

Camera Text Recognition based on Perspective Invariants

Shijian Lu, Chew Lim Tan
Department of Computer Science
National University of Singapore, Kent Ridge, 117543
{lusj, tancl@comp.nus.edu.sg}

Abstract

As camera resolution increases, high-speed non-contact text capture through a digital camera is opening up a new channel for document capture and understanding. Unfortunately, perspective and geometric distortions in camera image of documents make it hard to recognize the document content properly. In this paper, we propose a character recognition technique, which is capable of recognizing camera text lying over a planar or smoothly curved surface in perspective views. In our proposed method, a few perspective invariants including character ascender and descender, centroid intersection numbers, and water reservoir are first detected. Camera texts are then recognized using a classification and regression tree (CART) structure. Experimental results show our method is fast and improves recognition performance greatly.

1 Introduction

As sensor resolution increases, more and more documents are captured through a digital camera due to its superior performance in speed, flexibility, and portability. Unfortunately, camera texts are normally coupled with two types of distortion, namely, perspective distortion introduced through the capturing process and geometric distortion resulting from the non-flat document surface. The two types of distortion may deteriorate the performance of the generic OCR systems seriously. The recognition techniques tolerant to various distortions are more desirable for the recognition of document text captured by a digital camera.

Geometric transformations and related invariants have been widely exploited for handwritten text recognition [1]. In [4], Oscar proposed a transformation-invariant character recognition technique that is able to accommodate a wide class of geometric transformation. In his method, each character is represented with a continuous family of Hidden Markov Models (HMMs) that is parameterized with the scale, slant, and other transformation parameters. Based on

the constructed HMMs, transformation parameters are finally determined through scoring each family and searching for the parameter values that maximizes the scores.

In [5, 6], the perturbation methods are proposed with the aim of distortion-tolerant text matching. The perturbation method tries to reverse the distortion through restoring the input image to one of the standard templates by using a set of geometrical transformations including rotation, slant, and some other ones. In [7], Wakahara introduces a handwritten text recognition technique where the shape deformation is modeled using the local and global affine transformation. Optimal transformation are finally determined through the iterative application of weighted least-squares fitting based on the input pattern and reference patterns.

In [8], Zenzo suggests a feature-based handwritten recognition approach where a set of shape features that are invariant to position, scaling, and rotation are exploited. In this paper, we take a similar approach for the recognition of camera text in perspective views. We propose to recognize camera texts using a set of character shape features that are tolerant to perspective and geometric distortion. Camera texts are first segmented and horizontal and vertical text direction are determined. A set of perspective invariants including character ascender and descender, centroid intersection numbers, and water reservoirs are then detected. Camera texts are finally recognized using a classification and regression tree (CART) structure.

2 Camera Text Recognition

In this section, we present the procedure of camera text recognition. In particular, we will divide this section into four subsections, which deal with camera document preprocessing, camera document analysis, perspective invariant extraction, and camera text classification, respectively.

2.1 Camera Document Preprocessing

Document text must be segmented from the background before the ensuing processing. Compared with scanned

CAD	acei, m-o, rs, u-x, z	bdfhklt	gpqyj
VCR	fh-j, l-n, rt, u-y	b-d, k, o-q	ae <i>g</i> *sz
LWR	b-f, h-r, t, u-w	ags, x-z	
RWR	ab, d, g-j, l-r, t, u-w, y	cefkstxz	
TWR	ab, de, f-j, l-t, z	cekuvxy	w
BWR	ab, d-g, ij, l, o-t, uv, yz	chknsxw	m

Table 1. Character classification based on perspective invariants.

documents, camera documents are normally more susceptible to shading degradation. Therefore, adaptive thresholding technique is normally required. We adopt Niblack’s method [2] since his method generally outperforms others in terms of speed and segmentation quality.

