Geometric Rectification of Camera-captured Document Images

Jian Liang*, Daniel DeMenthon†, and David Doermann†

*Jian Liang is with Amazon.com; Seattle, WA; USA. Email: jliang@amazon.com.

†Daniel DeMenthon and David Doermann are with University of Maryland; College Park, MD; USA.

Abstract

Compared to typical scanners, handheld cameras offer convenient, flexible, portable, and non-contact image capture, which enables many new applications and breathes new life into existing ones. However, camera-captured documents may suffer from distortions caused by non-planar document shape and perspective projection, which lead to failure of current OCR technologies. We present a geometric rectification framework for restoring the frontal-flat view of a document from a single camera-captured image. Our approach estimates 3D document shape from texture flow information obtained directly from the image without requiring additional 3D/metric data or prior camera calibration. Our framework provides a unified solution for both planar and curved documents and can be applied in many, especially mobile, camera-based document analysis applications. Experiments show that our method produces results that are significantly more OCR compatible than the original images.

Index Terms

Camera-based OCR, image rectification, shape estimation, texture flow analysis.

I. INTRODUCTION

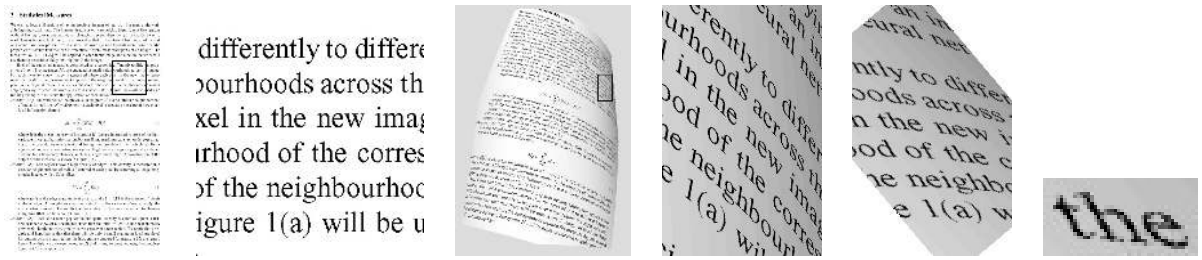
Recent technical advances in digital cameras have led the OCR community to consider using them instead of scanners for document capture [1]. Cameras are portable, long range, and non-contact imaging devices that enable many new document analysis applications. Secondly, cameras can be easily integrated with portable computing devices such as PDAs, cell phones, or media players. And lastly, many more digital cameras are manufactured, distributed and owned than scanners. Together, these factors contribute to the growing interest in camera-based document analysis.

For example, handheld devices equipped with cameras, such as PDAs and cell phones, are ideal platforms for mobile OCR applications such as recognition of street signs in foreign languages, out-of-office digitization of documents, and text-to-voice input for the visually impaired. In the industrial market, high-end cameras have been used for digitizing thick books and fragile historic manuscripts unsuitable for scanning; in the consumer market, camera-based document capture is utilized in the desktop environment [2].

The challenge in camera-based applications is that, due to the differences between scanners and cameras, traditional scanner-oriented OCR techniques are usually not applicable. In particular, non-planar document shape and perspective projection, which are common in camera-captured

images, are not expected at all by traditional OCR algorithms. As a result, the performance of some of the state-of-the-art OCR packages on camera-captured documents is unacceptable.

For instance, Fig. 1 compares a clean scan to a synthetic camera image. First, curved text lines and margins in Fig. 1(c) can easily defeat most page segmentation techniques (e.g., [3], [4], [5]). Second, skewed characters make both segmentation and recognition difficult. To a lesser degree, these challenges also apply to planar pages. Our experiments show that the OCR performance on camera-captured documents, whether planar or curved, is substantially lower than for scanned images. Since we use noise-free, blur-free, high resolution images in these experiments, the influence of other effects is reduced to the minimum, which proves that pure 2D image enhancement will not be helpful.



(a) (b) (c) (d) (e) (f)

Fig. 1. Comparison between scanned and camera-captured document images. (a) A clear scan of a document. (b) An enlarged sub-image of (a). (c) The same document with curved shape captured by a projective camera. (d) An enlarged sub-image of (c) with similar content as (b). (e) Text line segmentation might be possible (locally) after rotating (d) so that text lines are roughly horizontal. (f) At character level, segmentation is still difficult even after local deskewing, and distorted characters are also difficult for OCR.

The key problem in document image rectification is to obtain the 3D shape of the page. In the literature, there have been three major approaches. The first assumes explicit 3D range data obtained through special equipments [6], [7], [8]; the second approach simplifies the problem by assuming flat pages [9], [7]; and the third approach assumes restricted shape and pose of the page [10], or additional metric information of markings on the page [11], [12]. There are also methods that only rely on 2D warping techniques [13]. However, because of the lack of 3D information, such methods are restricted to flat pages with small perspective distortion (see discussion in Sec. III-B.1).

In this paper, we present a rectification framework that extracts the 3D document shape from

a single 2D image and performs a shape-based geometric rectification to restore the frontal-flat view of the document. Fig. 2 illustrates the system level concept of our framework. The output image is comparable to scanned images and is significantly more OCR compatible than the input. Compared to previous approaches, our method does not need additional 3D/metric data, prior camera calibration, or special restrictions on document shape and pose. Our method borrows insights from previous work ([14], [9], [15], [16], [13]). One of our contributions is that our framework unifies the processing of planar and curved pages. Secondly, our approach removes the cylinder shape assumption or restricted pose assumption for curved pages, and as such allows most general cases of smoothly curved pages to be handled. These properties make our approach suitable to unconstrained mobile applications.

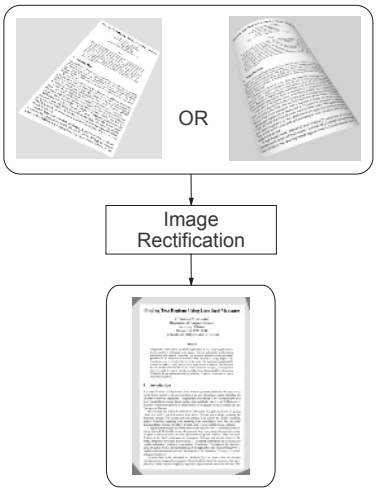


Fig. 2. High level illustration of geometric document image rectification.

Our framework requires three basic assumptions. First, the document page should contain sufficient printed text content. Second, the document is either flat or smoothly curved (i.e., not torn or creased). And third, the camera is a *standard* pin-hole camera in which the x-to-y sampling ratio is one and the principal point (where the optical axis intersects the image plane) is located at the image center. Most digital cameras satisfy this third assumption. Under these assumptions, we show that we can constrain the physical page by a developable surface model, obtain a planar-strip approximation of the surface using texture flow data extracted from the text in the image, and use the 3D shape information to restore the frontal-flat document view.

II. Background and Related Work

A. Document Capture Without Rectification

In the industry, cameras have been used for a long time in projects such as digitizing library collections, where precious books cannot be disassembled. Special care is taken to keep the pages as flat as possible. For example, thick books are only half opened. In the desktop environment, fixed overhead cameras [17] or mouse-mounted cameras [18] are used to replace scanners. In all these cases, the document is assumed flat and the hardware is set up so that there is no perspective distortion. However, this is difficult to achieve in mobile applications.

B. Document Capture With Rectification

In cases where non-planar shape and perspective projection cause distortion in camera-captured document images, geometric rectification is necessary before other document analysis algorithms can be applied. The key issue involved in rectification is to obtain the 3D information of the document page. A direct approach is to measure the 3D shape using special equipment such as structured light projector [6], [7], or stereo vision techniques with camera calibration [8]. Another approach requires the measurement of 2D metric data about the document page to infer the 3D shape [11], [12]. The dependence on additional equipment or prior metric knowledge prevents these approaches from being used in an unconstrained mobile environment.

Under the assumption that document surfaces are planar, the rectification can be achieved using only 2D image data [7], [14], [9], [19]. Apparently, these methods cannot handle curved documents such as opened books. In [15], [10], opened books are rectified under the condition that the camera's optical axis is perpendicular to the book spine.

Because of the difficulty involved in estimating 3D shape from 2D images, there is also work on rectification using pure 2D warping techniques to restore the linearity of text lines that appear curved in captured images [13]. These methods usually demand carefully positioning of the document with respect to the camera. For example, [13] describes a system for scanning thick books in which the scanning camera must have its optical axis orthogonal to the flat portion of the document page.

C. Shape Estimation From Images

There are many shape-from-X techniques that extract 3D shape from 2D images. However, we find that generally they are not appropriate for our task. Structure-from-motion is excluded because, in this paper, we assume a single image as the input. Shape-from-shading is excluded because it requires strong knowledge of lighting which is unknown in most cases. Shape-from-texture is a possible solution since printed text presents a regular pattern. However, traditional texture *gradient* analysis ([20], [21]) may be less accurate with document images since it is difficult to define textons in text. Shape-from-contour utilizes the symmetry or other metric information of two-dimensional contours on the surface. In practice, metric data is usually absent, while symmetry is vulnerable to occlusion. Therefore, in general, most shape-from-X techniques can provide a rough qualitative shape estimation but not accurate quantitative data to support rectification of document images.

D. Physical Modeling of Curved Documents

The shape of a curved document belongs to a family of 2D surfaces called *developable* surfaces, as long as the document is not torn, creased, or deformed by a soak-and-dry process. In mathematical terms, developable surfaces are 2D manifolds that can be isometrically mapped to an Euclidean plane. In other words, developable surfaces can unroll to a plane without tearing or stretching. This developing process preserves intrinsic surface properties, such as arc length and angle between lines on the surface.

The developable surface model is used in [7], [6], [8] to fit the 3D range data of a curved document. In our work, we do not assume a priori 3D data. Instead, we use the developable surface model to constrain the 3D shape estimation process.

