A multi-stage strategy to perspective rectification for mobile phone camera-based document images

Xu-Cheng Yin Jun Sun Satoshi Naoi
Fujitsu R&D Co. Ltd, Beijing, China
{xuchengyin; sunjun; naoi}@cn.fujitsu.com

Katsuhito Fujimoto Hiroaki Takebe Yusaku Fujii Koji Kurokawa
Fujitsu Laboratories Ltd, Kawasaki, Japan
{fujimoto.kat; takebe.hiroaki; fujii.yusaku; cross}@jp.fujitsu.com

Abstract

Document images captured by a mobile phone camera often have perspective distortions. Efficiency and accuracy are two important issues in designing a rectification system for such perspective documents. In this paper, we propose a new perspective rectification system based on vanishing point detection. This system achieves both the desired efficiency and accuracy using a multi-stage strategy: at the first stage, document boundaries and straight lines are used to compute vanishing points; at the second stage, text baselines and block aligns are utilized; and at the last stage, character tilt orientations are voted for the vertical vanishing point. A profit function is introduced to evaluate the reliability of detected vanishing points at each stage. If vanishing points at one stage are reliable, then rectification is ended at that stage. Otherwise, our method continues to seek more reliable vanishing points in the next stage. We have tested this method with more than 400 *images including paper documents, signboards and posters. The image acceptance rate is more than* 98.5% *with an average speed of only about* $60ms$.

1. Introduction

Recently, camera-based document analysis becomes a hot research field [5, 9]. With widespread usage of cheap digital ***cam***eras built-in the ***mobile*** phone (***MobileCam*** in abbreviation thereafter) in people's daily life, the demand for simple, instantaneous capture of document images emerges. Different from the traditional scanned image, lots of the MobileCam based document images have perspective distortions. Consequently, rectifying MobileCam based perspective document images becomes an important issue. In document analysis, there are various works on correction of perspective documents captured by general digital cameras [1, 2, 3, 4, 6, 7, 8]. Many of these methods use document boundaries and text lines to vote vanishing points.

There are several challenges for rectifying MobileCam based perspective documents, such as speed, robustness, non-boundary documents, etc. [10]. In conclusion, efficiency and accuracy are two important issues in designing a rectification system for MobileCam based perspective documents. In our previous research [10], a hybrid approach to vanishing point detection is proposed to rectify such perspective documents. In this paper, we focus on the rectification system design. Our proposed approach achieves both the desired efficiency and accuracy using a multi-stage strategy: at the first stage, fast vanishing point detection is performed with document boundaries and straight lines; at the second stage, text baseline are robustly extracted and then utilized to calculate vanishing points; and at the third stage, character tilt orientations are voted for the vertical vanishing point. A profit function is introduced to evaluate the reliability of detected vanishing points at each stage. If vanishing points at one stage are reliable, then perspective rectification is performed at the end at that stage and then the whole system is finished. In our method, vanishing point detection is performed by a hybrid approach [10].

The remainder of this paper is organized as follows. Section 2 simply introduces the hybrid approach to vanishing point detection. And in Section 3, we describe the multi-stage rectification strategy. Section 4 is the experiments and result analysis. Finally we conclude the paper in Section 5.

2. The hybrid approach to vanishing point detection [10]

Approaches to vanishing point detection for perspective document rectification can be classified into two approaches: direct and indirect approaches. In our previous work, we propose a hybrid approach for vanishing point detection by integrating the direct and indirect approaches efficiently [10]. This hybrid approach first votes and clusters line intersections into vanishing point candidates. Then projection analysis from perspective views on these candidates is performed. Finally, the vanishing point is obtained by combining the previous two steps.

After all lines are extracted and the line intersections are calculated by line pairs, the intersection points are partitioned by clustering, and cluster centers are selected as reliable vanishing point candidates. And each candidate has a weight from clustering. The weight of the ith candidate, $VP_i(x, y)$, is given by

$$w_i^c(VP_i(x, y)) = \frac{N_i}{\sum_{i=1}^{N_{cluster}} N_i}, \qquad (1)$$

where $N_{cluster}$ is the number of resulting clusters, and N_i is the number of points in the ith cluster. Each of these weights can be regarded as the profit function for the indirect approach. There is

