Pattern Recognition Conference, Kauai, December 2001

[11] G. Fangi, G. Gagliardini, E.S. Malinverni, “Photointerpretation and small scale stereoplotting using digitally rectified photographs by geometrical constraints”, in

Int. Archives of Photogrammetry and Remote Sensing, vol. XXXIV, part. 5/C7, Germania, 2001, pp. 160-167
[12]T.M. Breuel, "A Review of Branch-and-Bound Algorithms for Geometric and Statistical Layout Analysis", Huitième Colloque International Francophone sur l'Ecrit et le Document CIFED 2004, La Rochelle, 21-25 juin 2004
[13] D.J. Ittner and H.S. Baird, "Language-Free Layout Analysis", Proc. IAPR 2nd International Conf. on Document Analysis & Recognition, Tsukuba Science City, Japan, October 1993, pp.336-340

Isolated Character Recognition by Searching Features in Scene Images

Kazuya NEGISHI[*], Masakazu IWAMURA[†], Shinichiro OMACHI[*], and Hirotomo ASO[*]
[*]Tohoku University, 6–6–05 Aoba, Aramaki, Aoba-ku, Sendai-shi, 980–8579 Japan
[†]Osaka Prefecture University, 1–1 Gakuencho, Sakai-shi, Osaka, 599–8531 Japan

Abstract

Conventional segmentation technique cannot extract an isolated character and a touching character. In this paper, to utilize information of such characters, we propose a novel character recognition method based on extracting feature points and voting. The voting algorithm of the proposed method is similar to the generalized Hough transform. This method enables us to extract and recognize such troublesome characters in relatively shorter computational time. The effectiveness of the proposed method is confirmed by experiments.

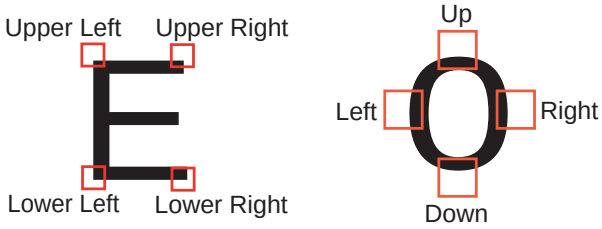
1 Introduction

Accuracy of character recognition for segmented character image is enough for practical use. However, a conventional segmentation method can be applied to only *well-defined* problems such that the characters constitute a string and a character is completely separated from other characters and so on. Therefore, an *isolated character* which does not constitute a character string, and a touching character which is connected to other characters are hard to extract. In this paper, we propose a novel character recognition method which executes segmentation and recognition of a character simultaneously. The proposed method extracts features and then votes. The voting algorithm is similar to the generalized Hough transform [1, 2, 3]. The proposed method allows segmentation of an isolated character, a touching character and so on in relatively shorter computation time.

2 Preparation

2.1 Input image and reference image

"Input image" is a relatively large image such as a photograph including isolated characters. "Reference image" is a relatively small image of one of the 52 alphabet letters and 10 numerals.



(a) Corners. (b) Curves.

Figure 1. Features.

2.2 Histogram of edge directions

The proposed method searches for eight kinds of features described in Sec. 2.3 from the square area of 5×5 pixels in the input image and the reference images. The square area is called *searching area*. Each feature is characterized by a histogram of edge direction in a searching area.

A histogram is constructed as follows. In advance, the edge direction and the edge intensity of each pixel is calculated with the Sobel filter. The edge direction is quantized into n bins whose width is $\theta = \frac{2\pi}{n}$ radian. Namely, n intervals of bins are defined as

$$\left[-\frac{\pi}{n} + \frac{2\pi}{n}t, \frac{\pi}{n} + \frac{2\pi}{n}t\right), \quad 0 \leq t \leq n-1. \qquad (1)$$

Then, a histogram whose bin width is $\theta = \frac{2\pi}{n}$ radian is constructed in a searching area. To eliminate the effect of noise, a pixel whose edge intensity is less than a threshold is regarded as a *no edge pixel*. The histogram does not contain such *no edge pixels*. Finally the histogram is normalized so that the sum of all the bin values to be 1. This normalized histogram is used as a feature vector of n dimensions for the searching area.

2.3 Features

A corner, a curve, a branch, an intersection and a bending point have been considered as effective features in character recognition [4]. In the proposed method, four corners and four curves are used as features because they are easy to extract by search and seem to be effective. Four corners are "upper left", "upper right", "lower left" and "lower right" (See Fig. 1(a)). Four curves are "up", "down", "left" and "right" (See Fig. 1(b)).

To extract features robustly, larger θ is desired. However, to extract complex features, smaller θ is required. Therefore, θ depends on features: $\theta = \pi/2$ (i.e., $n = 4$) was used for corners, and $\theta = \pi/8$ (i.e., $n = 16$) for curves. A loose feature of four bins can extract corners of both regular font such as Arial and oblique font such as Franklin.

2.4 Similarity

To evaluate how much a searching area is similar to each feature, a similarity measure proposed by Swain et al. [5] is used. Let H be the histogram of the searching area, M be the histogram of a corner feature or a curve feature. Then, the similarity between H and M are

$$S_{HM} = \sum_{t=1}^{Q} \min(H_t, M_t), \tag{2}$$

where H_t and M_t are the t-th bin value of H and M respectively, and Q is the number of bins of the feature (i.e., 4 or 16). Since both H and M are normalized histograms, $0 \leq S_{HM} \leq 1$. If the similarity of the searching area is larger than a threshold T_1, the searching area is regarded as a feature point. The range of T_1 is $0 \leq T_1 \leq 1$.

3 Segmentation and recognition of isolated characters

The proposed method is similar to the generalized Hough transform. Therefore we explain the generalized Hough transform at first, then we explain the proposed method.

3.1 The generalized Hough transform [3]

The generalized Hough transform is used to detect a figure that cannot be analytically easily expressed although it has the particular silhouette. We explain the generalized Hough transform without considering rotation and expansion or reduction below.

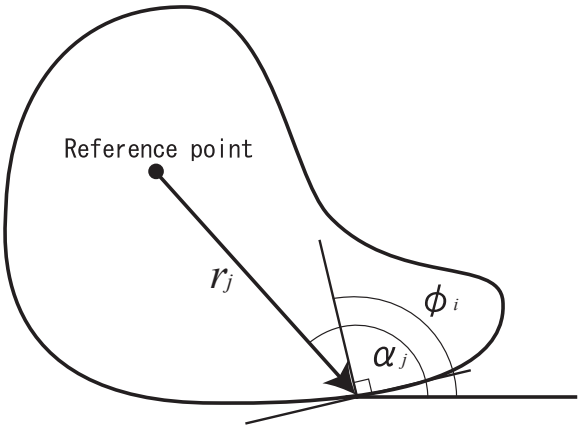


Figure 2. The generalized Hough transform.