E. Texture Flow and Shape Perception

Psychological observations suggest that a texture pattern which exhibits local parallelism gives a viewer the perception of a continuous flow field [22], [23], which we call a *texture flow* field. A typical example is the pattern of a zebra's stripes. Through a projection process (performed by a camera or a human visual system), a 3D flow field projects to a 2D field on the image plane. Under some mild assumptions, 2D flow fields effectively reveal the underlying 3D surface shape [24].

In documents, there are two important clues that a human visual system can use to infer the shape. First, document pages form developable surfaces. Second, there exist two well-defined texture flow fields representing local text line and vertical character stroke directions, respectively. On a flat document, the two fields are individually parallel and mutually orthogonal everywhere. This property is preserved locally for curved documents under the developable surface model. Therefore, a human visual system can quickly grasp the local surface orientations using the texture flow fields and integrate them using the global surface model to obtain depth perception.

III. Approach

A. Overview

We propose a framework that rectifies the image of a generally curved document, from the analysis of a page shape model and texture flow properties. It requires that there is sufficient text in the view. Our workflow begins with detecting the text area and the texture flow fields (Sec. III-B.1), then distinguishes planar and curved pages by verifying the linearity property of texture flow fields (Sec. III-B.2). Next it uses geometric properties of planar surfaces (Sec. III-C) or curved developable surfaces to estimate the 3D shape of the page (Sec. III-D), and lastly uses the 3D shape to unwarp the image. The output image is a frontal view of the flat page, just as from a scan.

B. Preprocessing

1) 2D Texture Flow Detection in Document Images: The first step in our processing is text identification, which locates the text area in the image and binarizes the text. Our algorithm is a gradient-based method [25]. Text identification is a difficult problem in itself deserving further research [1]. Therefore we do not address its details in this paper.

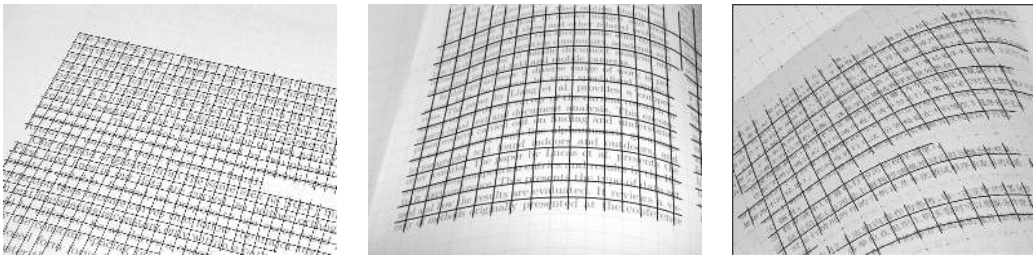
After text is found, we detect the local text line and vertical character stroke directions, which we define as the *major* and *minor* texture flows, respectively.

We formulate the major texture flow detection as a local skew detection problem. In document image analysis, skew detection finds the orientation of text lines with respect to the horizontal axis. In scanned documents, there is typically one global skew angle for the entire page. In camera-captured curved documents, the local skew angle varies across the entire image. Nevertheless, locally, it is roughly consistent. We apply the classic projection profile analysis

[26] to detect the skew angle in a small neighborhood. Then we use a relaxation labeling approach [27] to smooth out possible errors and obtain a coherent result [25].

We use directional filters to extract the linear structures of characters [28]. Vertical strokes are very common in text, thus the response of the filter usually exhibits a maximum when the filter's direction aligns with vertical strokes. Horizontal strokes also result in a maximum, but it can be detected and removed by comparing its direction to the major texture flow.

Fig. 3 shows estimated texture flows in real images. Note that in Fig. 3(b) the minor texture flow lines are straight and aligned with the cylinder generatrix, while in Fig. 3(c) both texture flow lines are curved, which represents the most general case. The synthetic image in Fig. 1(c) belongs to the latter case. For such cases, skipping 3D structure computation and using texture flows alone to rectify the images by 2D warping would result in an incorrect output. The rest of this paper describes our method of extracting 3D structure from texture flows and using it to process general curved document images. Nevertheless, in Sec. V we discuss simplifications for constrained cases (e.g., Fig. 3(b)).



(a) (b) (c)

Fig. 3. Texture flow results on real images. (a) A planar page, (b) an open book with cylindrical shape and (c) a page with non-cylindrical shape.

2) Surface Classification: Perspective projection preserves linearity, so straight text lines on planar documents remain straight in camera-captured images. Furthermore, these coplanar and parallel 3D lines share a common vanishing point in the image [29]. These two properties do not hold true for curved text lines on curved documents. Under perspective projection, while one curve lying on a *plane of sight* (a plane passing through the optical center) has a straight line as its projection in the image, multiple text lines on a curved document surface cannot simultaneously satisfy this requirement. Their projections cannot converge at a single point,

either. Therefore, we can determine whether the document is planar or curved by testing the linearity and convergence of text lines, which, in our case, can be verified using the major texture flow field. The minor texture flow field can be used, too.

Let $\{\mathbf{l}_i\}$ be a set of texture flow tangent lines (lines in the direction of the texture flow passing through any point) represented with the formalism of projective geometry. Under the planar page hypothesis, all these flow tangent lines $\{\mathbf{l}_i\}$ converge at a vanishing point, say $\mathbf{v}$ (in homogeneous representation, too), which can be written as

$$\mathbf{l}_i^\top \mathbf{v} = 0, \forall i.$$

This means that $\mathbf{v}$ lies in the null space of the sub-space spanned by $\{\mathbf{l}_i\}$; in other words, the rank of $\mathbf{L} = (\mathbf{l}_1, \ldots, \mathbf{l}_N)$ is less than three. By contrast, under the curved document hypothesis, $\mathbf{v}$ does not exist, which means that the null space of $\mathbf{L}$ is $\emptyset$ and $\mathbf{L}$ has full rank.

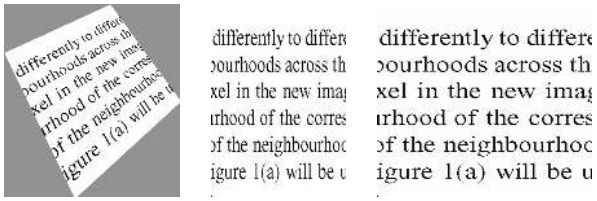
We use SVD decomposition to compute the eigenvalues of $\mathbf{L}$. Let S_1 and S_3 be the largest and smallest eigenvalues, respectively. We use S_3/S_1 as the convergence quality measure. If it rests below a predefined threshold, we decide that $\mathbf{L}$ does not have full rank.

C. Rectification of Planar Documents

This section covers the rectification of planar document images. From projective geometry, we know that the homography depends on the plane orientation $\mathbf{N}$ and camera focal length f. Together, they determine the position of the plane in the camera's coordinate system. In the following, we first deduce $\mathbf{N}$ and f from texture flow fields. Then we construct the homography, and show that at the end $\mathbf{N}$ and f do not need to be explicitly sought. Although $\mathbf{N}$ and f do not show up in the final formula, their estimates can help us to benchmark the precision of our method.

1) Page Plane Estimation: From the surface classification step described in the previous section, we obtain $\mathbf{v}_h$ and $\mathbf{v}_v$, the vanishing points of major and minor texture flow tangent lines. As [30] shows, a full metric rectification for a general projective transformation has five degrees of freedom. The line $\mathbf{l}_\infty$ connecting $\mathbf{v}_h$ and $\mathbf{v}_v$ is the vanishing line of the world plane, which involves two degrees of freedom and reduces the projective transformation to an affine transformation. The positions of the vanishing points in the world plane (the infinity points at North and East) allow us to remove the shearing and rotation from the affine transformation.

This leaves us with an unknown x-to-y aspect ratio[1] which cannot be determined using only the two vanishing points (see Fig. 4).



(a) (b) (c)

Fig. 4. Non-unique image rectification results. (a) A perspective distorted image. (b) and (c) are two possible rectification results that have different x-to-y aspect ratios. Both (b) and (c) are OCR compatible.

It is shown [30] that additional metric data, such as a length ratio or an angle (other than the right angle between the two texture flows) on the world plane can help solve for the last degree of freedom. We take a different approach; we assume that the principal point rests at the image center. This is usually true unless the image is cropped. Under this assumption, suppose that the two vanishing points are $\mathbf{v}_h = (x_h, y_h)^\top$ and $\mathbf{v}_v = (x_v, y_v)^\top$, then the 3D directions of the major and minor texture flows in the camera coordinate system are given by

$$\begin{aligned} \mathbf{V}_h &= (\mathbf{v}_h^\top, f)^\top, \\ \mathbf{V}_v &= (\mathbf{v}_v^\top, f)^\top, \end{aligned} \tag{1}$$

where f is the focal length. Due to their orthogonality in the 3D plane, i.e.,

$$\mathbf{V}_h^\top \mathbf{V}_v = 0, \tag{2}$$

it follows that

$$f = \sqrt{-\mathbf{v}_h^\top \mathbf{v}_v} = \sqrt{-(x_h x_v + y_h y_v)}, \tag{3}$$

if $\mathbf{v}_h^\top \mathbf{v}_v < 0$. When f is known, the plane normal $\mathbf{N}$ is also fixed by

$$\mathbf{N} = (\mathbf{V}_h \times \mathbf{V}_v)/|\mathbf{V}_h \times \mathbf{V}_v|,$$

[1]This is different from the x-to-y sampling ratio in the camera model, which we assume to be one.

Special care should be taken when either $\mathbf{v}_h$ or $\mathbf{v}_v$ lies at the infinity of the image plane. When a vanishing point lies at infinity, say $\mathbf{v}_h$, then the z-component of $\mathbf{V}_h$ is 0, regardless of f. Therefore Eq. 2 does not involve the focal length and we cannot solve for it. If both vanishing points lie at infinity, the document is simply parallel to the image plane and there is no perspective distortion. We need only rotate the image such that the two vanishing points map to the East and North. If only one vanishing point is at infinity, there is foreshortening along the direction of the other vanishing point. In this case, we are back to the situation where we can remove the perspective distortion up to an unknown aspect ratio.

