Flattening Curved Documents in Images

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

Abstract

Compared to scanned images, document pictures captured by camera can suffer from distortions due to perspective and page warping. It is necessary to restore a frontal planar view of the page before other OCR techniques can be applied. In this paper we describe a novel approach for flattening a curved document in a single picture captured by an uncalibrated camera. To our knowledge this is the first reported method able to process general curved documents in images without camera calibration. We propose to model the page surface by a developable surface, and exploit the properties (parallelism and equal line spacing) of the printed textual content on the page to recover the surface shape. Experiments show that the output images are much more OCR friendly than the original ones. While our method is designed to work with any general developable surfaces, it can be adapted for typical special cases including planar pages, scans of thick books, and opened books.

1. Introduction

Digital cameras have proliferated rapidly in recent years due to their small size, ease of use, fast response, rich set of features, and dropping price. For the OCR community, they present an attractive alternative to scanners as imaging devices for capturing documents because of their flexibility. However, compared to digital scans, camera captured document images often suffer from many degradations both from intrinsic limits of the devices and because of the unconstrained external environment. Among many new challenges, one of the most important is the distortion due to perspective and curved pages. Current OCR techniques are designed to work with scans of flat 2D documents, and cannot handle distortions involving 3D factors.

One way of dealing with these 3D factors is to use special equipments such as structured light to measure the 3D range data of the document, and recover the 2D plane of the page [1, 12]. The requirement for costly equipment, however, makes these approaches unattractive.

The problem of recovering planar surface orientations from images has been addressed by many researchers inside the general framework of shape estimation [5, 7, 10], and applied to the removal of perspective in images of flat documents [3, 4, 11]. However, page warping adds a non-linear, non-parametric process on top of this, making it much more difficult to recover the 3D shape. As a way out, people add in more domain knowledge and constraints. For example, when scanning thick books, the portion near the book spine forms a cylinder shape [8], and results in curved text lines in the image. Zhang and Tan [16] estimate the cylinder shape from the varying shade in the image, assuming that flatbed scanners have a fixed light projection direction. In terms of camera captured document images, Cao et al. [2] use a parametrical approach to estimate the cylinder shape of an opened book. Their method relies on text lines formed by bottom-up clustering of connected components. Apart from the cylinder shape assumption, they also have a restriction on the pose that requires the image plane to be parallel to the generatrix of the page cylinder. Gumerov et al. [6] present a method for shape estimation from single views of developable surfaces. They do not require cylinder shapes and special poses. However, they require correspondences between closed contours in the image and in the unrolled page. They propose to use the rectilinear page boundaries or margins in document images as contours. This may not be applicable when part of the page is occluded.

Another way out is to bypass the shape estimation step, and come up with an approximate flat view of the page, with what we call *shape-free* methods. For scans of thick bound volumes, Zhang and Tan [15] have an-

other method for straightening curved text lines. They find text line curves by clustering connected components, and move the components to restore straight horizontal baselines. The shape is still unknown but image can be OCRed. Under the same cylinder shape and parallel view assumptions as Cao et. al have, Tsoi et al. [14] flatten images of opened books by a bilinear morphing operation which maps the curved page boundaries to a rectangle. Their method is also shape-free. Although shape-free methods are simpler, they can only deal with small distortions and can not be applied when shape and pose are arbitrary.

Our goal is to restore a frontal planar image of a warped document page from a single picture captured by an uncalibrated digital camera. Our method is based on two key observations: 1) a curved document page can be modeled by a developable surface, and 2) printed textual content on the page forms *texture flow* fields that provide strong constraints on the underlying surface shape [9]. More specifically, we extract two texture flow fields from the textual area in the projected image, which represent the local orientations of projected text lines and vertical character strokes respectively. The intrinsic parallelism of the texture flow vectors on the curved page is used to detect the projected rulings, and the equal text line spacing property on the page is used to compute the vanishing points of the surface rulings. Then a developable surface is fitted to the rulings and texture flow fields, and the surface is unrolled to generate the flat page image.