$$f_{indirect}(VP_i(x, y)) = w_i^c(VP_i(x, y)). \qquad (2)$$

In order to get a more stable vanishing point, we use a direct approach to refine vanishing point candidates in the above search space. For each cluster center, projection analysis from a perspective view is performed [1]. And the derivation-squared-sum of the projection profiles is calculated by

$$f'_{direct}(VP_i(x, y)) = \sum_{j=1}^{N_B - 1} (B_{j+1} - B_j)^2, \qquad (3)$$

where B is a projection profile with $VP_i(x, y)$, and N_B is the number of projection bins. This is the profit function for the direct approach. For a computational convenience, the used profit is changed into a coefficient by

$$f_{direct}(VP_i(x, y)) = \frac{f'_{direct}(VP_i(x, y))}{\sum_{i=1}^{N_{cluster}} f'_{direct}(VP_i(x, y))}. \qquad (4)$$

Then we linearly combine Equation (2) and Equation (4),

$$g(VP_i(x, y)) = \frac{1}{2} f_{indirect}(VP_i) + \frac{1}{2} f_{direct}(VP_i). \qquad (5)$$

And the resulting vanishing point is given by

$$VP_k(x, y) = \operatorname*{argmax}_i g(VP_i(x, y)). \qquad (6)$$

The last step is to confirm the resulting vanishing point. We use the horizontal vanishing point to explain the rejection strategy. The derivative-squared-sum of the resulting horizontal vanishing point is $f'_{direct}(V_x, V_y)$, which is calculated by Equation 3. The unchanged horizontal vanishing point is $(-\infty, 0)$. And the derivative-squared-sum of it is $f'_{direct}(-\infty, 0)$. If the following condition is satisfied, then the final horizontal vanishing point is (V_x, V_y):

$$f'_{direct}(V_x, V_y) > (1 + \varepsilon) f'_{direct}(-\infty, 0), \qquad (7)$$

where $0 < \varepsilon < 1$, and in our method, $\varepsilon = 0.1$. Otherwise, we take a rejection strategy, and the final vanishing point will be $(-\infty, 0)$.

3. Multi-stage strategy

In general, there are many clues for horizontal or vertical directions in a perspective document, such as document boundaries, straight lines, text baselines, left and right aligns of text block, character tilt orientations, and so on (see Figure 1). Here, aligns means boundaries of paragraph or text block.

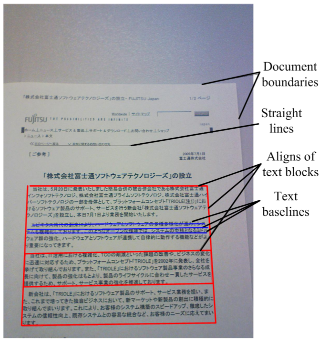

Figure 1. Direction clues for a perspective document.

After survey on more than 1000 MobileCam based document images, we found that there are 10% ˜ 20% document images with full-boundaries; and there are 30% ˜ 40% images which have enough direction clues with straight lines and text baselines for vanishing point detection. However, there are also about 30% images with no-boundaries. Consequently, a multi-stage strategy should be adapted in order to get a more fast and reliably robust performance.

3.1. Method overview

In order to build a system with both high speed and accuracy, we use a three-stage framework for vanishing point detection and evaluation, which is described in Figure 2.

At the first stage, vanishing point detection is based on document boundaries and straight lines. Then, a profit function (see Section 3.2) is used to evaluate the reliability of a detected vanishing point. If the result is enough reliable, then do perspective rectification directly and the rectification system is finished. Otherwise, we perform vanishing point detection at the second stage with computing and voting vanishing points by text baselines in addition to document boundaries and straight lines. If it is not successful,

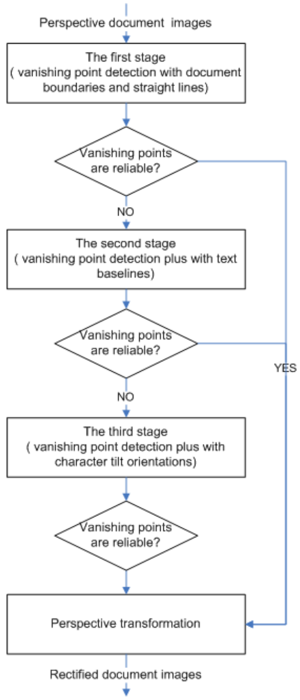
Perspective document images
The first stage
(vanishing point detection with document boundaries and straight lines)
Vanishing points are reliable?
NO
The second stage
(vanishing point detection plus with text baselines)
Vanishing points are reliable?
YES
NO
The third stage
(vanishing point detection plus with character tilt orientations)
Vanishing points are reliable?
Perspective transformation
Rectified document images