Edge direction	Locations of feature points from the reference point
ϕ_1	(r_{11}, α_{11}), (r_{12}, α_{12}) ,...
ϕ_2	(r_{21}, α_{21}), (r_{22}, α_{22}) ,...
⋮	⋮

Table 1. R table of the generalized Hough transform.

3.1.1 Description of a figure

In the generalized Hough transform, a target figure is registered by feature points. This process equals to construct an R table as shown in Table 1. At first, a reference point in the target figure is set as shown in Fig. 2 and each edge point of the target figure (we call this feature point) is also set. A vector from the reference point to the j-th feature point (which corresponds to a combination of r_j and α_j in Fig. 2) is calculated, and the edge direction of the feature point (ϕ_i in Fig. 2) and the vector is registered in the R table. The edge direction is used as an index.

3.1.2 Detection of a figure

The following procedure detects the target figure. First, all the edge directions are examined. For convenience of explanation, the edge direction at coordinate (X, Y) is assumed to be ϕ. Secondly, candidate locations of the reference point are calculated from the edge direction ϕ by referring the R table. The candidate of the reference point calculated is called *voting point*. Then the voting value at the voting point is increased by one degree. Here, if $\phi = \phi_2$, the voting points will be $(X - r_{21}cos\alpha_{21}, Y - r_{21}sin\alpha_{21})$ and $(X - r_{22}cos\alpha_{22}, Y - r_{22}sin\alpha_{22})$. After all the votes

are finished, figures whose reference point have large voting values are detected.

3.2 The proposed method

The difference between the proposed method and the generalized Hough transform is summarized as follows.

(1) Although a vote in the generalized Hough transform is based on edge directions, a vote in the proposed method is based on types of the detected features.

(2) Although a vote in the generalized Hough transform increases the voting value only at the reference point, a vote in the proposed method increases not only the reference point but also the points around the reference point. This enables to detect figures robustly even if fonts are different.

(3) Although the maximum number of the voting value is undecided in the generalized Hough transform, it is described by $F_s^{(k)}$ defined in Sec. 3.2.1 in the proposed method.

3.2.1 Description of a figure: detection of the feature points from the reference image

First of all, all the corner and curve feature points are detected from the reference images. The information on the positions and types of the feature points are used for description of a figure. However, since the feature points tend to be detected too much, these points are narrowed down to a representative point as illustrated in Fig. 3. The representative point is decided to have the highest similarity among the points around it within q pixel distance. $q = 5$ is used in this paper.

Then the number of the points represented by a representative point is used as the *voting weight*. Let $W_{ij}^{(k)}$ be the voting weight of the j-th representative point of the i-th feature of character k. The position of the representative point is determined by *voting vectors* as illustrated in Fig. 4. In the generalized Hough transform, the vector corresponds to a combination of r_j and α_j. It is defined as follows. Let the origin be the upper left corner of the reference image of a character. Let $N_i^{(k)}$ and $R_i^{(k)}$ be the number of occurrences of the i-th feature in the reference image and that of their representative points respectively. Let $(x_{ij}^{(k)}, y_{ij}^{(k)})$ be the coordinate of the j-th representative point of the i-th feature in the character k. The voting vector is defined as

$$\boldsymbol{f}_{ij}^{(k)} = (x_{ij}^{(k)}, y_{ij}^{(k)}), \quad 1 \leq i \leq 8,\ 1 \leq j \leq R_i^{(k)}. \qquad (3)$$

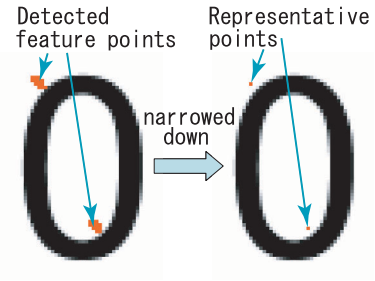


Figure 3. Detected feature points of the upper left corner feature from the reference image (left) and their representative points (right).

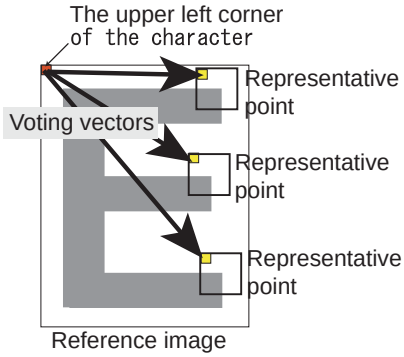


Figure 4. Feature point extraction from a reference image. Three representative points represent three clumps of feature points extracted as the upper right corner. A voting vector is a vector from the upper left corner of the character image to a representative point.

The total number of occurrence of eight features in the reference image of character k is defined by

$$F_S^{(k)} = \sum_{i=1}^{8} N_i^{(k)}. \qquad (4)$$

This value is used in Sec. 3.2.2.

3.2.2 Detection of a figure: segmentation

At first, all the feature points are detected from the input image, and these feature points are also narrowed down into representative points as in Sec. 3.2.2. Then, figures are detected by voting. The categories of the characters and their locations are also determined by voting at the *voting point*

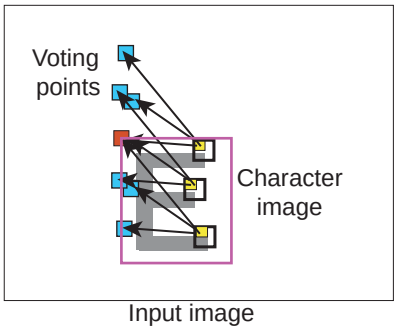


Figure 5. Feature point extraction and segmentation from the input image. There are three voting points for each representative point in this example.

$\boldsymbol{P}_{ijl}^{(k)}$. $\boldsymbol{P}_{ijl}^{(k)}$ is given as

$$\boldsymbol{P}_{ijl}^{(k)} = \boldsymbol{x}_{il} - \boldsymbol{f}_{ij}^{(k)}, \quad 1 \leq j \leq R_i^{(k)}, \text{ for all } k, \tag{5}$$

where $\boldsymbol{x}_{il}$ is the l-th representative point of the i-th feature in the input image. In short, the voting point is a candidate of the upper left corner of a character image as illustrated in Fig. 5.

To allow deformations of a character image, the neighbors of the voting point are also voted. The voting rule is as follows: Let $v_{ij}^{(k)}(x,y)$ be the *voting table*, which stands for the voting value at (x,y) of the input image. In advance, it is initialized as $v_{ij}^{(k)}(x,y) = 0$ for all (x,y). The neighbors within r pixels distance from the voting point are voted as

$$v_{ij}^{(k)}(x,y) = W_{ij}^{(k)}, \quad \text{for } (x,y) \text{ s.t. } \left|\left|(x,y) - \boldsymbol{P}_{ijl}^{(k)}\right|\right| \leq r. \tag{6}$$

In this paper, $r = 10$ is used. Note that even if the same point is voted more than once, the voting value is $W_{ij}^{(k)}$ as long as the same voting vector is used.