Preprocessing is then accomplished through two rounds of size filtering. Noise of small sizes is first removed through the first round filtering. We set the threshold at 10 because labeled character components normally contain much more than 10 pixels. $Size_{mdn}$, the median size of the remaining document components, is then determined through a simple sorting process. Noise of bigger size and text components of smaller size are further removed through the second round filtering where the threshold is set at:

$$T = k_t \cdot Size_{mdn} \quad (1)$$

where k_t normally lies between 0.2-0.4. We set it at 0.3 in our system. Punctuation and small character components such as the top part of “i” and “j” are accordingly removed.

2.2 Camera Document Analysis

Horizontal and vertical text direction must be determined before the ensuing feature detection. We estimate vertical text directions through the identification of character i and l . Horizontal direction is then determined based on the x -line and base line of text.

Characters i and l always indicate vertical text direction in the presence of perspective distortion. They can be identified from other characters based on the distance:

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

where n denotes the number of character pixels and l refers to the straight line determined through the least square fitting of character pixels. Function d gives the distance between the i^{th} pixel p_i and the straight line l .

We fit x lines and base lines through the classification of character extremum points (CEP), which refer to the highest and lowest character pixels in the vertical text direction

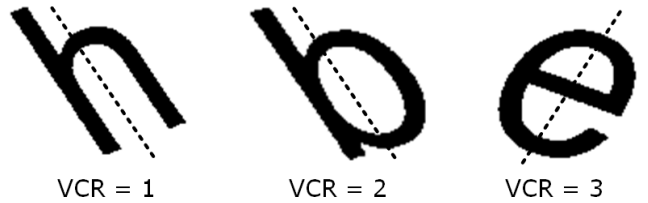


Figure 1. The number of centroid intersections for characters “h”, “b”, and “e”.

determined above. Therefore, each labeled character holds two CEP lying around the related x -line and base line positions, respectively. We adopt the point tracing technique [3] to classify CEP to different text lines.

For documents lying over a planar or smoothly curved surface, the shape of text lines within the camera images can be modeled using a cubic polynomial curve in most cases. The horizontal character direction can accordingly be determined as the tangent of the related x -line or base line at the character centroid position. The x and y axes in Figure 2 illustrate the horizontal and vertical text direction.

2.3 Perspective Invariant Extraction

The first invariant feature is character ascender and descender (CAD). CAD remains invariant to the perspective distortion as they always lie far above or below the x -lines or base line in perspective views. Practically, CAD can be easily detected based on the distance between the CEP and the related x -lines and base lines of text.

With the CAD, lowercase Roman letters can be classified into three categories as given in the first row in Table 1. It should be clarified that the top part of character i and j has been filtered out in the preprocessing in Section 2.1. Besides CAD, the position of CEP can be further exploited for character classification. For example, taking vertical text direction as the reference, the top CEP of character “b” always lie to the left of character centroid, but the top CEP of “d” keeps lying to the right of character centroid instead.

The second perspective invariant refers to the number of vertical character runs (VCR), which count the number of character strokes in vertical direction. We define VCR as the number of intersections between character strokes and a vertical scan lines that passes through the character centroid with orientation being the same as that of the nearest i or l . VCR is perspective invariant because the number of character strokes remain unchanged in perspective view.

Figure 1 illustrates the definition of VCR. Similar to the CAD, the VCR is tolerant to character size and fonts as well. Though some text font such as “Time New Roman” may contain serif at the x -line and base line position, the VCR

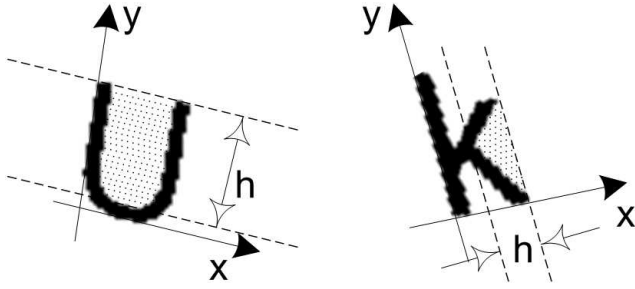


Figure 2. Top and right water reservoirs of distorted characters “u” and “k”.

will not be affected as serif does not affect the VCR defined above. The only exception is character *g* whose VCR number may be 3 or 4, depending on the font of the character typed. Based on VCR information, lowercase Roman characters can be classified into three categories as given in the second row in Table 1 where the number of VCR in three columns are 1, 2, and 3, respectively.