When either $\mathbf{v}_h$ or $\mathbf{v}_v$ lies near the infinity, due to the noise in texture flow detection, we may arrive at vanishing point positions that are theoretically impossible. It could be that $\mathbf{v}_h^\top \mathbf{v}_v > 0$; or at least one vanishing point lies at the infinity, but the directions of the two vanishing points are not orthogonal.

Note that these cases where we cannot solve for f happen most often when the camera's optical axis is nearly perpendicular to the document. In such cases, the rectification homography is underdetermined. Fortunately, for such camera configurations the error introduced by the uncertainty in the rectification homography is small.

2) Metric Rectification: When perspective foreshortening is strong and rectification is most needed, we can compute f and $\mathbf{N}$ and then compute the full metric homography in the following way:

Consider an arbitrary point, (x_0, y_0), in the image plane. In the camera's 3D coordinate system, its position is $(x_0, y_0, f)^\top$. It follows that the corresponding 3D point located in the plane of the document is $\mathbf{W} = d(x_0, y_0, f)^\top$, where d (> 0) is an unknown depth factor. Define unit vectors $\overline{\mathbf{V}}_h = \mathbf{V}_h/|\mathbf{V}_h|$ and $\overline{\mathbf{V}}_v = \mathbf{V}_v/|\mathbf{V}_v|$. Suppose that we set up a 2D coordinate system in the document plane so the x-axis is aligned with $\overline{\mathbf{V}}_h$ while the y-axis is (must be) aligned with $\overline{\mathbf{V}}_v$. Every point on the document plane, thus, has a 2D coordinate. Assume that $\mathbf{W}$ has coordinate (x_0', y_0') on the page, then the 3D position of any point (x', y') on the page is

$$\begin{aligned}\mathbf{P} &= (x' - x_0')\overline{\mathbf{V}}_h + (y' - y_0')\overline{\mathbf{V}}_v + \mathbf{W} \\ &= \begin{pmatrix} \overline{\mathbf{V}}_h & \overline{\mathbf{V}}_v & \mathbf{W} \end{pmatrix} \begin{pmatrix} 1 & 0 & -x_0' \\ 0 & 1 & -y_0' \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}.\end{aligned}$$

A general projective camera model can be parameterized by a 3×3 upper triangular matrix $\mathbf{K}$ [29]. Most digital cameras have unit x-to-y ratio and zero shear. Also, the principal point offset is typically zero. Therefore, the matrix $\mathbf{K}$ simplifies to

$$\mathbf{K} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where f is the focal length. A 3D point $\mathbf{P} = (X, Y, Z)^\top$ in the camera's coordinate system projects to a point (x, y) in the image by

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \mathbf{K} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \tag{4}$$

where $x = u/w$, and $y = v/w$.

Overall, the homogeneous transformation from document plane to image plane is the concatenation

$$\mathbf{H} = \mathbf{K} \begin{pmatrix} \overline{\mathbf{V}}_h & \overline{\mathbf{V}}_v & \mathbf{W} \end{pmatrix} \begin{pmatrix} 1 & 0 & -x'_0 \\ 0 & 1 & -y'_0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5}$$

The inverse of $\mathbf{H}$ maps every point in the image plane back to the frontal-flat view of the document page and is called the *rectification* matrix. That is,

$$(x, y) \overset{\mathbf{H}^{-1}}{\rightarrow} (x', y'). \tag{6}$$

In Eq. 5, d and (x'_0, y'_0) can take any value. The value of (x'_0, y'_0) determines the translation of the rectified image within the destination plane. This translation cannot be derived from the image itself, nor is it relevant from the viewpoint of rectification. The depth factor d determines the scale of the rectified image — the larger the depth, the larger the rectified image. Similarly, this depth factor cannot be determined using only the image. Additional metric information must be known to fix d.

In our implementation, we choose $d = 1$, $(x_0, y_0) = (0, 0)$, and $(x'_0, y'_0) = (0, 0)$. Therefore, $\mathbf{W} = (0, 0, f)\top$. Let $\mathbf{v}_h = (x_h, y_h)^\top$ and $\mathbf{v}_v = (x_v, y_v)^\top$. And let

$$\begin{aligned} \alpha &= 1/|\mathbf{V}_h| = 1/\sqrt{x_h^2 + y_h^2 - (x_h x_v + y_h y_v)} \\ \beta &= 1/|\mathbf{V}_v| = 1/\sqrt{x_v^2 + y_v^2 - (x_h x_v + y_h y_v)}, \end{aligned}$$

where we replaced f by its expression found in Eq. 3. Then, Eq. 5 becomes

$$\begin{aligned} \mathbf{H} &= \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha x_h & \beta x_v & 0 \\ \alpha y_h & \beta y_v & 0 \\ \alpha f & \beta f & f \end{pmatrix} \\ &= f \begin{pmatrix} \alpha x_h & \beta x_v & 0 \\ \alpha y_h & \beta y_v & 0 \\ \alpha & \beta & 1 \end{pmatrix}. \end{aligned} \tag{7}$$

Computing the inverse of $\mathbf{H}$ allows us to compute the component x' and y' of the rectification mapping described by Eq. 6.

$$\begin{aligned} x' &= \alpha \frac{y_v x - x_v y}{(y_h - y_v)x + (x_v - x_h)y + (x_h y_v - x_v y_h)}, \\ y' &= \beta \frac{x_h y - y_h x}{(y_h - y_v)x + (x_v - x_h)y + (x_h y_v - x_v y_h)}, \end{aligned} \tag{8}$$

which maps a point (x, y) in the input image to the point (x', y') in the rectified image. Since f and $\mathbf{N}$ do not appear in Eq. 8, we can rectify a planar document even if they are not available because of the reasons discussed in the previous section. In those cases, the metric rectification is partial.

Because we cannot determine from texture flow analysis alone whether $\mathbf{V}_h$ points toward left or right of the flat document and whether $\mathbf{V}_v$ points toward top or bottom,the rectified image may end up being flipped vertically, horizontally, or both. In the last case the page is simply rotated by $180°$, as if it was scanned upside down. There is no simple way to tell that a document is upside down (sophisticated training based methods do exist [31]). The flipping in the first two cases can be removed easily, though. We take any three non-collinear points in the input image and record their clockwise order. In the output image we find their clockwise order too. If the two orders are different, then we flip the image vertically (or horizontally).

Some rectified results for real images are shown in Fig. 5 and Fig. 6. Fig. 5 consists of examples where perspective is absent or weak so that full metric rectification is unnecessary or only partial metric rectification is available. Fig. 6 shows images of strong perspective and the documents restored with full metric homography.

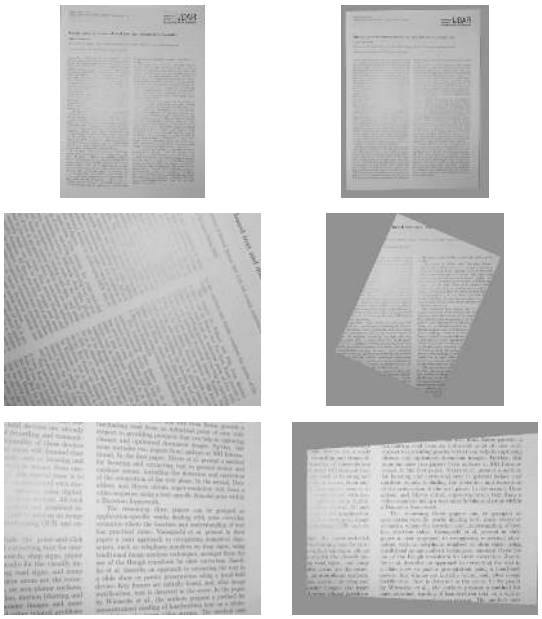


Fig. 5. Rectification results for real images with small perspective distortion.

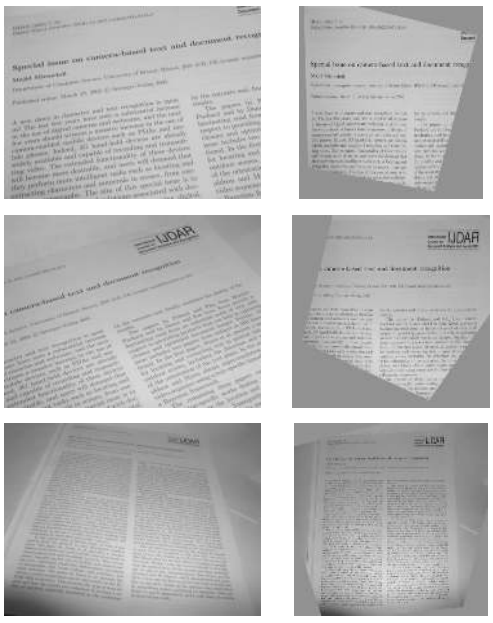


Fig. 6. Rectification results for real images with strong perspective distortion.

D. Rectification of Curved Documents

A curved document is more difficult to rectify than a planar page. Our approach is to decompose the curved surface to piecewise planar strips, based on the developable surface model.

1) Surface Modeling with Strip Approximation: A smoothly curved document can be modeled by a developable surface. Developable surfaces represent particular cases of a more general class of surfaces called *ruled* surfaces. Ruled surfaces are envelopes of a one-parameter family of straight lines (called *rulings*) in 3D space and each ruling lies entirely on the underlying surface. In other words, a ruled surface is the locus of a moving line in 3D space.

Developable surfaces are further constrained as they are envelopes of a one-parameter family of *planes*. As a result, all points along a ruling on a developable surface share one tangent plane. Given this property, we can approximate a developable surface with a finite number of planar strips that come from the family of tangent planes. More specifically, we divide a developable surface into pieces defined by a group of rulings. Each piece is approximated by a planar strip on the tangent plane along a ruling centered in this piece. Then, the de-warping of the developable surface can be achieved by rectifying planar strips piece by piece (see Fig. 7). As the number of planar strips increases, the approximation becomes more reliable, and the piecewise rectification becomes more accurate.