After voting, let $V^{(k)}(x,y)$ be the sum of the voting tables $v_{ij}^{(k)}(x,y)$. Namely,

$$V^{(k)}(x,y) = \sum_i \sum_j v_{ij}^{(k)}(x,y). \tag{7}$$

$V^{(k)}(x,y)$ stands for the possibility that the upper left corner of the character image of category k exists at (x,y). Let T_2 be the threshold of character segmentation where $0 \leq T_2 \leq 1$. If

$$V^{(k)}(x,y) \geq F_S^{(k)} T_2 \tag{8}$$

is satisfied, (x,y) will be determined to be the upper left corner of an image of character k. Note that, $F_S^{(k)} = \max V^{(k)}(x,y)$.

3.2.3 Comparison with the traditional template matching method

Compared with the simple traditional template matching method, the proposed method can detect a deformed character more robustly. Therefore the computational cost of the proposed method is a little larger than that of a simple template matching method.

The details of the computational cost are described below. When the size of the reference image is $M \times M$, and that of the input image is $N \times N$, the computational cost of the simple template matching method is $O(M^2N^2)$. In the proposed method, for the feature extraction, it is $O(8 \times 5^2N^2)$ because it searches 8 kinds feature of 5×5 with template matching method. $O(8 \times 5^2N^2)$ is less than $O(M^2N^2)$ for $M > 10\sqrt{2}$, i.e. $M \geq 15$. For the voting algorithm, let $X_i^{(k)}$ be the number of representative point of the i-th feature of character k in the input image. Then, the computation cost of the voting algorithm of the proposed method is $O(r^2 R_i^{(k)} X_i^{(k)})$. Since the voting algorithm is added to template matching method, the computation cost of the proposed method is larger than the simple template matching method.

4 Experiments

To confirm the effectiveness of the proposed method, three experiments were carried out against (1)different fonts, (2)touching characters, (3)isolated characters. As the preliminary to all the experiments, both the reference and input images were converted into gray scale and smoothing filter was applied to them.

The proposed method ignores *no edge pixels*, and does not use the information of them for segmentation and recognition. However, combination of many *no edge pixels* and a few edge pixels in a searching area can cause miss detection of the feature points. Therefore, a searching area in which over 70% pixels are *no edge pixels* is ignored.

Recognition results are classified into three categories for evaluation: “Correct”, “Match”, and “Miss”. If the extracted character images consist of only the same category as the reference one, it is classified into “Correct”. If the extracted images include the same category as the reference one and more than one other characters, it is classified into “Match”. If all the character images of the same category as the reference one are not extracted, it is classified into “Miss”.

4.1 Against different fonts

Input images of several fonts were prepared. When the fonts of the reference image and the input image were the

Font	Correct	Match	Miss
Arial Black	18 (29%)	43 (69%)	1 (2%)
Franklin	15 (24%)	44 (71%)	3 (5%)
Maru Gothic	22 (35%)	23 (37%)	17 (27%)

Table 2. The number of occurrences and its ratio fot different fonts. The sum of ratios is not always 100% because the ratios are rounded.

same, recognition rate was 100%. Therefore, experimental results against different fonts are shown here. Arial was used for the reference images, and Arial Black, Franklin and Maru Gothic were used for the input image: Arial Black is bold, Franklin is oblique, Maru Gothic has rounded corners. Two thresholds $T_1 = 0.75$ and $T_2 = 0.85$ were used. Recognition results are summarized in Table. 2.

The table shows that almost 30% were classified into "Correct". Also, more than 95% of Arial Black and Franklin, and more than 70% of Maru Gothic were classified into either "Correct" or "Match". Though images of "Match" contain images of other categories, they can be recognized by conventional charactrer recognition methods for extracted character images. Therefore, sum of "Correct" and "Match" shows the effectiveness of the proposed method.

Then, we consider the causes of "Miss". 27% of Maru Gothic were classified into "Miss". One of the reason is thresholds. When $T_2 = 0.75$ were employed instead of $T_2 = 0.85$, all categories were correctly extracted (in detail, 6 categories were "Correct" and the rest were "Match"). Therefore, this problem can be solved by determining the proper threshold for the image.

4.2 Against touching characters

An image including touching characters [6] was used as the input image. "S" of Arial was used as the reference image. $T_1 = 0.75$ and $T_2 = 0.8$ were used. The size of the reference image was changed to fit the largest "S" in the input image. The recognition result in Fig. 6 shows that "S" was correctly extracted ("Correct"). In addition, "e" was also extracted ("Match").

In this experiment, since the size of the reference image fit the largest "S" in the input image, smaller "S"s were not extracted. Therefore, an invariant method for the size of the input image are required.

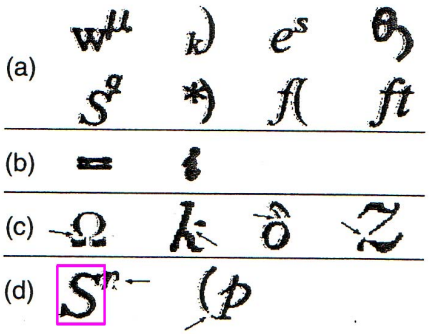


Figure 6. Recognition of touching characters: the reference image was "S".

4.3 Against isolated characters

To confirm the effectiveness of the proposed method against isolated characters in scene images, experiments were carried out. For all images, $T_1 = 0.75$ and $T_2 = 0.8$ were used. The size of an input image was reduced so that the size of the character in the input image is equal to that of the reference one.

The proposed method utilized the Sobel filter for extracting the edge direction in the input and the reference images. However, there is a problem that the extracted edge direction is different in the case of the black lettering on a white background and the white lettering on a black one, since the proposed method determines the edge direction that based on the black lettering on a white background. Therefore, we used reversed color to avoid this problem if the image includes white lettering on a black background.

The experimental results are shown in Figs. 7 to 19. Arial font wes used for the reference image. For each figure, the character used as the reference image is shown in the caption. In summary, the following two results were obtained.

(1) Over 90 percents of the characters in the input images were segmented by the proposed method. This result confirms the effectiveness of the proposed method for isolated characters in scene images. Though images of "Match" contain images of other categories, they can be recognized by conventional charactrer recognition methods for extracted character images. Therefore, hereafter we consider the causes of "Miss" such as "3" in the number plate of a car in Fig. 18(a) and "C" in the thermometer in Fig. 15(b). Two main causes look to be (i) threshold T_2, and (ii) essential difference between character shapes of an input image and a reference one. For the cause (i), change in threshold T_2 helps recovering from the failure of extraction. For example, the characters in Fig. 19 were not extracted when

$T_2 = 0.8$. However, when T_2 was changed into 0.7, both images were extracted completely. This indicates an automatic selection method of the threshold is required. For the cause (ii), it is difficult for the proposed method to extract figures. For example, the shape of "1" in the number plate in Fig. 18(b) is similar to "I" of Arial font rather than "1". When "I" was used as the reference image, "1" was extracted successfully. Here is another example. Fig. 21 includes "3" of the reference image and that in the number plate in Fig. 18(b). The two figures show that there is great difference in extracted feature points.