Printed textual content provides the most prominent and stable visual features in document images [3, 11, 2, 15]. In real applications, other visual cues are not as reliable. For example, shade may be biased by multiple light sources; contours and edges may be occluded. In term of the way of using printed textual content in images, our work differs from [15, 2] in that we do not rely on connected component analysis which may have difficulty with figures or tables. The mixture of text and non-text elements will also make traditional shape-from-texture techniques difficult to apply, while our texture flow based method can still work. Overall, compared to others' work, our method does not require a flat page, does not require 3D range data, does not require camera calibration, does not require special shapes or poses, and can be applied to arbitrary developable document pages.

The remainder of this paper is organized into five sections. Section 2 introduces developable surfaces and describes the texture flow fields generated by printed text on document pages. Section 3 focuses on texture flow field extraction. We describe the details of surface estimation in Section 4, and discuss the experimental
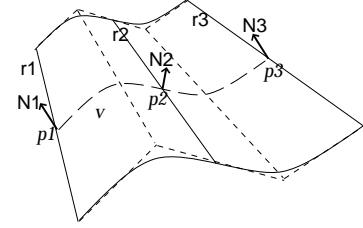

Figure 1. Strip approximation of a developable surface.

results in Section 5. Section 6 concludes the paper.

2. Problem Modeling

The shape of a smoothly rolled document page can be modeled by a developable surface. A developable surface can be mapped isometrically onto a Euclidean plane, or in plain English, can be unrolled onto a plane without tearing or stretching. This process is called *development*. Development does not change the intrinsic properties of the surface such as curve length or angle formed by curves.

Rulings play a very important role in defining developable surfaces. Through any point on the surface there is one and only one ruling, except for the degenerated case of a plane. Any two rulings do not intersect except for conic vertices. All points along a ruling share a common tangent plane. It is well known in elementary differential geometry that given sufficient differentiability a developable surface is either a plane, a cylinder, a cone, the collection of the tangents of a curve in space, or a composition of these types. On a cylindrical surface, all rulings are parallel; on a conic surface, all rulings intersect at the conic vertex; for the tangent surface case, the rulings are the tangent lines of the underlying space curve; only in the planar case are rulings not uniquely defined.

The fact that all points along a ruling of a developable surface share a common tangent plane to the surface leads to the result that the surface is the envelope of a one-parameter family of planes, which are its tangent planes. Therefore a developable surface can be piecewise approximated by planar strips that belong to the family of tangent planes (Fig. 1). Although this is only a first order approximation, it is sufficient for our application. The group of planar strips can be fully described by a set of reference points $\{P_i\}$ along a curve on the surface, and the surface normals $\{\mathbf{N}_i\}$ at these points.

Suppose that for every point on a developable surface we select a tangent vector; we say that the tangents are *parallel* with respect to the underlying surface

if when the surface is developed, all tangents are parallel in the 2D space. A developable surface covered by a uniformly distributed non-isotropic texture can result in the perception of a parallel tangent field. On document pages, the texture of printed textual content forms two parallel tangent fields: the first field follows the local text line orientation, and the second field follows the vertical character stroke orientation. Since the text line orientation is more prominent, we call the first field the *major* tangent field and the second the *minor* tangent field.

The two 3D tangent fields are projected to two 2D flow fields in camera captured images, which we call the *major* and *minor* texture flow fields, denoted as E_M and E_m. The 3D rulings on the surface are also projected to 2D lines on the image, which we call the 2D rulings or projected rulings.

The texture flow fields and 2D rulings are not directly visible. Section 3 introduces our method of extracting texture flow from textual regions of document images. The texture flow is used in Section 4 to derive projected rulings, find vanishing points of rulings, and estimate the page shape.

3. Texture Flow Computation

We are only interested in texture flow produced by printed textual content in the image, therefore we need to first detect the textual area and textual content. Among various text detection schemes proposed in the literature we adopt a simple one since this is not our focus in this work. We use an edge detector to find pixels with strong gradient, and apply an *open* operator to expand those pixels into textual area. Although simple, this method works well for document images with simple backgrounds. Then we use Niblack's adaptive thresholding [13] to get binary images of textual content (Fig. 2). The binarization does not have to be perfect, since we only use it to compute the texture flow fields, not for OCR.