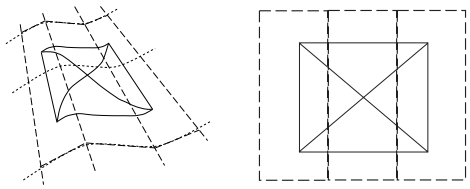


Fig. 7. Strip-based approximation to a developable surface

2) Projected Ruling Estimation: We call the projections of 3D rulings in the image *projected rulings*, or *2D rulings*. Similarly, we distinguish 2D texture flows and their 3D counterparts. In this section, we describe our method of detecting 2D rulings using 2D texture flow fields in document images.

Recall that all points along a ruling on a curved document share the same tangent plane. It follows that the 3D texture flow vectors at these points all lie in this tangent plane. Furthermore, all these 3D major (and minor) texture vectors are parallel. This claim becomes apparent once

we develop the document onto the tangent plane, in which process any vector on the tangent plane remains intact. In the developed document, the major texture flow vectors are obviously parallel, and so are the minor texture flow vectors. On the other hand, if major (and minor) texture flow vectors at all points along a 3D curve on a curved document are parallel, this curve *must* be a straight 3D ruling. In this case, the tangent planes at these points must be all parallel since their normals are the cross products of major and minor texture flow vectors. Because of the continuity of the 3D curve, these tangent planes collapse to just one. On a developable surface, this is possible only if the points are all on a ruling or the surface is a plane. Based on the above analysis, we have the following properties:

The 3D major and minor texture flow vectors along any 3D ruling on a developable document surface are parallel, respectively.

The 3D major and minor texture flow vectors along a non-ruling curve on a non-planar developable document surface cannot both be parallel.

As a result, under the perspective projection of a camera system, if a given line in the image is a 2D ruling, the 2D major (and minor) texture flow vectors along it converge at a common vanishing point (see Fig. 8). This vanishing point may be at infinity if the 3D flow vectors are parallel to the image plane. If these 2D major (or minor) texture flow vectors do not converge at a single point, this line is certainly not a projected ruling. Suppose we have a reference point, and parameterize any line through it by its angle θ. Let the convergence quality measure (defined in Sec. III-B.2) of the 2D vectors along this line be $c(\theta)$, then the optimal ruling estimate should minimize $c(\theta)$. We name $c(\theta)$ the *ruling quality measure* of the line in question.

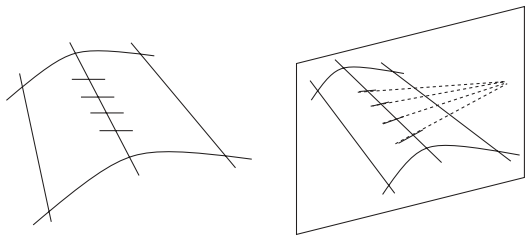


Fig. 8. 3D texture flow vectors along a 3D ruling are parallel and project to convergent 2D texture flow vectors along a 2D ruling. The 3D vectors do not have to be orthogonal to the 3D ruling.

Based on the analysis in the previous section, we need a finite number of rulings to divide

a developable surface into strips. For a group of 2D rulings, there is another global constraint. Through any point on a non-planar ruled surface there is one and only one ruling. This means that any two 3D rulings do not intersect. Consequently, the non-occluded parts of the two 2D rulings do not intersect, either. The only exception is the apex of a cone, which cannot appear inside the text area of a page, or the page would have a crease at the cone apex and would no longer be smooth.

We combine the individual and global constraints in computing a group of optimal 2D rulings. First we find the text area bounding box, and use the $c(\theta)$ measure to find an initial ruling estimate through the box center. Along the line orthogonal to this estimated ruling, we select N points. These points will serve as the reference points. Through each point we find an optimal ruling, and these rulings provide a division of the surface. Let the rulings be denoted as $\{r_i\}_{i=1}^N$, the angles between them be $\{\theta_i\}_{i=1}^N$, and the quality measure of each ruling be $c(\theta_i)$. The non-intersecting constraint is captured by

$$\Psi(\theta_i, \theta_j) = \begin{cases} \infty, & \text{if } r_i \text{ and } r_j \text{ intersect in text area,} \\ 0, & \text{otherwise, or if } i = j. \end{cases}$$

We define

$$Q(\{\theta_i\}) = \sum_{i=1}^{N} c(\theta_i) + \sum_{i,j=1}^{N} \Psi(\theta_i, \theta_j)$$

as the overall objective function of the group of 2D ruling candidates, which combines the local and global constraints. The optimal 2D rulings, then, are the ones that minimize Q. Because of the way we define Ψ, we actually are looking for a solution that minimizes the first term in Q while keeping the second term zero.

To solve this minimization problem, we first simplify the second term in Q by redefining

$$\begin{aligned} Q(\{\theta_i\}) &= \textstyle\sum_{i=1}^{N} c(\theta_i) + \sum_{i=1}^{N-1} \Psi(\theta_i, \theta_{i+1}) \\ &= \textstyle\sum_{i=1}^{N-1} [c(\theta_i) + \Psi(\theta_i, \theta_{i+1})] + c(\theta_N), \end{aligned}$$

i.e., we only require that the neighboring rulings do not intersect. Because the reference points are lined up sequentially, when no neighboring rulings intersect in the text area, the non-neighboring rulings cannot intersect, either. That is, $\sum_{i=1}^{N-1} \Psi(\theta_i, \theta_{i+1}) = 0$ implies that $\sum_{i,j=1}^{N} \Psi(\theta_i, \theta_j) = 0$. Hence our simplification does not change the minimization point in the solution space. Second, for each θ_i, we only consider a finite set of candidate values, or states. In other words, we

quantize the angles. Now we have a series of N points (or nodes) each with a number of candidate states, and the objective function Q is the sum of terms where each term only depends on the states of two subsequent nodes. This is a typical case where Q can be minimized using the classic dynamic programming method [32]. Fig. 9 shows two real images with the estimated 2D rulings overlaid.

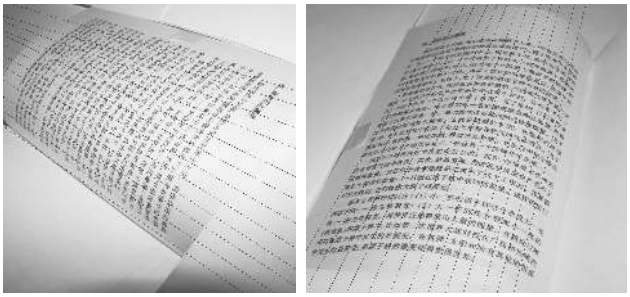


Fig. 9. Projected rulings detected in real images.

3) Vanishing Point Estimation for Rulings: Under perspective projection, a 3D line projects to a 2D line terminating at its *vanishing point* [29]. Given the position of the optical center, the direction of the 3D line is solely determined by the vanishing point, and vice versa. Similar to [9], our method of vanishing point detection is inspired by the following observation: Text lines are equally spaced on the page but, due to perspective, have varying distances in the image. The changes in text line spacing along a 2D ruling reveals the vanishing point of the ruling. In [9], Clark et al. implicitly use the margin of a justified paragraph (or the central line of a centered paragraph) as the ruling, apply projection profile analysis to find text line positions, and relate them to the vanishing point using two parameters. These two unknowns are solved by a search in a 2D parameter space. This method works only on a planar page consisting of a single justified or centered paragraph, and the search space is quite large.

Our method offers four improvements. First, we do not rely on justified or centered paragraphs to establish the 2D ruling. Second, we can handle multiple paragraphs with different text line spacing. Third, we address curved pages with *curve-based* projection profile (CBPP) analysis. And fourth, we simplify the computation to a one-parameter linear system which provides a stable, fast, and closed form solution.

Fig. 10 shows an example of finding intersections of text lines with a 2D ruling using CBPP

analysis. The peaks in the binarized profile (Fig. 10(c)) indicate the text line positions. Suppose **r** and **R** are the 2D and 3D rulings, respectively (see Fig. 11). Let $\{p_i\}_{i=1}^{M}$ and $\{P_i\}_{i=1}^{M}$ be the text line positions along **r** and **R**, where M is the number of text lines. The actual values of $\{p_i\}$ and $\{P_i\}$ are not important since only the line spacings are used. Within a paragraph, $\Delta = |P_{i+1} - P_i|$ is constant. By the invariant *cross-ratio* property [29], we have:

$$\begin{aligned} \frac{|p_{i+1}-p_i||p_{i+3}-p_{i+2}|}{|p_{i+2}-p_i||p_{i+3}-p_{i+1}|} &= \frac{|P_{i+1}-P_i||P_{i+3}-P_{i+2}|}{|P_{i+2}-P_i||P_{i+3}-P_{i+1}|} \\ &= \frac{\Delta\cdot\Delta}{2\Delta\cdot 2\Delta} = \frac{1}{4}, \forall i, \end{aligned} \tag{9}$$

if p_i through p_{i+3} come from the same paragraph. Otherwise, if Eq. 9 does not hold, then at least one gap between them is different, which divides two paragraphs.

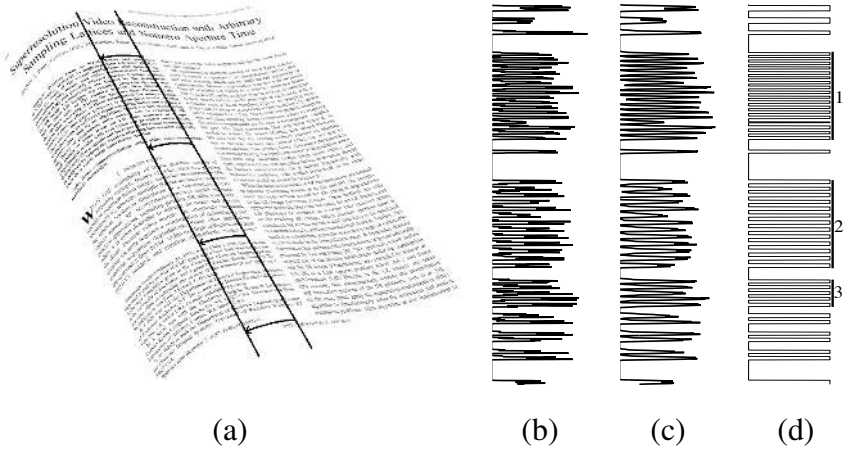


Fig. 10. Finding the intersections of text lines with 2D rulings. (a) Two nearby 2D rulings define the base lines of the projection, while the local text line directions define the curved projection path. (b) The curve-based projection profile. (c) Smoothed result of (b). (d) Binarized result of (c), in which three paragraphs are identified.