(2) The average computation time of the proposed method was 5.2 seconds, while that of the simple template matching method was 0.73 seconds. The machine used is 2.4GHz Pentium 4 with 1024MB memory. The proposed method takes more time than the template matching method because the neighbors within 10 pixels distance from the voting point are voted to allow deformations of a character image. The proposed method can detect a deformed character by this procedure, while the simple template matching method cannot.

5 Conclusion

In this paper, we proposed a novel character recognition method which executes segmentation and recognition simultaneously. The proposed method is based on histogram of edge directions. Experimental results showed the effectiveness of the proposed method against (1)different fonts, (2)touching characters, and (3)isolated characters. Developing automatic selection method of thresholds and invariant method for the size of the input image are future works.

Acknowledgment This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 16500096, and Young Scientists (B), 17700205, 2005.

References

[1] D. Ballard, Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, vol.13, no.2, pp111–122, 1981.

[2] D.H. Ballard, C.M.Brown, Computer vision. *Japan computer association*, 1987. Written in Japanese.

[3] T. Matsuyama, Y. Kuno, A. Imiya, Computer vision :technical criticism and future perspective, *Shingijutsu communications*, 1998. Written in Japanese.

[4] K. Mori. *Pattern Recognition*. IEICE, Tokyo, 1988. Written in Japanese.

[5] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[6] S. Uchida, A. Nomura, and M. Suzuki. Quantitative analysis of mathematical documents. *IEICE Technical Report*, 2004. Written in Japanese.

[7] K. Negishi, M. Iwamura, S. Omachi, and H. Aso. Isolated Character Recognition by Search of the Partial Regions. *Forum on Information Technology*, pp37–38, 2004. Written in Japanese.

Figure 7. A recycle mark: "1".

(a) "6".

(b) "P".

Figure 8. Signboards of a bus stop and parking.

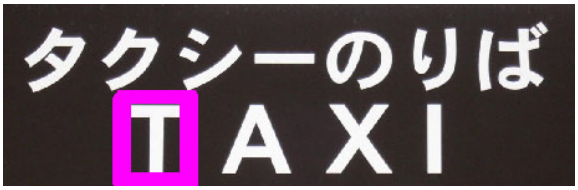


Figure 9. Signboards of a taxi stand: "T".

Figure 10. A tachometer of a car: “4”.

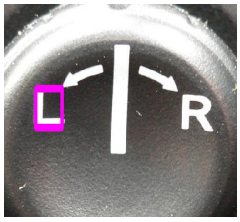
Figure 11. A speed meter of a car: “0”.

(a) “D”.

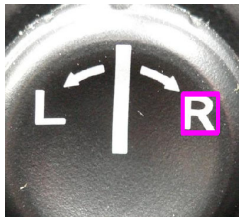
(b) “N”.

(c) “P”.

Figure 12. A shift indicator of a car.



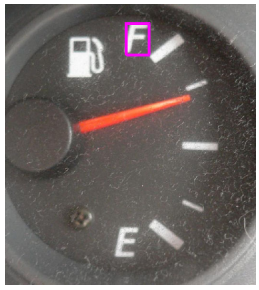
(a) “L”.



(b) “R”.

Figure 13. A mirror adjuster of a car.

(a) “E”.



(b) “F”.

Figure 14. A fuel gauge of a car.

(a) “H”.

(b) “C”.

Figure 15. A thermometer of a car.

Figure 16. A traffic sign: “4”.

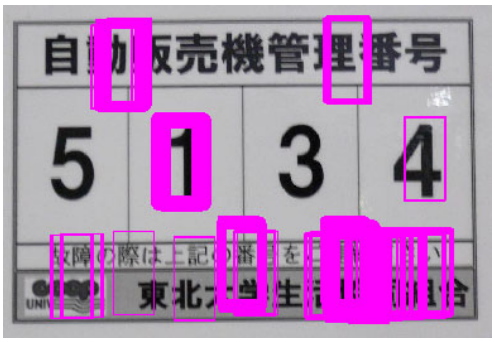


Figure 17. A management number of a vending machine: “1”.

(a) “3”.

(b) “I ”. Thereshold T_2 = 0.95.

Figure 18. A number plate of a car.

(a) “2”.

(b) “5”.

Figure 19. Signboards of an address and a number. Thereshold T_2 = 0.7.

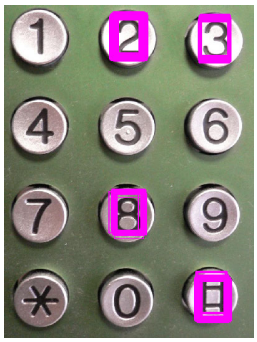
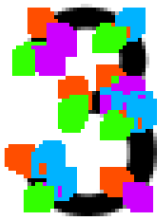
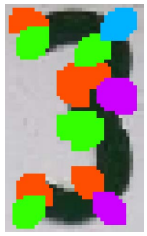


Figure 20. Push buttons of a telephone: “2”.



(a) “3” of the Arial font.



(b) “3” in the number plate in Fig. 18.

Figure 21. Extracted feature points.

Perspective Correction Methods for Camera-Based Document Analysis

L. Jagannathan and C. V. Jawahar
Center for Visual Information Technology
International Institute of Information Technology
Gachibowli, Hyderabad, India

Abstract

In this paper, we describe a spectrum of algorithms for rectification of document images for camera-based analysis and recognition. Clues like document boundaries, page layout information, organisation of text and graphics components, apriori knowledge of the script or selected symbols etc. are effectively used for removing the perspective effect and computing the frontal view needed for a typical document image analysis algorithm. Appropriate results from projective geometry of planar surfaces are exploited in these situations.

1. Introduction

Document images are omnipresent. Textual content in the form of books, newspapers and articles have been traditionally digitized using at-bed scanners and read with the help of OCRs. These reading systems may not be appropriate in situations where mobile, portable or non-contact reading systems are needed. Cameras, which can scan text without contact even on non-planar surfaces, is an emerging alternative to the conventional scanners. In general, cameras are small in size, lightweight and easy to use. Although many of the present day scanners outperform the popular cameras in resolution, the cameras remain attractive alternative especially in situations described above and for non-critical applications. Advances in sensor technology is expected to take the camera-based systems more favorable.