Figure 2. The multi-stage strategy.

then vertical vanishing point detection runs at the third stage by character vertical strokes in addition to document boundaries, straight lines and text baselines. In other words, if one horizontal or vertical vanishing point is unreliable, then it will be re-detected in the next stage. If detected vanishing points are still unreliable at the last stage, then the resulting image is the same as the original one. In this method, document boundaries, straight lines, text baselines, and character vertical strokes are detected and extracted by heuristic rules and statistical analysis [10].

3.2. The profit function

As we know, if the number of straight and illusory lines for clustering is large, most of intersection points are near to the vanishing point, the derivative-squared-sum of the projection profiles from a perspective view is largely more than the one of other point candidates, and the detected document boundaries are reliable, then the detected vanishing point will be reliable. Consequently, given a detected vanishing point (successful after the rejection strategy), $VP(x, y)$, a profit function about its reliability can be defined as

$$f(VP(x,y)) = \sum_{i=1}^{N} a_i \times r_i(VP(x,y)) + b, \qquad (8)$$

where a_i is a coefficient, $0 \leq a_i \leq 1$, and $\sum_{i=1}^{N} a_i = 1$. And b is a constant. $r_i(VP(x, y))$ is a reliability value with one aspect, and $0 \leq r_i \leq 1$. In our system, we simply use $a_i = \frac{1}{N}$ for $i = 1, ..., N$, and $b = -0.60$. Though some more adaptive ways with optimization problems can be applied to evaluate these coefficients.

In our method, this profit function is about four aspects ($N = 4$ in Equation (8)): the number of detected lines, the consistency of intersection points, the projection analysis from perspective views, and the document boundaries. If

$$f(VP(x,y)) \geq 0, \qquad (9)$$

then this vanishing point will be reliable; otherwise, we should continue to detect a more reliable vanishing point.

3.3. The first stage

At the first stage, vanishing point detection is performed with document boundaries and straight lines. Because the number of detected lines are small, we just set $r_1(VP(x, y)) = |b|$, and use the consistency of intersection points, the results from projection analysis, and the document boundaries to evaluate the reliability.

After vanishing point detection by the hybrid approach in Section 2, for a detected vanishing point, the consistency of intersection points of a profit value is given by

$$r_2(VP(x,y)) = f_{indirect}(VP(x,y)), \qquad (10)$$

where $f_{indirect}(VP(x, y))$ is the weight about consistency for one cluster in clustering (see Equation (2)). And the profit value about projection analysis is given by

$$r_3(VP(x,y)) = f_{direct}(VP(x,y)), \qquad (11)$$

where $f_{direct}(VP(x, y))$ is a weight by calculating projection profiles from a perspective view (see Equation (4)). As for the reliability of document boundaries, we simply consider the length of boundaries. There are two horizontal boundaries and two vertical boundaries for a document with full-boundary. For a horizontal vanishing point, if two horizontal boundaries both are longer than $\frac{1}{2}$ width of the whole image, then there is

$$r_4(VP(x,y)) = 1.00; \qquad (12)$$

if only one horizontal boundary is longer, then $r_4(VP(x, y)) = 0.50$; otherwise, $r_4(VP(x, y)) = 0.00$. And for a vertical vanishing point, the length threshold is $\frac{1}{2}$ height of the whole image.

Given horizontal and vertical vanishing points detected at this stage, $HVP(x, y)$ and $VVP(x, y)$, if

$$f(HVP(x,y)) \geq 0, \;\; f(VVP(x,y)) \geq 0, \qquad (13)$$

then these horizontal and vertical vanishing points are reliable respectively.

If both horizontal and vertical vanishing points are reliable, then do perspective rectification directly; otherwise, our method again detects the vanishing point for the unreliable one in the next stage.