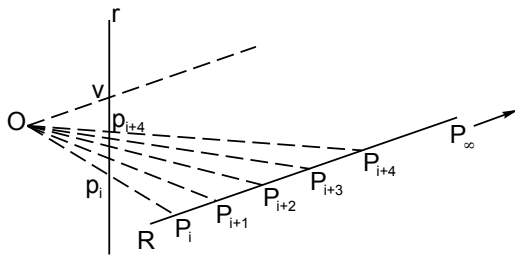


Fig. 11. Vanishing point **v** of a 2D ruling **r** corresponds to the point at infinity on the 3D ruling **R**.

If we let P_{i+3} converge toward ∞, then p_{i+3} converges toward v, which is the position of the vanishing point along r. In that case, Eq. 9 becomes

$$\begin{aligned} \frac{|p_{i+1}-p_i||v-p_{i+2}|}{|p_{i+2}-p_i||v-p_{i+1}|} &= \lim_{P_{i+3}\to\infty} \frac{|P_{i+1}-P_i||P_{i+3}-P_{i+2}|}{|P_{i+2}-P_i||P_{i+3}-P_{i+1}|} \\ &= \frac{1}{2}, \forall i, \end{aligned} \tag{10}$$

for any (p_i, p_{i+1}, p_{i+2}) in a paragraph. Eq. 10 represents a linear system in terms of v. With multiple text lines grouped into multiple paragraphs, we solve for optimal v in a Least Square sense.

4) Global Shape Optimization: Based on the planar strip approximation model, a curved document is divided into strips by the rulings. Ideally, each strip can be rectified independently using the method designed for planar documents. As a result, we obtain the surface normals to the strips and the camera focal length which fully describe the 3D shape of the document. However, such a result is usually noisy because each strip is small and does not contain sufficient information. Our solution is to globally constrain the strips by the properties of developable surfaces and printed text in documents.

Let us first define the variables used in this section (see Fig. 12). All points and vectors are defined in the camera's 3D coordinate system and consist of three components. All vectors are of unit length, unless otherwise noted. For any point $\mathbf{s}$ in the image plane, we use two vectors, $\mathbf{t_S}$ and $\mathbf{b_S}$, to represent the 2D major and minor texture flow directions. Across the document area, we have a group of M reference points, $\{\mathbf{p}_i\}_{i=1}^M$, and the estimated 2D rulings through them, whose directions are represented by a group of vectors, $\{\mathbf{r}_i\}_{i=1}^M$. The z component of either $\mathbf{s}$ or any $\mathbf{p}_i$ simply equals f, while the z components of vectors $\mathbf{t}$ and $\mathbf{b}$ are both equal to 0. On the 3D surface, the corresponding variables are denoted by upper case letters. The 3D surface normal to the planar strip between $\mathbf{R}_i$ and $\mathbf{R}_{i+1}$ is defined as $\mathbf{N}_i$. The 3D surface normal $\overline{\mathbf{N}}_i$ along $\mathbf{R}_i$ is then approximated by $\eta(\mathbf{N}_{i-1}+\mathbf{N}_i)$ where $\eta(\cdot)$ represents the normalization operation (i.e., $\eta(\mathbf{v}) = \mathbf{v}/|\mathbf{v}|$).

Except for surface normals, the other vectors on the 3D surface can be computed from their 2D projections using the following back-projection equations [24]:

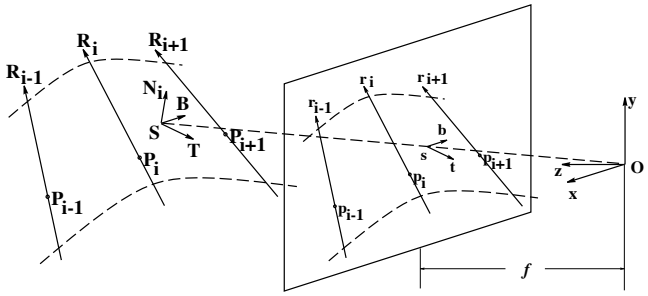


Fig. 12. Definitions of variables. **O** denotes the optical center, $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ represent the camera's coordinate system, and focal length f defines the distance between **O** and the image plane.

$$\begin{aligned} \mathbf{R}_i &= \eta((\mathbf{r}_i \times \mathbf{p}_i) \times \overline{\mathbf{N}}_i), \\ \mathbf{T}(\mathbf{s}) &= \eta((\mathbf{t_S} \times \mathbf{s}) \times \mathbf{N}_i), \\ \mathbf{B}(\mathbf{s}) &= \eta((\mathbf{b_S} \times \mathbf{s}) \times \mathbf{N}_i), \end{aligned} \tag{11}$$

where $\mathbf{s}$ is any point within the i^{th} planar strip (with $\mathbf{N}_i$ as its normal). Our global shape optimization process involves constraints expressed in terms of **N**, **R**, **T** and **B**. Through Eq. 11, these constraints are fundamentally functions of $\{\mathbf{N}_i\}$ and f.

The following four constraints are derived from the properties of developable surfaces and printed text in documents:

- *Orthogonality between surface normals and rulings:*

  When two rulings are very close, we have that the normal at any point on the surface between the two rulings are approximately orthogonal to either ruling, i.e., $\mathbf{N}_{i-1}^\top \mathbf{R}_i \approx \mathbf{N}_i^\top \mathbf{R}_i \approx 0$. Eq. 11 ensures that $\mathbf{R}_i^\top(\mathbf{N}_i - \mathbf{N}_{i-1}) = 0$, so we only need to check if $\mathbf{R}_i^\top(\mathbf{N}_i - \mathbf{N}_{i-1}) = 0$. We define $\mu_1 = \sum_{i=1}^{L-1}(\Delta\mathbf{N}_i^\top \mathbf{R}_i)^2$ as the measurement, where $\Delta\mathbf{N}_i = \mathbf{N}_i - \mathbf{N}_{i-1}$.
- *Parallelism of text lines within each strip:*

  Suppose that we select J sample points inside the i^{th} strip. The 3D text line directions at these points are denoted by $\{\mathbf{T}_{ij}\}_{j=1}^J$. We use $\mu_2 = \sum_i \sum_j (\mathbf{T}_{ij} - \overline{\mathbf{T}}_i)^2$ to measure their parallelism, where $\overline{\mathbf{T}}_i = (\sum_{j=1}^J \mathbf{T}_i)/J$.
- *Geodesic property of text lines:*

  Let the angle between $\overline{\mathbf{T}}_{i-1}$ and $\mathbf{R}_i$ be θ_i, and the angle between $\overline{\mathbf{T}}_i$ and $\mathbf{R}_i$ be γ_i. When we flatten the document strip by strip, the two angles must remain intact. On the flat

document, the text lines should be straight, which means $\theta_i + \gamma_i = \pi$, or, $\cos\theta_i + \cos\gamma_i = 0$, i.e., $\overline{\mathbf{T}}_{i-1}^{\top}\mathbf{R} + \overline{\mathbf{T}}_i^{\top}\mathbf{R} = 0$. Overall, this is measured by $\mu_3 = \sum_i((\overline{\mathbf{T}}_{i+1} - \overline{\mathbf{T}}_i)^{\top}\mathbf{R}_i)^2$.

- *Orthogonality between 3D major and minor texture flow fields:*

  This constraint is measured by $\mu_4 = \sum_i \sum_j (\mathbf{T}_{ij}^{\top}\mathbf{B}_{ij})^2$, where $\mathbf{B}_{ij}$ is defined similar to $\mathbf{T}_{ij}$.

Ideally, all the four constraint measurements should be zero.

We have two regularization terms that help us stabilize the solution:

- *Smoothness:*

  We use $\mu_5 = \sum_i (\Delta\mathbf{N}_i)^2$ to measure the surface smoothness. A large value indicates abrupt changes in normals to neighboring strips and 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,

$$F = \sum_{s=1}^{6} \alpha_s \mu_s,$$

where α_s is the weight representing the importance and our confidence in μ_s. Notice that each μ_s is the sum of terms where each term depends only on the normal to one strip, or the normals of two neighboring strips. This property is inherited by F. Based on Eq. 11, F is fundamentally determined by $\{\mathbf{N}_i\}$ and f. The optimization problem is to find $\{\mathbf{N}_i^*\}$ and f^* that minimize F.

The unit length constraint is actually a "hard" condition under which F should be optimized. It is incorporated into F so that the optimization becomes unconditional and easier to formulate. The smoothness weight depends on the curvature of the specific page. The other four factors, ideally, should reflect the accuracies of estimated features. For example, the accuracy of texture flow translates into the accuracy of $\mathbf{T}$ and $\mathbf{B}$, and thus affects the choice of α_4. In practice, however, both curvature and accuracies are unknown. Our decision is to assume the same accuracy for all estimated features and set all of α_1 through α_4 to 1. We set α_6 twice as large to emphasize the "hard" condition. As for α_5, we find that 10^{-2} works fine for our dataset.

In our experiments, we tested different weight values, changing each individual factor by as much as 30%. For certain inputs, slight changes in corresponding outputs were observed. However, our experiments show that on average the effect is not significant.

Good initial values of $\{\mathbf{N}_i\}$ and f are essential for solving the non-linear optimization problem. We obtain them with the help of vanishing points of rulings. First, we assume that f is known,

then the vanishing point of $\mathbf{r}_i$ determines the direction of $\mathbf{R}_i$, which eliminates one degree of freedom from $\mathbf{N}_i$ since $\mathbf{R}_i^\top \mathbf{N}_i = 0$. The remaining degree of freedom allows $\mathbf{N}_i$ to rotate in the plane orthogonal to $\mathbf{R}_i$. Our problem turns into finding the rotation angles.