Camera-based imaging process introduce many new challenges to the document understanding process [3]. Images acquired through cameras suffer from projective distortion, uneven lighting and lens distortion. Algorithms for understanding the images with these distortion would become much more complex due to the additional parameters to be taken care while designing them. Instead of this, we could use methods to remove these effects/distortions for intelligent processing of document images. A license plate reading system [7] analysing the traffic videos capture an image similar to Figure 1(a), while Figure 1(b) is better processed by a machine. The transformation between the two images is achieved by removing the perspective distortion. The perspective distortion of a planar surface can be understood as a projective transformation of a planar surface. A projective transformation is a generalised linear transformation (Homography) defined in a homogeneous coordinate system. Different clues in the document image itself could be used for the purpose of rectification. Boundaries of documents, page layout and textual structure provide important clues to rectify the perspective distortion. Where gross structure is absent in the document, word level or character level information could be used in recovering the fronto-parallel view from an arbitrary view. In this paper we explore various rectification techniques that are useful for projective correction of document images.

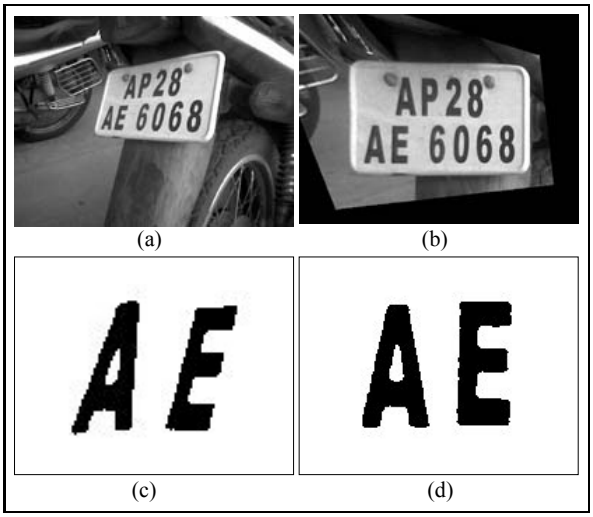


Figure 1. Original Image of a license plate (a) and its perspective corrected version (b). Two selected characters ('A' and 'E') are shown before (c) and after (d) rectification.

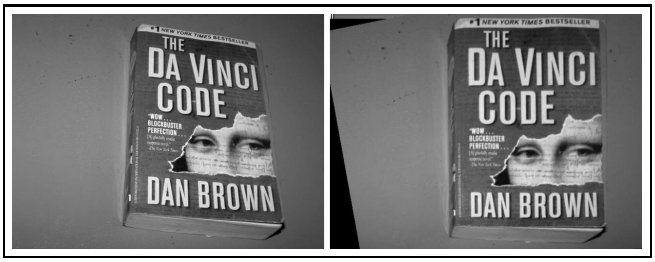


Figure 2. Rectification of the book image. Input image (shown on left) is corrected to obtain the fontal view (right).

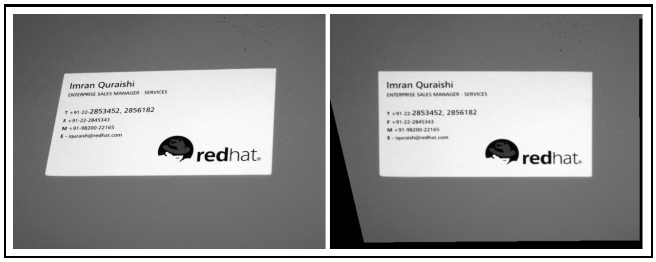


Figure 3. A perspective image of a visiting card is rectified with the help of its bounding rectangle.

The emerging area of camera based document analysis can bene t a lot from the recent results in projective geometry of planar (and even non planar) surfaces [4]. Conventional document image understanding problems were formulated to take care of similarity transformation (translation, rotation and/or scaling) by assuming orthographic projection. The distortion introduced by a projective transformation is more general and apriori knowledge about the image is necessary for accurate recti cation. Perspective distortion does not preserve distances between the points, angles etc. which are normally used to correct skew.

In this paper, we describe a series of techniques for perspective correction based on the imaging model described in the next section. We show that recti cation is possible based on the document boundaries, document layout or document content. In all these situations, basic algorithm is described and results are shown on sample images. Major contribution of this paper is in demonstrating the intelligent use of commonly available clues for perspective recti cation, rather than in proposing a problem speci c recti cation technique.

2. Camera-Based Imaging

Though different camera models exist, the pinhole camera model is popular because of its mathematical tractability. The image formation equation for such a camera is

$$\mathbf{p} = \mathcal{M}\mathbf{P} \tag{1}$$

where $\mathbf{p}$ is the image point, $\mathbf{P}$ is the world point and $\mathcal{M}$ is the camera matrix composed of the internal calibration parameters and external parameters like the pose of the camera. Points $\mathbf{p}$, and $\mathbf{P}$ are represented in homogeneous coordinates and are of dimensions 3×1 and 4×1 respectively. The camera matrix $\mathcal{M}$ is a 3×4 matrix.

A perspective camera preserves line incidences and induces a linear transformation in a projective space. Parallel lines in projective space intersect at a point at in nity . Imaging a planar document image can be understood also as a projective (general linear) transformation of the world document [4] to an image plane. When you image with a pinhole camera, parallel lines cease to be parallel and intersect at the point corresponding to the transformed point at in nity. Sets of parallel lines intersect at different points at in nity, and all of them lie on the line at in nity , l_∞. Since a projective transformation preserves collinearity, the line at in nity remains a line after transformation. The determination of this line in the transformed space (in the image) aids in the perspective recti cation process. An image can be recti ed by mapping this line in the transformed space to the line at in nity ($l_\infty = [0, 0, 1]^T$).

When planar objects are imaged, the images observed from multiple views are related by a linear projective transformation, referred to as Homography.

$$\mathbf{x_i}' = \mathbf{H}\mathbf{x_i} \tag{2}$$

$\mathbf{x_i}'$ and $\mathbf{x_i}$ are 3×1 vectors and could correspond to the images of a same point. The homography $\mathbf{H}$ is a matrix of size 3×3. This is de ned only upto a scaling and hence has only 8 unknowns. Given four corresponding points (8 equations) in a general position, $\mathbf{H}$ can be uniquely computed. Perspective recti cation involves recovery of the frontal view of the image by determining the homography starting from an arbitrary view. Corresponding points in two images are related by a linear transformation in the projective space. If $\mathbf{x}' = [x', y']^T$ and $\mathbf{x} = [x, y]^T$, the corresponding points in two images related by a homography then,

$$\begin{aligned} x' &= \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \\ y' &= \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \end{aligned} \tag{3}$$

The frontal view of an image can be recovered only upto a uniform scale if we can compute the homography. Since there are 8 unknowns a minimum of 8 equations are needed

to compute Homography. These can be computed from correspondences of 4 points [4].