3.4. The second stage

At this stage, horizontal vanishing point detection is based on horizontal text baselines in addition to document boundaries and straight lines. And the profit value for a horizontal vanishing point about the line number is

$$r_1(VP(x,y)) = \frac{min(n, TH_{line})}{TH_{line}}. \qquad (14)$$

In the above equation, n is the number of all straight horizontal lines and text baselines, and TH_{line} is a threshold for the line number. And in our method, it is simply set by $TH_{line} = 10$. And calculations of other profit values (r_2, r_3, and r_4) are the same as the ones in Section 3.3.

And vertical vanishing point is detected by left and right aligns of the above text baselines in addition to document boundaries and straight lines. Because the number of detected vertical lines are small, we just set $r_1(VP(x,y)) = |b|$ for evaluating the reliability of a vertical vanishing point.

Similar to Section 3.3, if the vertical vanishing point is reliable, then do perspective rectification directly; otherwise, our method again detects a more reliable vertical vanishing point at the third stage.

3.5. The third stage

The third stage is about vertical vanishing point detection by character tilt orientations in addition to document boundaries, straight lines and text baselines. As described in [10], in many situations, vertical clues are scarce. When an image is a partial portion of a whole document, there may be few or even no straight vertical lines. But character tilt orientations can be regarded as clue directions and character vertical strokes can be used as vertical clues [6, 10]. After a stable vertical stroke set is extracted by heuristic rules and statistical analysis, vertical vanishing point is detected by our hybrid approach.

Because each character stroke is very short, we just set $r_4(VP(x,y)) = |b|$. And r_2 and r_3 are the same as the ones in Section 3.3. And the profit value for a vertical vanishing point about the line number is

$$r_1(VP(x,y)) = \frac{min(n, TH_{stroke})}{TH_{stroke}}. \qquad (15)$$

In the above equation, n is the number of all straight vertical lines and character vertical strokes, and TH_{stroke} is a threshold. And in our system, it is simply set by $TH_{stroke} = 100$.

4. Experiments

There are 418 test samples captured by several mobile phone cameras (Fujitsu E882iES mobiles). These images are in RGB color format with a 1280×960 resolution. More than 90% images have perspective distortions. Some samples are shown in Figure 3.

Given a resulting vanishing point, $VP(x,y)$, the relative distance from the ground truth $VP_t(x,y)$ is calculated. If

$$\frac{|VP(x,y) - VP_t(x,y)|}{|VP_t(x,y)|} < T_{VP}, \qquad (16)$$

then $VP(x,y)$ is a correct vanishing point. In our system, the ground truth vanishing points are calculated from manually marked horizontal and vertical lines. When the difference in Equation (16) is less than the threshold ($T_{VP} = 0.05$), then there is no seemingly perspective distortion.

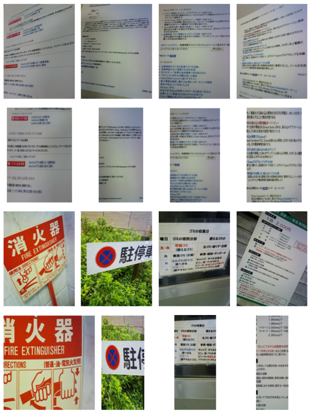

Figure 3. Perspective document samples. (The $1st$ and $3rd$ rows are the original images; and the $2nd$ and $4th$ rows are the corresponding rectified images.)

We divide our rectified images into five groups: (1) HIT, successful for perspective rectification in both horizontal and vertical directions; (2) $HHIT$, successful in the horizontal direction; (3) $VHIT$, successful in the vertical direction; (4) REJ, the rectified image is the same as the original image; (5) ERR, represents rectifying with wrong perspective angles.

In these samples, there are 75 images with full-boundaries which may be performed successfully for the first stage of our multi-stage strategy, 152 images with enough straight lines and text baselines for the second stage, And the remainder 191 document images for the third stage. As a result, if we only use the first stage, the second stage, or the third stage to detect vanishing points, the resulting performance will be largely limited.

In this experiment, we compared our method ($M1$) to our previous method with a hybrid approach to vanishing point detection [10] ($M2$). $M2$ detects a horizontal or vertical vanishing point only one time based on all lines derived from all stages without a multi-stage strategy. The accuracy results are described in Figure 4. Some rectified images with front-parallel views of our method are shown in Figure 3. And the resulting image is the inner rectangle area of the detected perspective quadrangle.