The local texture flow direction can be viewed as a local skew direction. We divide the image into small blocks, and use projection profile analysis to compute the local skew at the center of each block. Instead of computing one skew angle, we compute several prominent skew angles as candidates. Initially their confidence values represent the energy of the corresponding projection profiles. A relaxation process follows to adjust confidences in such a way that the candidates that agree with neighbors get higher confidences. As a result, the local text line directions are found. The relaxation process is necessary because due to randomness in small image blocks, the text line orientations may
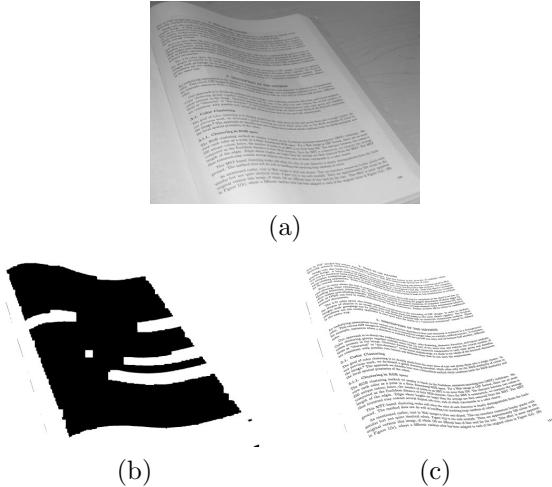

(a)

(b)   (c)

Figure 2. Text area detection and text binarization. (a) A document image captured by a camera. (b) Detected text area. (c) Binary text image.

not initially be the most prominent. We use interpolation and extrapolation to fill a dense texture flow field E_M that covers every pixel. Next, we remove the major texture flow directions from the local skew candidates, reset confidences for the remaining candidates, and apply the relaxation again. This time the results are the local vertical character stroke orientations. We compute a dense minor texture flow field E_m in the same way.

Fig. 3 shows the major and minor texture flow fields computed from a binarized text image. Notice that E_M is quite good in Fig. 3(c) even though two figures are embedded in the text. Overall, E_M is much more accurate than E_m.

4. Page Shape Estimation

4.1. Finding Projected Rulings

Consider a developable surface $\mathcal{D}$, a ruling R on $\mathcal{D}$, the tangent plane $\mathcal{T}$ at R, and a parallel tangent field $\mathbf{V}$ defined on $\mathcal{D}$. For a group of points $\{P_i\}$ along R, all the tangents $\{\mathbf{V}(P_i)\}$ at these points lie on $\mathcal{T}$, and are parallel. Suppose the camera projection maps $\{P_i\}$ to $\{p_i\}$, and $\{\mathbf{V}(P_i)\}$ to $\{\mathbf{v}(p_i)\}$. Then under orthographic projection, $\{\mathbf{v}(p_i)\}$ are parallel lines on the image plane; under spherical projection, $\{\mathbf{v}(p_i)\}$ all lie on great circles on the view sphere that intersect at two common points; and under perspective projection, $\{\mathbf{v}(p_i)\}$ are lines that share a common vanishing point. Therefore, theoretically if we have E_M or E_m, we can detect projected rulings by testing the texture flow orientations along a ruling candidate against the
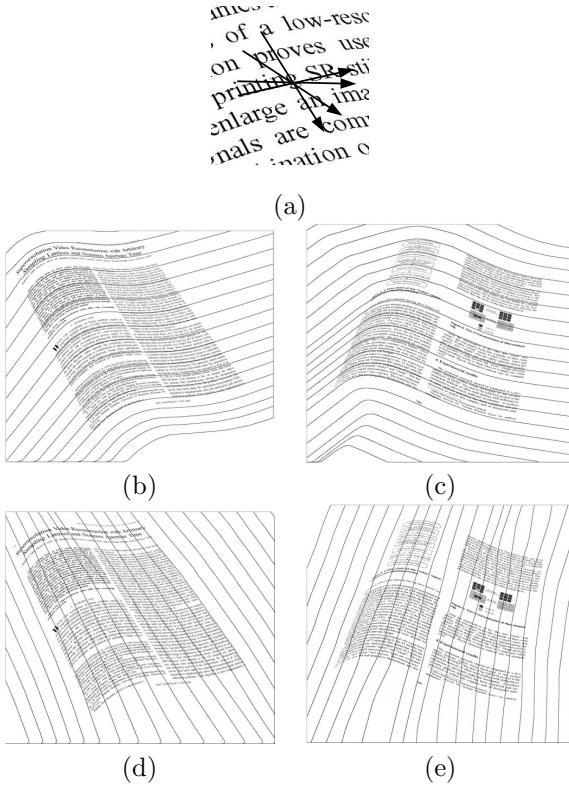
(a)