A projective homography can be understood to be the product of three components – similarity(H_s), affine (H_a) and projective (H_p), i.e., $H = H_s H_a H_p$. We are interested in removing the projective and affine components to obtain a similarity transformed (i.e., translated, rotated and/or scaled) version of the original image.

In the next three sections, we describe techniques which will allow us to compute the homography directly or at least its projective and affine components. Application of these transformation to the image results in a perspectively corrected document image. In this paper, we do not discuss the image processing steps needed for the implementation of some of these algorithms.

3. Document Boundaries

Text is omnipresent, however an important clue of textual content is its well distinguishable boundary. Consider the following applications:

1. A camera-based scanner designed to digitize books and manuscripts for a digital library.
2. A camera-phone based application to read and index visiting cards.
3. A camera to analyze 3D world by reading signboards and license plates.

In the above situations, document image boundary can be very useful for projective rectification. When the rectangular boundary is clearly distinguishable, it is possible to correct the image using the techniques describe below.

3.1. Aspect Ratio of the Documents

In most cases text is contained within well-defined rectangular boundaries. Rectangles after undergoing a projective transformation result in quadrilaterals. The vertices of the quadrilaterals could be used to obtain the homography between an arbitrary view to the frontal view. In case the original aspect ratio of the rectangle is known, then the vertices of the quadrilateral in the image can be mapped to the corners of the known rectangle. Thus exact rectification could be achieved. The basic algorithm is given below,

Algorithm

1. Identify the corners of the bounding quadrilateral in the given image.
2. Map each vertex of the quadrilateral to the corresponding vertex in the known rectangle.
3. Using equations like (Eq 3) and an additional constraint (eg. $||h||$ is of unit norm), find the corresponding coefficients h_{ij} of the Homography $\mathbf{H}$.
4. Using H rectify the image to the frontal view.

3.2. Parallel and Perpendicular lines

The document can also be corrected if two pairs of parallel lines and two pairs of perpendicular lines(in the original image) could be identified. This is useful if the explicit aspect ratio of the document is not available. It is also argued that the line detection is more reliable than point identification. We start with a pair of parallel lines (say opposite sides of the document image). The line passing through the two points of intersection of these pair of lines is a projective transformed version of the line at infinity $\mathbf{l}_\infty$. A transformation which maps this observed line to $\mathbf{l}_\infty$ is applied to remove the projective component of the homography H_p. If $\mathbf{l} = [l_1, l_2, l_3]^T$ represents the observed line at infinity, the pure projective transformation for rectification is given by,

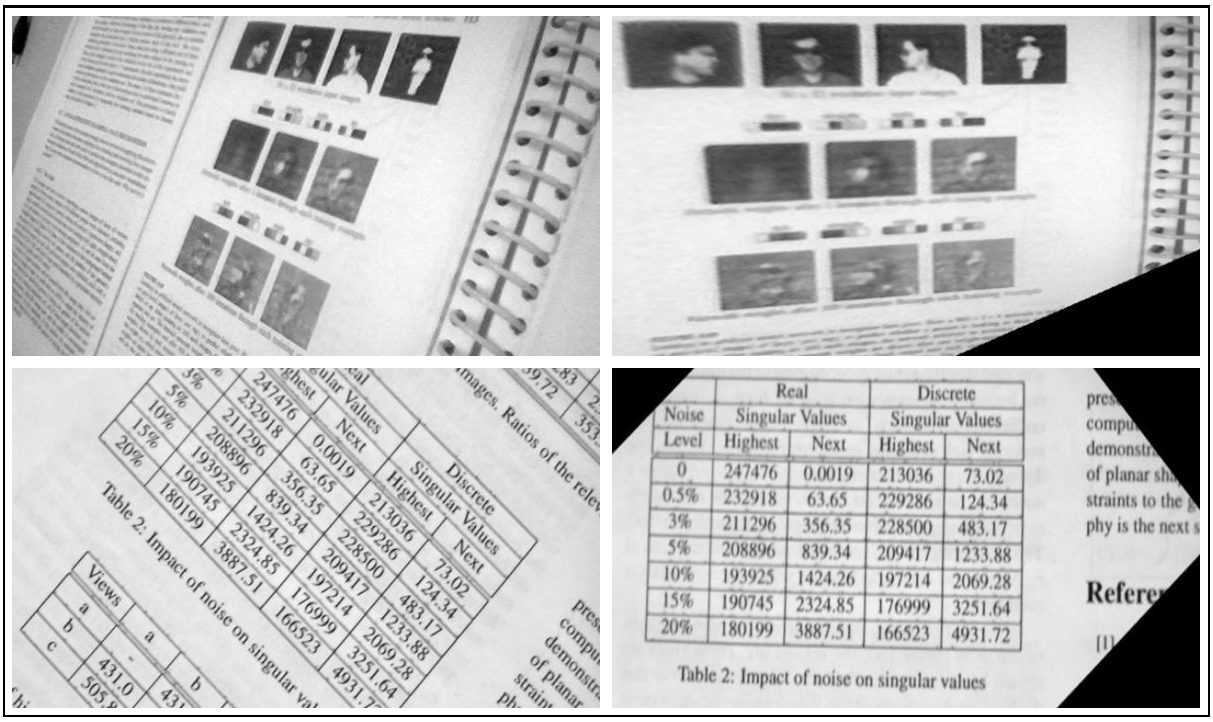
$$H_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix} \tag{4}$$

After a projective transformation parallel lines cease to remain parallel and perpendicular lines do not remain perpendicular. However after removing the pure projective component, parallelism is preserved, while perpendicularity is not preserved. Such images still have affine (H_a) and similarity (H_s) components.

The affine component (H_a) can be similarly determined using pairs of perpendicular lines. We identify a transformation H_a which rectify the angle between a pair of lines. By finding a transformation, which maps the pair of (originally) perpendicular lines to perpendicular (in the image), we can remove the affine component H_a. This can be done by identifying an absolute conic [4]. An image of a planar surface, where the projective component and affine components are removed, has only a similarity component left out. This image is ideal for conventional document image analysis algorithms.

Algorithm

1. Identify a pair of parallel lines which intersect in the perspective image.
2. Find the equation of the line ($\mathbf{l} = [l_1, l_2, l_3]$) passing through these points and rectify the image by removing projective component. ($\mathbf{l}_\infty = H_p^{-1}\mathbf{l}$)
3. Identify a pair of perpendicular lines and remove the affine components.
4. Resultant image is a frontal view (similarity transformed) version of the original image.



Noise Level	Real Singular Values Highest	Real Singular Values Next	Discrete Singular Values Highest	Discrete Singular Values Next
0	247476	0.0019	213036	73.02
0.5%	232918	63.65	229286	124.34
3%	211296	356.35	228500	483.17
5%	208896	839.34	209417	1233.88
10%	193925	1424.26	197214	2069.28
15%	190745	2324.85	176999	3251.64
20%	180199	3887.51	166523	4931.72

Table 2: Impact of noise on singular values

Figure 4. Correction of Perspective Distortion using Layout Information. Graphics blocks can also be used effectively for this purpose.

