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 traf-

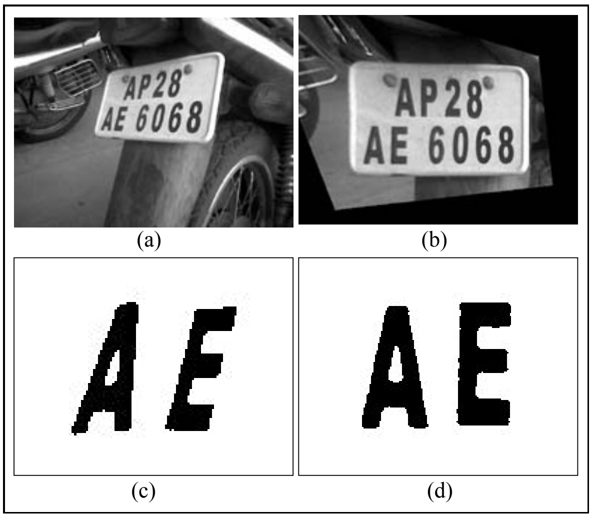

Figure 1. Original Image of a license plate (a) and its perspective corrected version (b). Two selected characters ('A' and 'E') are shown before (c) and after (d) rectification.

c 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) de- ned in a homogeneous coordinate system. Different clues in the document image itself could be used for the purpose of recti cation. 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 recti cation techniques that are useful for projective correction of document images.

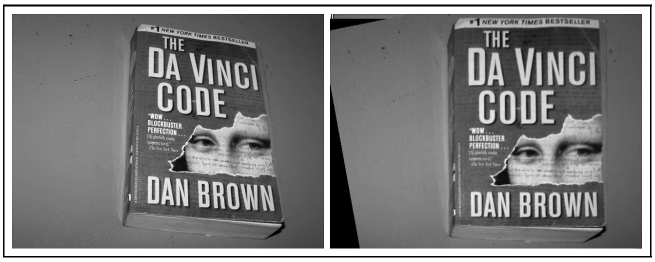

Figure 2. Rectification of the book image. Input image (shown on left) is corrected to obtain the fontal view (right).

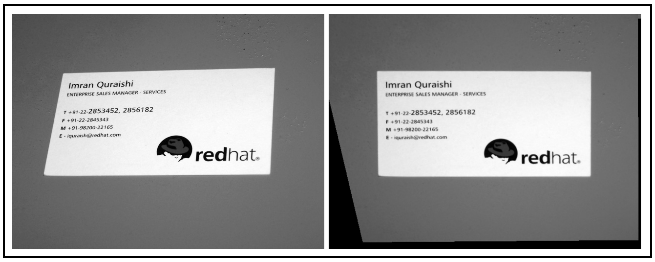

Figure 3. A perspective image of a visiting card is rectified with the help of its bounding rectangle.

The emerging area of camera based document analysis can benefit 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 rectification. 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 rectification 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 rectification, rather than in proposing a problem specific rectification 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 infinity . 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 infinity. Sets of parallel lines intersect at different points at infinity, and all of them lie on the line at infinity , l_∞. Since a projective transformation preserves collinearity, the line at infinity remains a line after transformation. The determination of this line in the transformed space (in the image) aids in the perspective rectification process. An image can be rectified by mapping this line in the transformed space to the line at infinity ($l_\infty = [0, 0, 1]^T$).

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 defined 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 rectification 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.