(b)   (c)

(d)   (e)

Figure 3. Texture flow detection. (a) The four local skew candidates in a small block. After relaxation the two middle candidates are eliminated. (b) (c) Visualization of major texture flow field. (d) (e) Visualization of minor texture flow field.

above principles.

However, due to the errors in estimated E_M and E_m, we have found that texture flow at individual pixels has too much noise for this direct method to work well. Instead, we propose to use small blocks of texture flow field to increase the robustness of ruling detection.

In a simplified case, consider the image of a cylindrical surface covered with a parallel tangent field under orthographic projection. Suppose we take two small patches of the same shape (this is possible for a cylinder surface) on the surface along a ruling. We can show that the two tangent sub-fields in the two patches project to two identical texture flow sub-fields in the image. This idea can be expanded to general developable surfaces and perspective projections, as locally a developable surface can be approximated by cylinder surfaces, and the projection can be approximated by orthographic projection. If the two patches are not taken along the same ruling, however, the above property will not hold. Therefore we have the following pseudo-code for detection of a 2D ruling that passes through a given point (x, y) (see Fig. 4):

1. For each ruling direction candidate $\theta \in [0, \pi)$ do the following

    (a) Fix the line $l(\theta, x, y)$ that passes through (x, y) and has angle θ with respect to the x-axis

    (b) Slide the center of a window along l at equal steps and collect the major texture flow field inside the window as a sub-field $\{E_i\}_{i=1}^n$, where n is the number of such sub-fields

    (c) The score of the candidate $l(\theta, x, y)$ is

    $$s(\theta) = \frac{\sum_{i=2}^n d(E_{i-1}, E_i)}{n} \qquad (1)$$

    where $d(E_{i-1}, E_i)$ measures the difference between two sub-fields, which in our implementation is the sum of squared differences

2. Output the θ that corresponds to the smallest $s(\theta)$ as ruling direction

We have found that the result has weak sensitivity to a large range of window sizes or moving steps.

To find a group of projected rulings that cover the whole text area, first a group of reference points are automatically selected, then for each point a projected ruling is computed. Because any two rulings do not intersect inside the 3D page, we have an additional restriction that two nearby projected rulings must not intersect inside the textual area.

As Fig. 4 shows, our ruling detection scheme works better in high curvature parts of the surface than in flat parts. One reason is that in flat parts the rulings are not uniquely defined. On the other hand, note that when the surface curvature is small, the shape recovery is not sensitive to the ruling detection result, so the reduced accuracy in ruling computation does not have severe adverse effects on the final result.