Results We demonstrate the results of the above mentioned algorithms in three different situations. A license plate imaged through a CCD camera is shown recti ed with the help of its boundary in Figure 1 (a) and (b). Two characters are shown isolated in Figure 1 (c), with their recti ed images in Figure 1(d). An image of a book is shown recti- ed with the help of its four corners in Figure 2. The boundary of the book is clearly distinguishable from the background and hence the bounding quadrilateral is determined. A sample image of a visiting card under perspective imaging is shown in Figure 3 with its recti ed version. The recti ed text is suitable for recognition which is dif cult in the projective domain.

4. Rectification Using Page Layout

Document layout is another powerful clue for perspective correction of document images. Layout as well as the structural information of document images could be useful in multiple ways. Some of them could include:

1. Repetitive or apriori known structure of cells in tables can be a very useful clue for recti cation of forms, imaged using camera based scanners. Boxes provided for writing pin codes could be useful while digitizing and rectifying postal addresses.
2. Column layout information of pages or layout-template used by a speci c publisher or magazine could be equally useful in perspective correction. Limited previous work exists in this direction in the form of perspective recti cation using paragraph boundaries [1].
3. Text and graphics block present in the image can also be used as evidence for recti cation. Often they are rectangular in nature, with sides aligned and follow a Manhattan layout.

Recti cation of tables and forms often results in large number of equations for homography estimation and thereby perspective correction. When there are more than four point correspondences (or eight equations) for estimation of homography, the recti cation can be done more robustly. In the previous section, we had seen a direct approach to compute the planar homography from four point correspondences by solving $\mathbf{x}_i = H\mathbf{x}'_i$. In presence of more than four points a system of homogeneous equation of the form $\mathbf{Ah} = \mathbf{0}$ is constructed, where the homography H is rearranged as a 9×1 vector $\mathbf{h}$. The solution to this system of equation is the eigen vector corresponding to the smallest eigen value of $\mathbf{A^T A}$. For numerical stability, the image coordinates in the $\mathbf{A}$ are normalized such that they are centered around zero and have unit

Figure 5. Identification of Vanishing Points Robustly Leads to Rectification. Many of our document images have enough clues for identification of vanishing lines in the form of parallel and perpendicular lines.

variance [4]

Algorithm

1. Identify multiple corresponding points in the image and the reference frame from the apriori known layout information (like two column document printed on a A4 page) or tables/forms with repetitive structure.
2. Form a homogeneous system of equations $\mathbf{A}\mathbf{h} = 0$ by rearranging the terms of Equation 3. ($\mathbf{A}$ is of dimension $n \times 9$, where $n >> 9$)
3. Solve the system by nding the eigen vector corresponding to the smallest eigen value of the matrix $\mathbf{A}^T\mathbf{A}$.

The boundaries of the textual/graphics boundary can also be used for homography computation and recti cation. Clark and Mirmehdi [1] demonstrate how paragraph alignment could be used in computing vanishing points. They suggest a method that uses the text lines and the alignment of the document to correct the projective distortion. When the image does not contain graphical elements, then homography could be deduced from the structure of the text itself. In document images with Manhattan layout, identi - cation of many horizontal and vertical lines is quite possible. These liner structures could be due to the text blocks, graphic blocks and document column boundaries. These lines also correspond to the paragraph boundaries or text lines as employed by [1]. We employ these clues in computing the horizontal and vertical vanishing points. This class of algorithms can be summarized as follows. More technical background needed for this algorithm could be seen in references [1, 2]

Algorithm

1. Using projection pro les nd the lines most likely horizontal in the recti ed document.
2. Alternately employ Hough Transform based method for nding the prominent direction of lines in a blurred version of the document image.
3. Using these information about the liner structure nd horizontal and vertical vanishing points.
4. Using vanishing points and rectify the image.

Results We demonstrate the application of these procedure on large number of example situations, They include the forms (Figure 5 bottom), Bar-codes (Figure 5 top). Forms contain many cells which can give better, robust estimates of vanishing points and can be used effectively to estimate homography. Bar codes contain many vertical line seg-

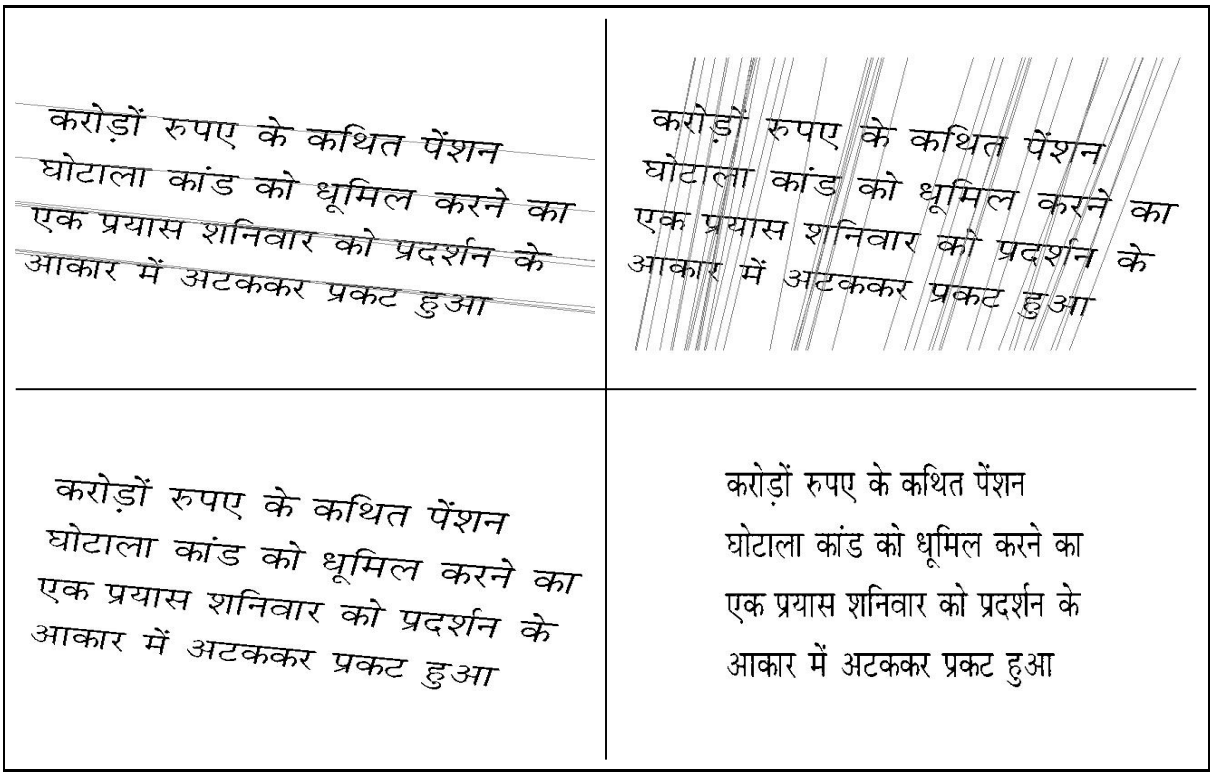


Figure 6. Identification of Horizontal and Vertical Vanishing Points and Perspective Correction of Hindi Text