Notice that we have a sequence of nodes $\{\mathbf{N}_i\}$, each of them having an unknown angle which we can quantize into a finite number of states, and the objective function is the sum of terms where each term depends only on the states of two subsequent nodes. Again, similar to the problem of finding a group of 2D rulings (see Sec. III-D.2), F is optimized using the dynamic programming method.

As for the focal length, we select a set of feasible values based on the physical lens constraint and perform the above process for each value. The f that results in minimum F is chosen as the best initial focal length.

Once we have the initial f and $\{\mathbf{N}_i\}$, we perform the non-linear optimization using a subspace trust region method based on the interior-reflective Newton method [33]. Fig. 13 presents the reconstructed surface normals and rulings corresponding to the right image in Fig. 9. The page shape is satisfactorily captured.

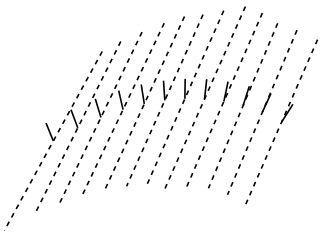


Fig. 13. Reconstructed surface normals (solid short lines) and rulings (dashed long lines) for the right image in Fig. 9

5) Piecewise Rectification: Given focal length, f, and its normal, $\mathbf{N}_i$, each strip can be rectified using Eq. 5 developed in Sec. III-C.2. The camera matrix $\mathbf{K}$ is determined by f, and the two axes, $\overline{\mathbf{V}}_h$ and $\overline{\mathbf{V}}_v$, are replaced by $\overline{\mathbf{T}}$ and $\overline{\mathbf{B}}$, which are computed using Eq. 11. We need to supply $\mathbf{W}$ and (x'_0, y'_0) to complete the computation. For the i^{th} strip, we rename (x'_0, y'_0) as (x'_i, y'_i) and choose $\mathbf{P}_i$ defined in Sec. III-D.3 as $\mathbf{W}$. The value of (x'_i, y'_i) controls the position of the i^{th} strip in the result image, and it should be such that neighboring strips are seamlessly connected. Once all strips are rectified, the flat document is obtained.

We start by setting an arbitrary depth, d_1, for $\mathbf{P}_1$. That is, $\mathbf{P}_1 = d_1 \mathbf{p}_1$, where $\mathbf{p}_1 = (x_1, y_1, f)^\top$

is the projection of $\mathbf{P}_1$ on the image plane and it is known. In our implementation, we choose $d_1 = 1$ and $(x'_1, y'_1) = (0, 0)$. These settings fulfill the requirement for computing $\mathbf{H}_1$ (defined in Eq. 5). Since we assume that both $\mathbf{P}_1$ and $\mathbf{P}_2$ are on the first strip, the point in the destination image where $\mathbf{p}_2$ maps to is given by $(x_2, y_2) \overset{\mathbf{H}_1^{-1}}{\rightarrow} (x'_2, y'_2)$. Also, we have $(\mathbf{P}_1 - \mathbf{P}_2)^\top \mathbf{N}_1 = 0$. Furthermore, if we write $\mathbf{P}_2 = (X_2, Y_2, Z_2)^\top$, we have

$$\begin{cases} x_2 &= fX_2/Z_2 \\ y_2 &= fY_2/Z_2 \end{cases}.$$

After some manipulation, we obtain

$$\begin{pmatrix} & \mathbf{N}_1^\top & \\ f & 0 & -x_2 \\ 0 & f & -y_2 \end{pmatrix} \mathbf{P}_2 = \begin{pmatrix} \mathbf{N}_1^\top \mathbf{P}_1 \\ 0 \\ 0 \end{pmatrix},$$

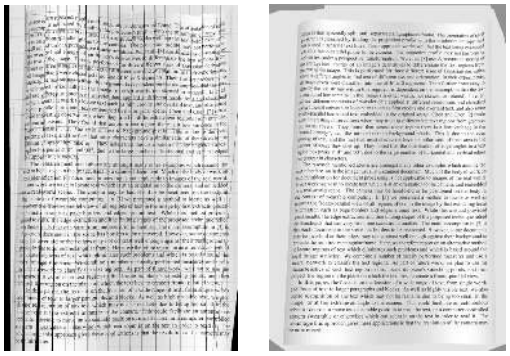
which gives us $\mathbf{P}_2$ based on f, $\mathbf{P}_1$, $\mathbf{N}_1$, and $\mathbf{p}_2$. In the same way, we can obtain $\mathbf{P}_i$ recursively from $\mathbf{P}_{i-1}$, $\mathbf{N}_{i-1}$ and $\mathbf{p}_i$, for any i.

Ideally, the rectified strips form the "mosaic" of the flat document. However, in practice, the mosaic is not seamless, due to the estimation noise at various steps. The problems include overlapping or gaps between neighboring strips (in order to keep text lines in both strips horizontal), and broken text line pieces not at the same horizontal level. The reason is that each strip is rectified with one homogeneous transformation consisting of only eight degrees of freedom. These eight parameters rectify the strip in an overall sense but are not sufficient to control the local behavior of the rectification. We address it with a local warping process. Essentially, we divide the strips into triangles and for each triangle we compute an affine transformation such that all the triangles in the original image map to seamless triangles in the destination image while keeping all text lines horizontal and straight. See Fig. 14 for a comparison before and after this warping process.

IV. Experiments and Evaluation

A. Example Results

We tested our system with both synthetic and real images. Fig. 15 compares the input and output. Overall, the rectified images are close to the frontal-flat view of the documents, despite some imperfection near text boundaries. Fig. 16 magnifies the top left and bottom left regions



(a) (b)

Fig. 14. Post-processing flattened strips to obtain seamless document image. (a) Piecewise rectification result with discontinuities between strips. (c) Triangle based warping result ensures a seamless flat document.

in the last input image in Fig. 15 and shows the corresponding output regions side by side. The zoomed regions show significant skew, and the character size in the bottom image is clearly larger than the top one because of foreshortening. Rectification removes skew and restores uniform character size. Even though rectification does not take care of the blur in the top image, the output image is much more readable than the input.

B. Evaluation Methodology

Since it is difficult to obtain ground truth 3D data for real images, we use synthetic images to quantitatively evaluate our algorithms. Synthetic images are generated using a module [28] that takes as input a flat document image, a shape model, a pose of the page with respect to the camera, a camera focal length, and outputs the perspective image along with ground truth data such as 2D texture flow fields, 2D rulings, vanishing points of rulings, and 3D surface normals. Our evaluation module automatically generates a set of synthetic images, compares the ground truth against the estimation, and summarizes the average errors. Furthermore, we use the OCR performance to measure the image quality from an application point of view. That is, we apply OCR to the original flat document, the synthetic curved document, and the rectified document. We take the OCR text of the flat document as ground truth, and use that to compute the OCR rates of the images before and after rectification.

For 2D texture flow fields, 2D rulings and 3D surface normals, which are vectors representing directions, we measure their precision by their direction errors which are angles. Such mea-

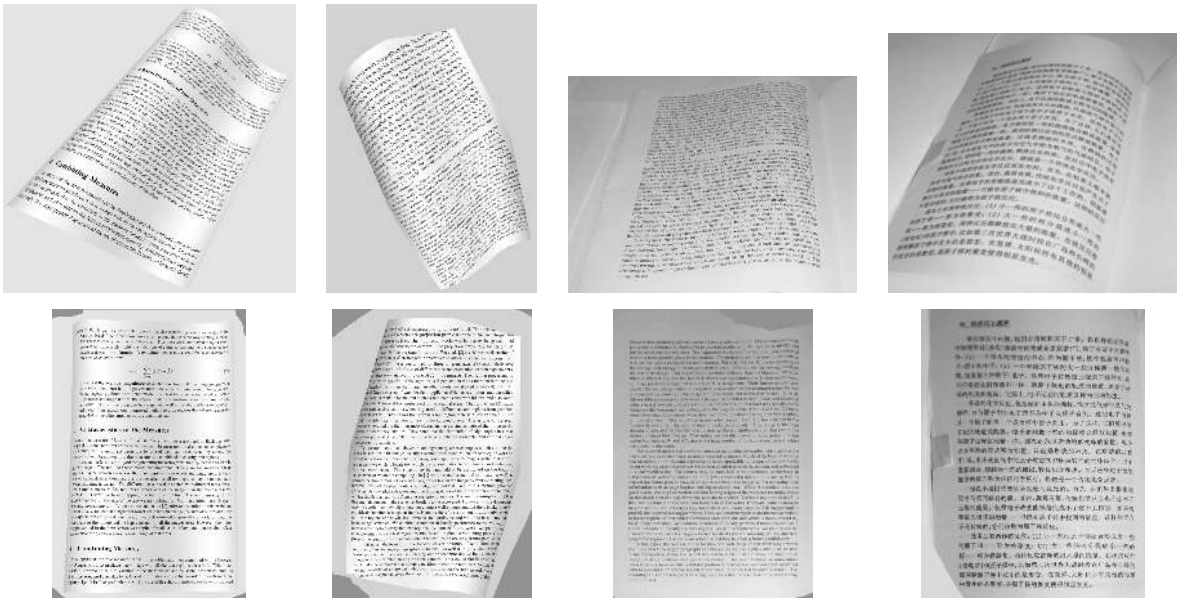


Fig. 15. Comparison of curved documents (top row) and rectification results (bottom row). In the top row, the left two inputs are synthetic images and the right two inputs are real images.