4.2. Computing Vanishing Points of Rulings

We compute the vanishing points of rulings based on the equal text line spacing property in documents. For printed text lines in a paragraph, the line spacing is usually fixed. When a 3D ruling intersects with these text lines, the intersections are equidistant in 3D space. Under perspective projection, if the 3D ruling is not parallel to the image plane, these intersections will project to non-equidistant points on the image, and the changes of distances can reveal the vanishing point position:

Let $\{P_i\}_{i=-\infty}^{\infty}$ be a set of points along a line in 3D space such that $|P_i P_{i+1}|$ is constant. A perspective projection maps P_i to p_i on the image plane. Then by the invariance of cross ratio we have
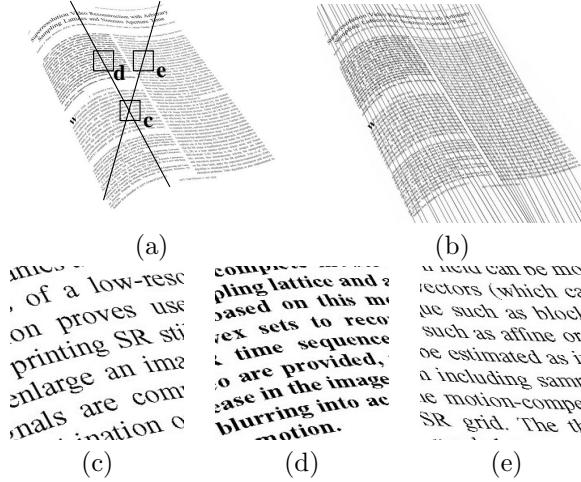
Figure 4. Projected ruling estimation. (a) Two projected ruling candidates and three image patches along the ruling candidates. (b) The estimated rulings. (c)(d)(e) Enlarged image patches. Notice that (c) and (d) have similar texture flow (but dissimilar texture) and are both different from (e).

$$\frac{|p_i p_j||p_k p_l|}{|p_i p_k||p_j p_l|} = \frac{|P_i P_j||P_k P_l|}{|P_i P_k||P_j P_l|}$$

$$= \frac{|i-j||k-l|}{|i-k||j-l|}, \forall i,j,k,l. \quad (2)$$

And as a result we have

$$\frac{|p_i p_{i+1}||p_{i+2} p_{i+3}|}{|p_i p_{i+2}||p_{i+1} p_{i+3}|} = \frac{1}{4}, \forall i, \quad (3)$$

and

$$\frac{|p_i p_{i+1}||p_{i+2} v|}{|p_i p_{i+2}||p_{i+1} v|} = \frac{1}{2}, \forall i, \quad (4)$$

where v is the vanishing point corresponding to p_∞ or $p_{-\infty}$.

We will come back to Eq. 4 and Eq. 3 after we describe how we find $\{p_i\}$. We use a modified projection profile analysis to find the intersections of a projected ruling and text lines. Usually a projection profile is built by projecting pixels in a fixed direction onto a base line, such that each bin of the profile is $\sum I(x,y : ax + by = 0)$. We call this a *linear projection profile*, which is suitable for straight text lines. When text lines are curved, we project pixels along the curve onto the base line (the projected ruling in our context), such that each bin is $\sum I(x,y : f(x,y) = 0)$ where f defines the curve. We call the result a *curve-based projection profile* (CBPP). The peaks of a CBPP corresponds to positions where text lines intersect the base line (assuming text pixels have intensity 1). Fig. 5
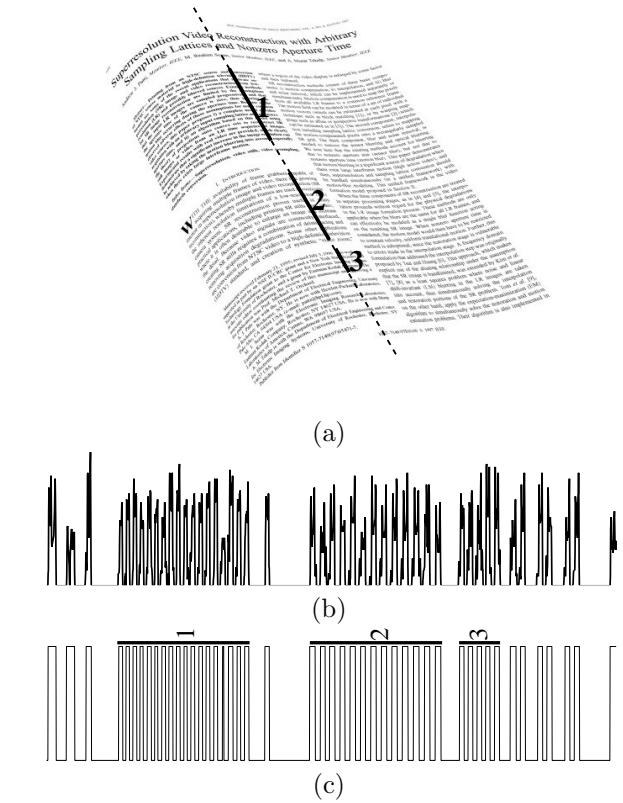