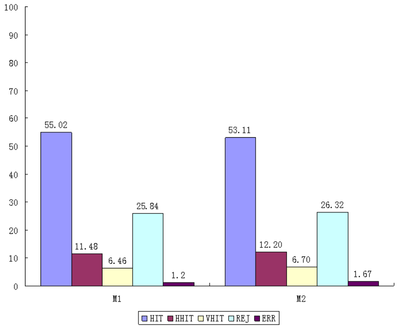

Figure 4. Accuracy (%) of $M1$ and $M2$.

The processing speed is shown in Table 1, where $Time$ represents the average processing time for each image without including the time for the final perspective transformation. Experiments are run on a DELL PC with 3GHz CPU, 2G Memory on a Windows XP OS.

Table 1. Average processing time.

	$M1$	$M2$
$Time$ (ms)	66	103

As shown in Figure 3, test samples include many different types. There are even some street signs and non-document images, and the number of these non-document images is about 20%. The REJ rate of our method is 25.84%, and the partial correct rates ($HHIT$ and $VHIT$) are also high. Most of these are mainly caused by non-document images and too large distortions. And most images with a REJ result are correctly rejected for their large distortions and unstable direction features. For a mobile phone with some proper interactive GUIs, users may accept the results of HIT, $HHIT$, $VHIT$, and REJ because the resulting image from these has a much better quality than (or a same quality as) the originally captured image. In this way, the acceptance rate of our method is 98.80%.

Compared with $M2$, our new method ($M1$) improves the HIT groups by 1.91%. As shown in Table 1, the average processing time of our method is largely less than $M2$, and the reduction time is $37ms$. In our multi-stage method, if vanishing points can be detected with reliable features at one stage, then we rectify perspective document directly at the end of this stage. Vanishing point detection is only performed in several stages for difficult document images. As a result, our new method is more adaptive and fast with a reliable robustness.

In conclusion, the acceptance rate of our new method is more than 98.50%, while the error rate is less than 1.50%. And the processing time is only about $60ms$. With serious or unstable distortions, we take the rejection strategy, which may be more acceptable for a mobile user. All these show that our rectification method is fast and relatively robust.

5. Conclusions

Perspective rectification of MobileCam based document images faces several challenges, such as speed, robustness, non-boundary documents, etc. In this paper, we present a multi-stage strategy to deal with these problems. This method achieves both the desired efficiency and accuracy using a three-stage framework for vanishing point detection. A profit function is introduced to evaluate the reliability of vanishing points at each stage. Only if a vanishing point candidate is unreliable, our method seeks a more reliable one in the next stage. The experiments on different MobileCam based document images show that our method has a good performance with an average speed of about $60ms$ on a regular PC.

References

[1] P. Clark and M. Mirmehdi. Rectifying perspective views of text in 3D scenes using vanishing points. *Pattern Recognition*, 36(11):2673–2686, 2003.

[2] C. R. Dance. Perspective estimation for document images. *Proceedings of SPIE Conference on Document Recognition and Retrieval IX*, pages 244–254, 2002.

[3] J. Liang, D. DeMenthon, and D. Deormann. Flattening curved documents in images. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 338–345, 2005.

[4] J. Liang, D. DeMenthon, and D. Deormann. Camera-based document image mosaicing. *Proceedings of International Conference on Pattern Recognition*, pages 476–479, 2006.

[5] J. Liang, D. Doermann, and H. P. Li. Camera-based analysis of text and documents: A survey. *International Journal on Document Analysis and Recognition*, 7(2-3):84–104, 2005.

[6] 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*, 23(5):541–553, 2005.

[7] C. Monnier, V. Ablavsky, S. Holden, and M. Snorrason. Sequential correction of perspective warp in camera-based documents. *Proceedings of International Conference on Document Analysis and Recognition*, 1:394–398, 2005.

[8] M. Pilu. Extraction of illusory linear clues in perspectively skewed documents. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 363–368, 2001.

[9] S. Pollard and M. Pilu. Building cameras for capturing documents. *International Journal on Document Analysis and Recognition*, 7(2-3):123–137, 2005.

[10] X.-C. Yin, J. Sun, K. Fujimoto, H. Takebe, Y. Fujii, K. Kurokawa, and S. Naoi. Perspective rectification for mobile phone camera-based documents using a hybrid approach to vanishing point detection. *Fujitsu Technical Reports*, 2007.