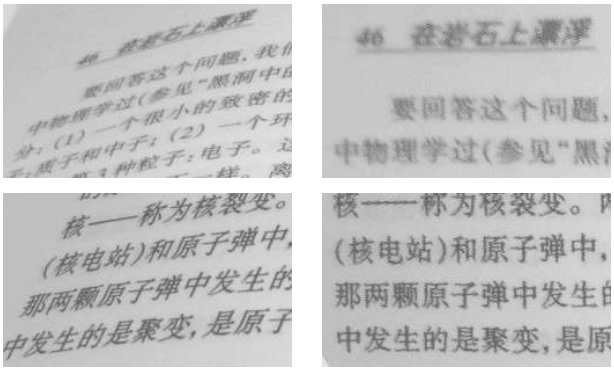


Fig. 16. Enlarged regions of original image and rectified result. These images are taken from the last pair of images in Fig. 15.

surements are independent of the image scale. For vanishing points of rulings or camera focal length, which are scalar numbers, a direct difference between the truth and estimate is, however, dependent on the image scale. Instead, we choose the following alternative benchmarks that are scale independent.

First, the precision of the vanishing point of a ruling can be equivalently measured by the precision of induced 3D ruling direction. This gives us an angle value. Since the 3D ruling

direction is also related to the position of the optical center, we assume perfect knowledge of the focal length at this step.

Second, we benchmark the focal length estimation in a similar way. We take a reference point (other than the principal point) in the image and compare two rays from this point to the optical centers given by the correct focal length and the estimated value, respectively, which provides an angle difference. In our test, we choose one image corner — any corner produces equivalent result if the principal point coincides with the image center — so the angle between the ray and the optical axis has the physical interpretation of being half of the *field of view*. By this interpretation, the error in field of view measures the focal length accuracy.

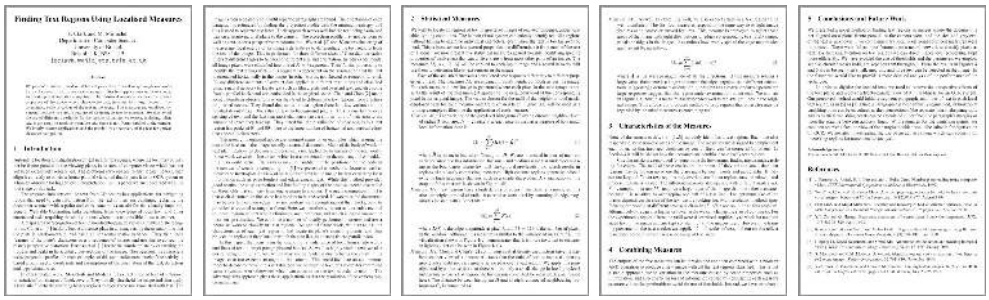
C. Evaluation Results

In the first experiment, we collected five clean document images at 300dpi. Their sizes are all 1600×2500 pixels. For each image, we created two other versions that have some parts cropped to test our algorithms' ability to handle occlusion. We designed four sets of pose parameters, including the rotation and translation of the document page in the camera's coordinate system, plus the camera focal length. The combination of five pages, three cropped versions (one without cropping), and four poses gives us 60 synthetic images of planar documents (see Fig. 17(a)(b)(c)). For curved pages, we designed two cylinder shapes (see Fig. 17(d)), which doubled the total number to 120. The first and last images in Fig. 17(d) represent typical opened books, while the other two images represent more general cases.

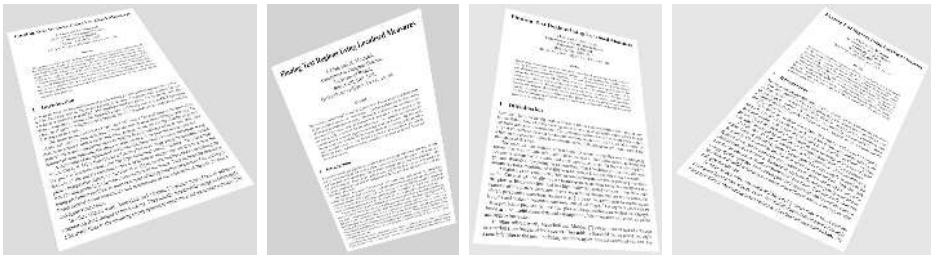
Due to the limit of space, we summarize the quantitative evaluation results in Table I. As for rectified images, not all of them are shown. The output images corresponding to the first and last input images in the last row in Fig. 17 are given in Fig. 15. For more rectified images, see Fig. 5, 6, 15, and 16.

The first half of Table I shows the evaluation on 2D and 3D features represented by angles in degree. The second half of the table compares the OCR performance before and after rectification. We used OmniPage Pro 12 for OCR. All the numbers are averages.

Overall, the accuracies of both 2D and 3D features are satisfactory. In particular, we obtain an encouraging accuracy of about 2.4 degrees in terms of 3D surface normals. For curved pages, the effect of global shape optimization on top of initial estimation is evident. Between curved and planar pages, although the accuracies of 2D texture flow fields (especially the major one) of

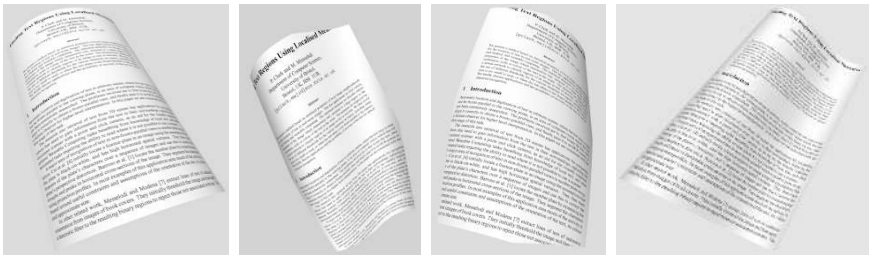


(a)



(b)

(c)



(d)

Fig. 17. Synthetic document image samples. From left to right (a) flat page no. 1 through no. 5, (b) pose no.1 through no. 4, (c) images with different croppings, (d) curved documents in which the first and the third come from one shape and the second and the fourth come from another shape; the second and the third are cropped.

curved pages is lower than that of planar pages, the difference between 3D feature accuracies is almost negligible. This demonstrates the robustness of our model based global shape optimization method.

We can draw two conclusions from the OCR comparison data. First, even for planar pages, the character and word recognition rates before rectification are below 30%. This means that even without curved shape, perspective distortion presents a significant obstacle by itself. Second, the image quality measured by OCR performance shows an improvement of about three to four folds after rectification. Although there is still room for further improvement, these rates are already acceptable in many document analysis applications such as indexing and retrieval.

Accuracy	Planar pages	Curved pages
Major texture flow	0.31°	0.80°
Minor texture flow	0.91°	1.12°
2D ruling	N/A	1.82°
3D ruling	N/A	2.91°
Field of view (initial)	N/A	3.21°
Surface normal (initial)	N/A	3.90°
Field of view (final)	3.30°	3.08°
Surface normal (final)	2.40°	2.44°
OCR character rates (original)	26.14%	23.05%
OCR character rates (rectified)	97.08%	87.64%
OCR word rates (original)	22.92%	14.29%
OCR word rates (rectified)	95.91%	83.83%

TABLE I

EVALUATION SUMMARY OF 2D/3D FEATURES AND OCR PERFORMANCE.

In the second experiment, we investigate our system's applicable range in terms of the curvature of the document shape and its pose relative to the camera. In the first step, we fix the pose parameter set and vary the shape parameter set. We design seven shape models that gradually change from almost flat to extremely curved (see Fig. 18(a)). Each shape is applied to five document pages. The 3D feature evaluation results are summarized in the first half of Table II. In the second step, we fix the shape and vary the pose. Again we design seven poses with gradually increasing tilt (Fig. 18(b)). The evaluation results are shown in the second half of

Table II. It is not surprising to see the accuracy drop as the curvature or tilt increases. The last shape and pose are rather challenging as illustrated by Fig. 18(c) which shows enlarged portions of the most curved and tilted images, respectively.

Shape	no.1	no.2	no.3	no.4	no.5	no.6	no.7
FOV	0.98°	1.40°	1.33°	1.23°	0.61°	1.43°	1.10°
N	1.06°	1.34°	1.57°	1.43°	1.86°	2.52°	4.65°
Pose	no.1	no.2	no.3	no.4	no.5	no.6	no.7
FOV	0.67°	1.62°	0.87°	0.92°	1.73°	4.15°	7.66°
N	1.55°	2.00°	1.66°	1.54°	1.56°	3.09°	3.78°

TABLE II

EFFECTS OF CURVATURE/POSE ON 3D SHAPE ESTIMATION. (FOV: FIELD OF VIEW; **N**: SURFACE NORMAL)

V. DISCUSSION

A. Implementation Variations

The rectification method described in Sec. III-D involves more complicated computation than the steps for planar pages (Sec. III-C). The additional complexity is necessary because we assume a general developable surface model and unconstrained camera position. When additional constraints are available, the framework can be tailored to reduce the complexity of the approach and improve its accuracy.

Opened books present a typical case of curved documents in real life. An opened book usually forms a cylinder shape, and the minor texture flow vectors are all parallel and coincide with the rulings of the surface. Under these conditions, the ruling detection step (Sec. III-D.2) is no longer needed. Furthermore, the vanishing point of rulings is simply the convergence point of minor texture flow vectors. Thus the step illustrated in Sec. III-D.3 is not necessary, either. Inside the global shape optimization step (Sec. III-D.4), multiple rulings $\{\mathbf{R}_i\}$ reduce to a single $\mathbf{R}$. Combined together, these simplifications can result in a more efficient and accurate system, provided that the input is indeed an opened book.