Figure 5. Computing the vanishing point of a 2D ruling. (a) A 2D ruling in the document image. (b) The curve based projection profile (CBPP) along the ruling in (a). (c) The smoothed and binarized CBPP with three text blocks identified. In each text block, the line spacing between top lines in the image (to the left in the profile graph) is smaller than that between lower lines (although this is not very visible to the eye). Such difference is due to perspective foreshortening and is explored to recover the vanishing point. In this particular case, the true vanishing point is $(-3083.70, 6225.06)$ and the estimated value is $(-3113, 5907)$ (both in pixel units).

shows how we identify the text line positions along a ruling.

The sequence of text line positions is clustered into K groups, $\{p_i^k\}_{k=1}^K$, such that each group $\{p_i^k\}_{i=1}^{n_k}$ satisfies Eq. 3 within an error threshold. The purpose of the clustering is to separate text paragraphs, and remove paragraphs that have less than three lines.

To find the best vanishing point v that satisfies Eq. 4 for every group of $\{p_i^k\}_{i=1}^{n_k}$, first we represent p_i^k by its 1D coordinate a_i^k along the ruling r (the origin can be any point on r). We write $a_i^k = b_i^k + e_i^k$, where e_i^k is the error term and b_i^k is the true but unknown position of text line.

Under the assumption that e_i^k follows a normal distribution, the best v should minimize the error function

$$\mathcal{E}(\{a_i^k\}, \{b_i^k\}) = \sum_{k=1}^{K} \sum_{i=1}^{n_k} (e_i^k)^2$$

$$= \sum_{k=1}^{K} \sum_{i=1}^{n_k} (a_i^k - b_i^k)^2, \qquad (5)$$

under the constraints that

$$0 = F_j^k(c, \{b_i^k\})$$

$$= \frac{(b_j^k - b_{j+1}^k)(b_{j+2}^k - c)}{(b_j^k - b_{j+2}^k)(b_{j+1}^k - c)} - \frac{1}{2}, \qquad (6)$$

for $k = 1, \ldots, K, j = 1, \ldots, n_k - 2$,

where c is the 1D coordinate of v on r. The best c can be solved by Lagrange method,

$$c = \underset{\forall c, \{b_i^k\}}{\operatorname{argmin}} \left\{ \mathcal{E}(\{a_i^k\}, \{b_i^k\}) + \sum_{k=1}^{K} \sum_{j=1}^{n_k} \lambda_j^k F_j^k(c, \{b_i^k\}) \right\}, \qquad (7)$$

where $\{\lambda_i^k\}$ are coefficients to be determined.

4.3. Developable Surface Fitting

The projected rulings $\{r_i\}_{i=1}^{n}$ divide the image of the document page into slices (Fig. 4(b)). Suppose that the optical center of the camera is O, we call the plane $\mathcal{S}_i$ that passes through O and r_i a *dividing plane*. The name comes from the fact that the pencil of planes $\{\mathcal{S}_i\}_{i=1}^{n}$ divides the page surface $\mathcal{D}$ into pieces. Each piece will be approximated by a planar strip $\mathcal{P}_i$, and the whole page is fitted by a group of planar strips. The development of $\mathcal{D}$ is accomplished by unfolding the set $\{\mathcal{P}_i\}_{i=1}^{n+1}$ strip by strip onto the plane. In the unfolded plane, we expect the text lines to be continuous, parallel, straight, and orthogonal to the vertical character stroke direction. Therefore we propose the following four principles that a group of strips $\{\mathcal{P}_i\}_{i=1}^{n+1}$ should satisfy in order to generate a flat image of the desired properties.