ments, which are used to estimate vertical vanishing point; while their tips can be joined to form horizontal lines which are then used to nd horizontal vanishing points. Graphics units and tables are used for perspective correction in Figure 4. Note that even though application-wise they are different, fundamental method behind all these perspective recti cation technique is the intelligent use of clues which are hidden in the document layout.

5. Content Specific Rectification

Document boundaries and page layouts give useful information to aid recti cation. However, when the amount of text present is less (a few words or a few sentences) little knowledge is available about the layout or the boundary. Explicit knowledge of rectangles or quadrilaterals are absent and hence we need content speci c information to be used for projective correction. In such cases the properties of text (or the content of the image itself) could be used for recti cation.

Indic Scripts Text written in Indian scripts like Devanagari and Bangla have a *Sirorekha*, a horizontal line connecting the individual characters at the top. This information can be effectively used in estimating the horizontal vanishing point. Once the horizontal vanishing point is estimated an approximate idea of the position of the vertical vanishing point is known. This is because the vertical vanishing point would be approximately found perpendicular to the line of text. Using the hough transform technique and rejecting lines closer to the horizontal vanishing point we estimate a set of lines that would intersect at the vertical vanishing point. Determination of the vanishing point is strengthened by rejecting outliers and re ning the number of lines intersecting at the vanishing point.

Algorithm

1. Identify the horizontal vanishing point using the *Sirorekha*.
2. Generate projection pro les for lines closer to the perpendicular of the line joining the horizontal vanishing point with the centre of the document. Identify k best lines with the highest projection pro le and Compute the intersecting point.
3. Reject outliers among the lines if they lie too far from the intersecting point and recompute the intersection of all the lines and thereby the vertical vanishing point.

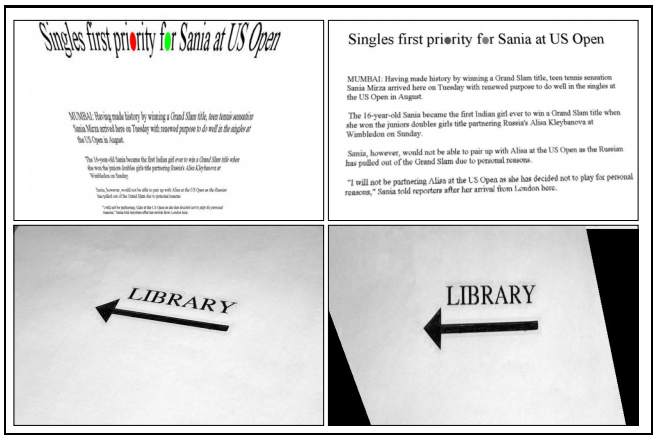


Figure 7. Rectification of Document Images using Content Specific Clues

Figure 6(c) shows perspective image of a Hindi document image. Figure 6(a) shows the determination of the horizontal vanishing point using the *Sirorekha*. We observe that the method is highly accurate. The vertical vanishing point is determined by ignoring some of the outliers produced. Figure 6(b) shows the lines that were close to the vertical line. The recti ed image is shown in Figure 6(d).

Rectification using apriori known symbols There exist many effective methods for projective recti cation of document images, when some apriori information is available about the content in the document images. Figure 7 (a) shows a perspectively distorted page and its recti ed version in Figure 7(b) as reported in [6]. This assumes the presence of conics in the document images for recti cation. The image shown in Figure 7(c) is recti ed using a method described in [5]. The contour of an apriori known shape (in this case the arrow symbol) is used for perspective recti - cation. Note that this method does not need explicit point correspondences. It needs only the contour of the 2D object, which is in general robust to compute. Such methods could be very effective in domain speci c reading systems.

6. Conclusions

In this paper, we have described various methods for projective correction of document images. Knowledge of document location and environment, structure of the document, content of the document etc. are used in obtaining the fronto-parallel view of the image. In this paper, we have not discussed the low-level implementation details of the algorithms. Focus has been in exploiting the hidden clues in document images for effective projective correction. Another important area of future interest is the geometric correction of non-planar surfaces.

References

[1] P. Clark and M. Mirmehdi. Estimating the orientation and recovery of text planes in a single image. *Proc. 12th British Machine Vision Conference*, pages 421–430, 2001.

[2] A. Criminisi and A. Zisserman. Shape from texture: Homography revisited. *Proceeding International Conference of Document Analysis and Recognition*, pages 606–616, 2003.

[3] D. Doerman, J. Liang, and H. Li. Progress in camera-based document image analysis. *Proceeding International Conference of Document Analysis and Recognition*, pages 606–616, 2003.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[5] M. P. Kumar, C. V. Jawahar, and P. J. Narayanan. Building blocks for autonomous navigation using contour correspondence. *Proceeding of International Conference of Image Processing*, 2004.

[6] M. P. Kumar, C. V. Jawahar, and P. J. Narayanan. Geometric structure computation from conics. *Proceeding of Indian Conference on Computer Vision Graphics and Image Processing*, pages 1–6, 2004.

[7] R. Lienhart and A. Wernicle. Localizing and segmenting text in images and videos. *IEEE TCSVT*, 12(4):256–268, 2002.

Keynote Presentation

New Chances and New Challenges in Camera-Based Document Analysis and Recognition

In-Jung Kim
Ph.D. / Manager / Senior Research Engineer
Cognitive Engineering Research Center, Inzisoft, Korea
e-mail: ijkim@inzisoft.com

During the past decades, document recognition systems were mainly developed for the scanner devices. However, as the digital camera is getting popular drastically and its performance is being improved rapidly, the digital camera became an attractive alternative imaging device for the document recognition system.

The popularization of digital camera can significantly extend the application area of pattern recognition technology, because the camera is portable, convenient and provides freedom to capture any object under any environment. However, it also brings new challenges to pattern recognition researchers.

This presentation will describe the new chances and the new challenges introduced by the popularization of digital camera. Then, it will propose a perspective to understand CBDAR system as well as some suggestions for the research direction of CBDAR. Finally, a brief description of mobile environment which is closely related to the camera device and some experiences in developing a camera based OCR product will be presented.

Author Index