The third perspective invariant refers to water reservoir (WR) [9]. The concept of WR is quite simple. If we pour water from left, right, top, and bottom of character, the cavity regions of the character where water will be stored are defined as a WR. As illustrated in Figure 2, WR labeled by dotted regions always exist despite perspective distortion. Furthermore, the height of WR *h* relative to character size (measured by the height of *x* zone at the position of the studied character) normally varies within a small range.

WR can be detected through character boundary analysis. Based on the number of WR, lowercase characters can be classified as given in Table 1 where the row TWR, BWR, LWR, and RWR (top, bottom, left, and right WR) give classification results. The three columns in each rows group characters with 0, 1, and 2 WR, respectively. Besides the number of WR, the position of WR can also be exploited for character classification. For example, characters *c* and *k* both have a top and a bottom WR. But the centroid of the top WR of *c* and *k* are below and above the character centroid, respectively. Furthermore, the relative position of two WR can also be exploited for character classification. For example, characters *s* and *z* both have a left and a right WR. But the right WR of “*s*” is above the left one, whereas the right WR of “*z*” is below the left one instead.

2.4 Camera Text Classification

With the three perspective invariants, lowercase character can be classified using a CART structure as illustrated in Figure 3. The rectangles enclose characters to be classified and the terms below refer to the invariants that are ex-

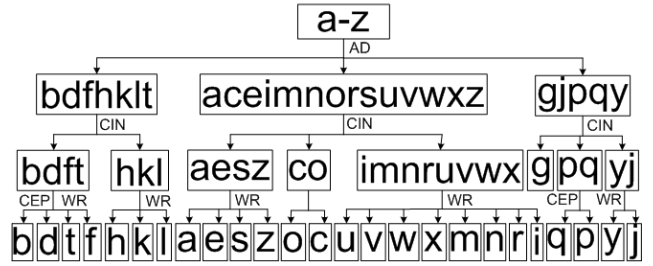


Figure 3. Character classification based on the CART structure.

ploited for the classification of characters within the rectangle above. With the 3 perspective invariants, we construct 26 rules for the classification of 26 lowercase characters. For example, character *h* can be uniquely identified by an ascender, single VCR, and single bottom WR and character “*m*” can be simply identified by two bottom WRs.

For document with rotation angle bigger than 90 degrees or smaller than -90 degrees, the detected CAD and WR features are totally different from the real ones. Such upside down situation can be first detected based on the statistics that the number of character ascender is nearly always much bigger than that of character descender for documents with a large number of characters. If documents are captured upside down, the CAD and WR features must be adjusted properly before the ensuing character classification.

It should be noted that we focus on lowercase character classification just for presentation clarity. In fact, the proposed perspective invariants are sufficient for the classification of 26 uppercase characters. For example, character *B* and *E* can be uniquely identified by a character ascender & three VCR and two right WR, respectively.

3 Experiments and Discussion

The proposed method has been implemented in C++ and some preliminary experiments have been conducted. We create 40 testing documents collected from books, webs, and proceedings and each document contains around 30 text lines and 2900 characters on the average. The 40 documents are first fixed over a planar surface and captured using a digital camera of 7 mega pixels. Then the same 40 documents are fixed over differently curved surfaces and captured with the same resolution. We keep the angle between camera optical axis and document normal smaller than 60 degrees so that the text can be viewed and segmented correctly.

Currently, each document takes around 12 seconds for text recognition. The execution time can be further reduced through code optimization. To evaluate the performance