Continuity Two adjacent planar strips are continuous at the dividing plane. Formally, for (i=1,...,n), let L_i^- be the intersection line of $\mathcal{P}_i$ and $\mathcal{S}_i$, L_i^+ be the intersection line of $\mathcal{P}_{i+1}$ and $\mathcal{S}_i$, and α_i be the angle between L_i^- and L_i^+, then the *continuity* principle requires that $\alpha_i = 0$. The overall measure of continuity is defined as $M_c = \sum |\alpha_i|$.

Parallelism The texture flow directions inside a planar strip map to parallel directions on the unfolded plane. If we select M sample points in the image area that correspond to $\mathcal{P}_i$, and after unfolding, the texture flow directions at these points map to directions $\{\beta_{ij}\}_{j=1}^{M}$, the measure of parallelism is $M_p = \sum_i \operatorname{var}(\{\beta_{ij}\}_{j=1}^{M})$, where $\operatorname{var}(\cdot)$ is the sample variance function. Ideally M_p is zero.

Orthogonality The major and minor texture flow fields inside a planar strip map to orthogonal directions on the unfolded plane. If we select M sample points in the image area that corresponds to $\mathcal{P}_i$, and after unfolding, the major and minor texture flow vectors at these points map to unit-length vectors $\{\mathbf{t}_{ij}\}_{j=1}^{M}$ and $\{\mathbf{v}_{ij}\}_{j=1}^{M}$, then the measure of orthogonality is $M_o = \sum_i \sum_{j=1}^{M} |\mathbf{t}_{ij} \cdot \mathbf{v}_{ij}|$. M_o should be zero in the ideal case.

Smoothness The group of strips $\{\mathcal{P}_i\}_{i=1}^{n+1}$ is globally smooth, that is, the change between the normals of two adjacent planar strips is not abrupt. Suppose $\mathbf{N}_i$ is the normal to $\mathcal{P}_i$, the measure of smoothness is defined as $M_s = \sum_i |\mathbf{N}_{i-1} - 2\mathbf{N}_i + \mathbf{N}_{i+1}|$, which is essentially the sum of the magnitudes of the second derivatives of $\{\mathbf{N}_i\}$. The smoothness measure is introduced as a regularization term that keeps the surface away from weird configurations.

Ideally, the measures M_c, M_p, and M_o should all be zero. M_s would not be zero except for planar surfaces, but a smaller value is still preferred. In practice we want all of them to be as small as possible.

The four measures are determined by the focal length f_0 (which decides the position of the optical center O), the projected rulings $\{r_i\}$, and the normals $\{\mathbf{N}_i\}$, where the normals and the focal length are the unknowns. The following pseudo-code describes how we iteratively fit the planar strips to the developable surface.

1. Compute an initial estimate of $\mathbf{N}_i$ and f_0

2. For each of M_c, M_p, M_o and M_s (denoted as M_x):

   (a) Adjust f_0 to minimize M_x

   (b) Adjust each $\mathbf{N}_i$ to minimize M_x

3. Repeat 2 until M_c, M_p, M_o and M_s are below preset thresholds (currently selected through experiments and hopefully adjustable adaptively in future work)

The initial values of $\mathbf{N}_i$ and f_0 are computed in the following way:

For any point its normal is $\mathbf{N} = \mathbf{T} \times \mathbf{R}$, where $\mathbf{R}$ is the unit vector in the direction of a 3D ruling R, and $\mathbf{T}$ is any other unit vector on the tangent plane at that point. Since we have the vanishing point of R, $\mathbf{R}$ is determined by f_0 only. As for $\mathbf{T}$ we simply assume it to be parallel to the image plane. Therefore $\{\mathbf{N}_i\}$ are all determined by the unknown f_0. We do a one dimensional search within a predefined range of possible focal lengths to find the value that minimizes $\sum_x M_x$ as the initial f_0. And from f_0 we compute the initial $\{\mathbf{N}_i\}$.