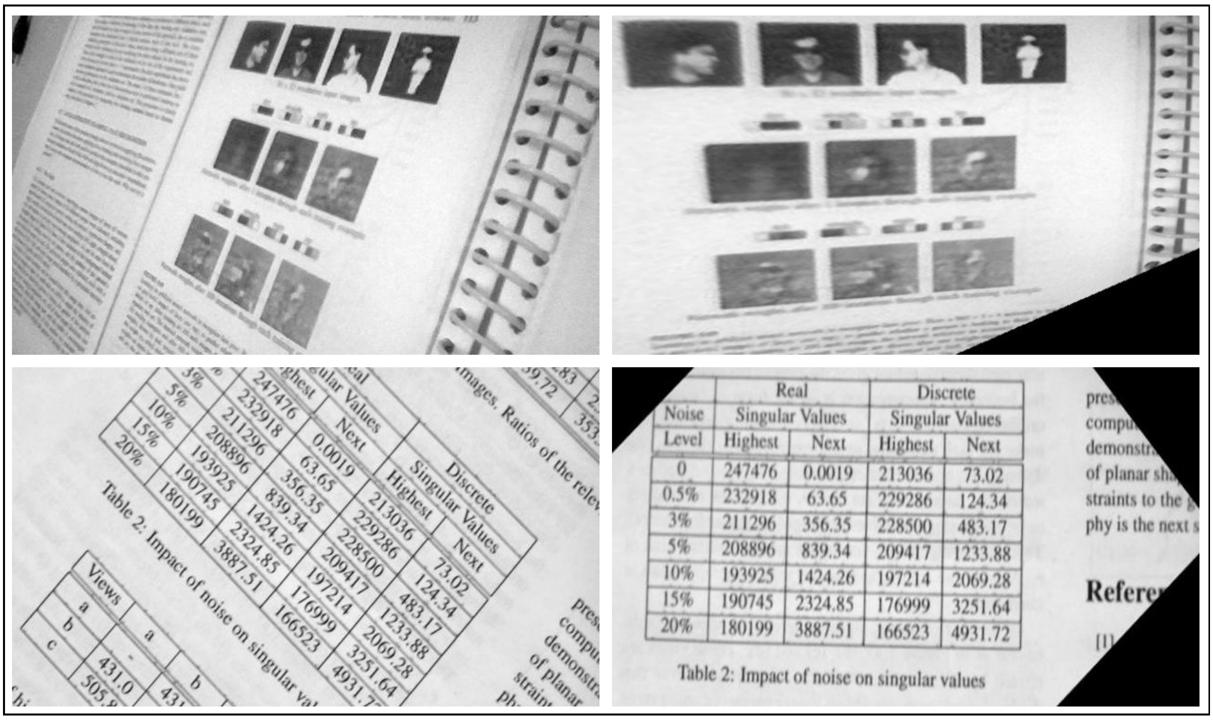

Figure 4. Correction of Perspective Distortion using Layout Information. Graphics blocks can also be used effectively for this purpose.

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 $\mathbf{Ah} = 0$ 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-

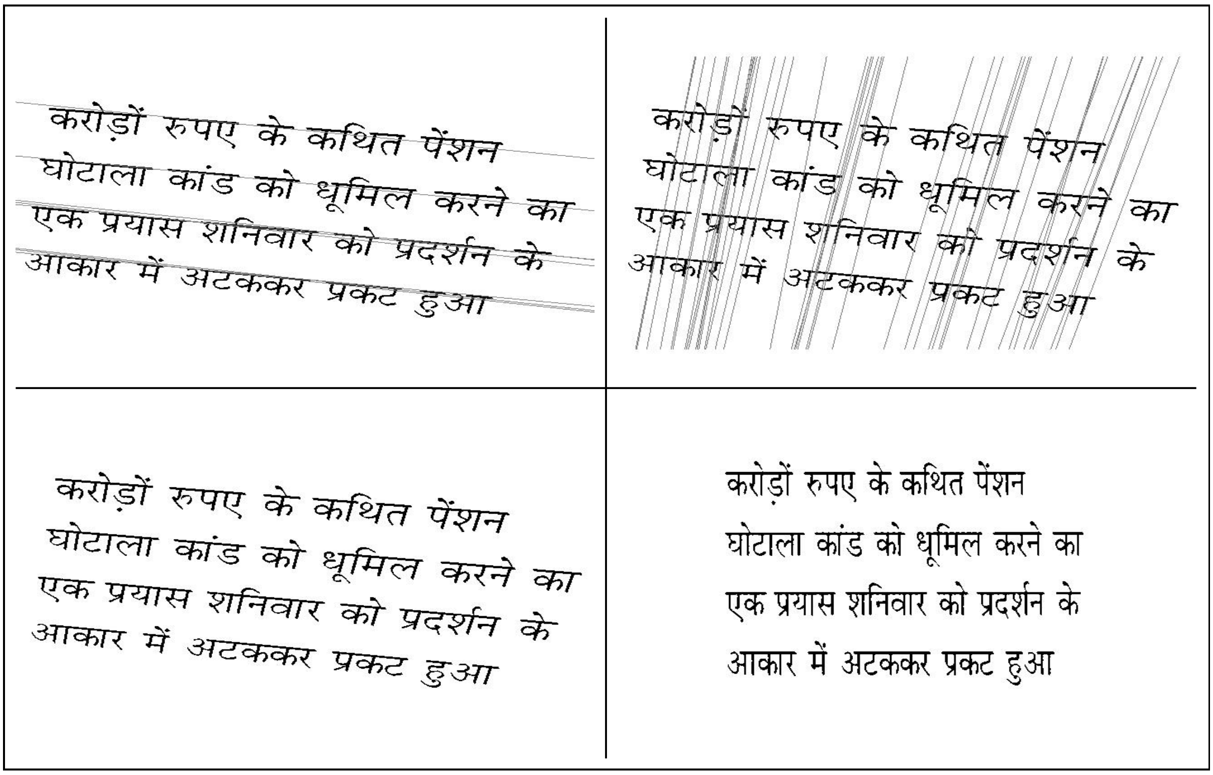

Figure 6. Identification of Horizontal and Vertical Vanishing Points and Perspective Correction of Hindi Text

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.

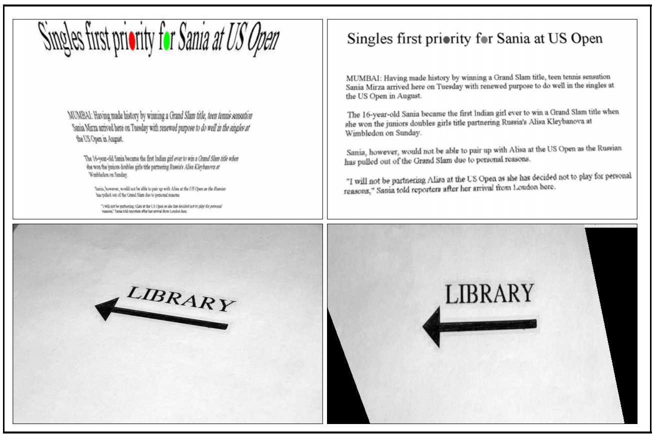

Figure 7. Rectification of Document Images using Content Specific Clues

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.