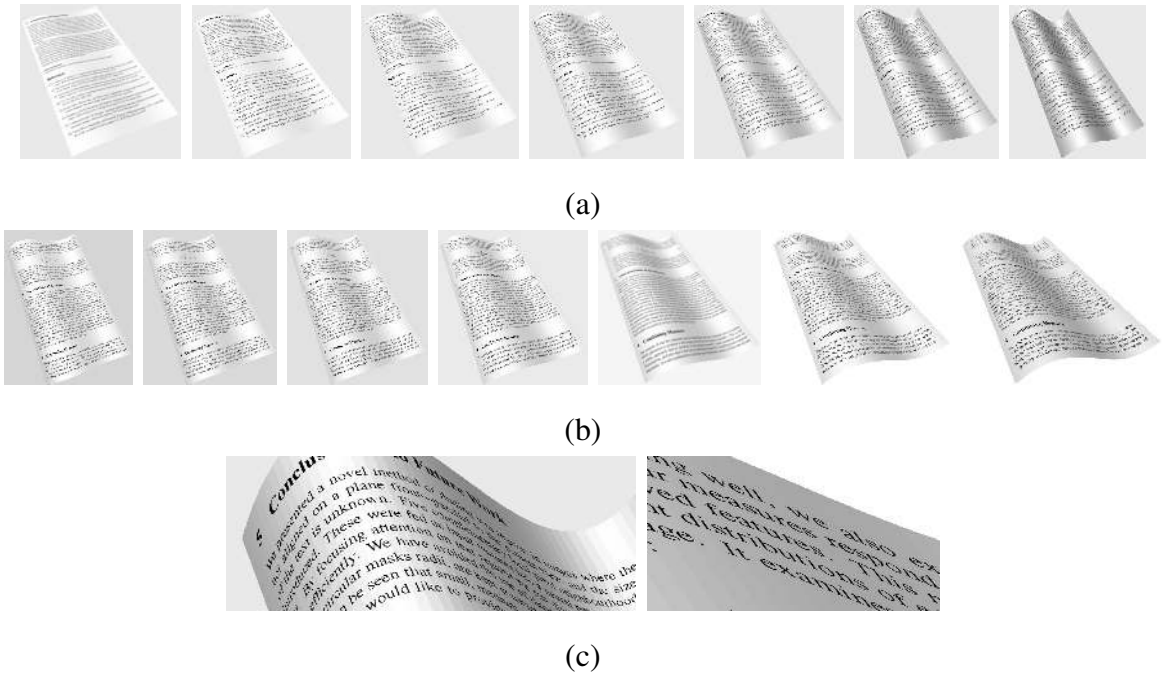


Fig. 18. Images used for testing the applicable range of the rectification system. From left to right: (a) seven shapes with increasing curvature (no. 1 through no. 7), (b) seven poses with increasing tilt (no. 1 through no. 7), and (c) enlarged details from the top part of the most curved page in (a) and most tilted page in (b).

Another even simpler customization is to implement only planar page rectification and shape classification modules. When the input is classified as non-planar, it is rejected; or the user can be prompted by an interactive interface to flatten the page and take another picture.

B. Parameter Selection

Beside the six weighting factors discussed in Sec. III-D.4, there are two other user defined parameters in our framework.

The first one is a shape threshold (see Sec. III-B.2) that classifies a document as planar or curved. It is preferable to set this parameter slightly in favor of a "planar" decision. Firstly, due to the inevitable noise in texture flow estimation, a threshold that is too tight will in practice label any document as "curved". Secondly, the ultimate risk of labeling a weakly curved page as planar is that text lines in the rectified image may still be weakly bent; however, when a nearly planar page is misclassified, the computation is much more expensive and the output is more prone to errors. In our implementation, we set this first threshold to 10^{-4}.

The second parameter controls the division of text lines into paragraphs (see Sec. III-D.3). Because the vanishing point computation (Eq. 10) requires at least three text lines in a paragraph, usually we tune the parameters in favor of fewer paragraph cuts and more text lines in each paragraph. In our experiments, the threshold is 10^{-2}.

C. Future Work Directions

There are several limitations in our current method and we would like to address them in the future. First, one of our basic assumptions is that the principal point is at the center of the image. While this is usually true for the entire camera-captured image, this does not hold if the image is cropped. To deal with this, we would need a method for estimating the position of the principal point. Second, currently our method only takes a single image as input. If multiple views are available, they provide complementary information that could improve the shape estimation accuracy. Third, our method does not rely on 3D range scanning or 2D metric data. However, when such information is available (e.g., from an inexpensive and low resolution IR camera attached to the optical camera), it is desirable to incorporate it into the computation.

VI. Conclusion

For camera-based document analysis, especially mobile applications, the distortion introduced by non-planar document surfaces and perspective projection is one of the critical challenges, if not the most important. We solve this problem with an automatic rectification approach which takes advantage of the developable surface constraint on curved pages and the properties of printed text in documents. Given a camera-captured image of a document, we estimate the 3D shape of the page as well as the camera's focal length based on texture flow fields extracted from the view, then restore the flat document image. With this method, a camera can emulate the function of a scanner and be used in various situations that scanners would be cumbersome or impractical. In experiments, we obtained significant improvement in OCR performance after rectification. The accuracy of shape estimation is also satisfactory, especially considering that we have only a single image without camera calibration. Our system can serve as a preprocessing unit in camera-based OCR applications, or the rectified images can be directly archived or presented for human viewing.

REFERENCES

[1] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: A survey," *International Journal on Document Analysis and Recognition*, vol. 7, no. 2+3, pp. 84–104, July 2005.

[2] M. J. Taylor, A. Zappala, W. M. Newman, and C. R. Dance, "Documents through cameras," *Image and Vision Computing*, vol. 17, no. 11, pp. 831–844, 1999.

[3] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1162–1173, Nov. 1993.

[4] G. Nagy, S. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals," *Computer*, vol. 25, no. 7, pp. 10–22, 1992.

[5] A. K. Jain and B. Yu, "Document representation and its application to page decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 294–308, 1998.

[6] M. S. Brown and W. B. Seales, "Image restoration of arbitrarily warped documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1295–1306, October 2004.

[7] S. Pollard and M. Pilu, "Building cameras for capturing documents," *International Journal on Document Analysis and Recognition*, vol. 7, no. 2+3, pp. 123–137, July 2005.

[8] A. Ulges, C. H. Lampert, and T. Breuel, "Document capture using stereo vision," in *Proceedings of the 2004 ACM Symposium on Document Engineering*, 2004, pp. 198–200.

[9] P. Clark and M. Mirmehdi, "On the recovery of oriented documents from single images," in *Proceedings of Advanced Concepts for Intelligent Vision Systems*, 2002, pp. 190–197.

[10] H. Cao, X. Ding, and C. Liu, "A cylindrical surface model to rectify the bound document image," in *Proceedings of the International Conference on Computer Vision*, vol. 1, 2003, p. 228.

[11] Y.-C. Tsoi and M. S. Brown, "Geometric and shading correction for images of printed materials a unified approach using boundary," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2004, pp. 240–246.

[12] N. Gumerov, A. Zandifar, R. Duraiswarni, and L. S. Davis, "Structure of applicable surfaces from single views," in *Proceedings of European Conference on Computer Vision*, 2004, pp. 482–496.

[13] Z. Zhang and C. L. Tan, "Correcting document image warping based on regression of curved text lines," in *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 1, 2003, pp. 589–593.

[14] P. Clark and M. Mirmehdi, "Estimating the orientation and recovery of text planes in a single image," in *Proceedings of the British Machine Vision Conference*, 2001, pp. 421–430.

[15] A. Ulges, C. H. Lampert, and T. M. Breuel, "Document image dewarping using robust estimation of curled text lines," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2005, pp. 1001–1005.

[16] H. Cao, X. Ding, and C. Liu, "Rectifying the bound document image captured by the camera: A model based approach," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2003, pp. 71–75.

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

[18] T. Nakao, A. Kashitani, and A. Kaneyoshi, "Scanning a document with a small camera attached to a mouse," in *Proceedins of IEEE Workshop on Applications of Computer Vision*, 1998, pp. 63–68.

[19] G. K. Myers, R. C. Bolles, Q.-T. Luong, J. A. Herson, and H. B. Aradhye, "Rectification and recognition of text in 3-D scenes," *International Journal on Document Analysis and Recognition*, vol. 7, no. 2+3, pp. 147–158, July 2005.

[20] J. Malik and R. Rosenholtz, "Computing local surface orientation and shape from texture for curved surfaces," *International Journal on Computer Vision*, vol. 23, no. 2, pp. 149–168, 1997.

[21] J. Gårding, "Shape from texture for smooth curved surfaces in perspective projection," *Journal of Mathematical Imaging and Vision*, vol. 2, pp. 327–350, 1992.

[22] O. Ben-Shahar and S. W. Zucker, "The perceptual organization of texture flow: A contextual inference approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 4, pp. 401–417, April 2003.

[23] A. R. Rao and R. C. Jain, "Computerized flow field analysis: Oriented texture fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 693–709, July 2003.

[24] D. C. Knill, "Contour into texture: Information content of surface contours and texture flow," *Journal of the Optical Society of America Association*, vol. 18, no. 1, pp. 12–35, Jan 2001.

[25] J. Liang, D. DeMenthon, and D. Doermann, "Flattening curved documents in images," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2005, pp. 338–345.

[26] D. X. Le, G. R. Thomas, and H. Weschler, "Automated page orientation and skew angle detection for binary document images," *Pattern Recognition*, vol. 27, no. 10, pp. 1325–1344, 1994.

[27] R. A. Hummel and S. W. Zucker, "On the foundations of relaxation labeling processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 267–287, 1983.

[28] J. Liang, "Processing camera-captured document images: Geometric rectification, mosaicing, and layout structure recognition," Ph.D. dissertation, University of Maryland, College Park, 2006.

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

[30] D. Liebowitz and A. Zisserman, "Metric rectification for perspective images of planes," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 1998, pp. 482–488.

[31] A. Vailaya, H. Zhang, C. Yang, F.-I. Liu, and A. Jain, "Automatic image orientation detection," *IEEE Transactions on Image Processing*, vol. 11, no. 7, pp. 746–755, 2002.

[32] D. B. Wagner, "Dynamic programming," *The Mathematica Journal*, vol. 5, no. 4, pp. 42–51, 1995.

[33] T. Coleman and Y. Li, "An interior trust region approach for nonlinear minimization subject to bounds," *SIAM Journal on Optimization*, vol. 6, pp. 418–445, 1996.