5. Experiments and Discussion

We have applied our method to both synthetic and real images. The synthetic images are generated by warping a flat document image around a predefined developable surface. Fig. 6 demonstrates some of the preliminary results of our experiments. We choose to use the OCR recognition rate as evaluation measure, since OCR is one of the final applications and other applications like search and indexing also rely on OCR results. As the encouraging recognition rates show, the unwarped images are much more OCR friendly than the original ones. The size of both the warped and flattened images are in the range of 2000×2000 to 3000×3000 pixels. Subsequent experiments are being conducted to test the system's performance on large scale data, and the results will be reported in a future publication.

The reason that results obtained from synthetic images are better than from real images is because synthetic images have better quality in terms of focus, noise and contrast. Since different parts of the page are at different distances from the camera, not every part can be clearly focused. Also the uneven lighting creates additional difficulty for the OCR engine.

Our method can handle document pages that form general developable surfaces. If additional domain knowledge is available to constrain the family of the developable surfaces, it is possible to adapt our method to the following special cases that are very typical in practice.

- Planar documents under perspective projection

  The common convergence point of the major texture flow field E_M is the horizontal vanishing point of the plane, while E_m gives the vertical vanishing point. Once the two vanishing points are known, techniques similar to [3], [12], and [4] can be applied to remove the perspective.
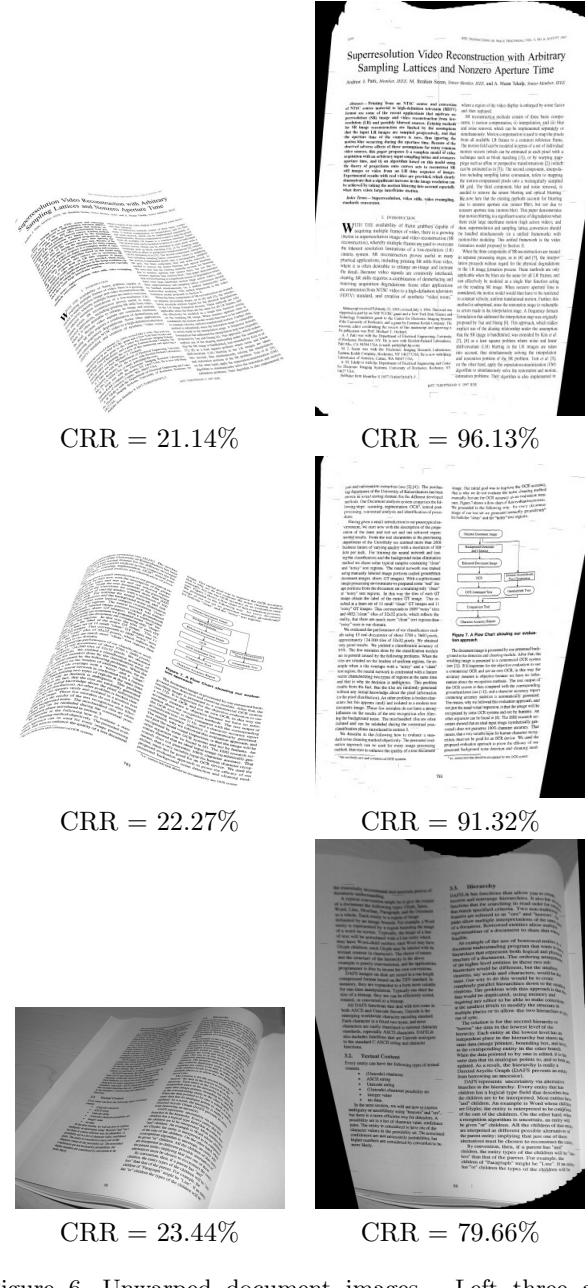
- Scanned images of thick bound books

CRR = 21.14%   CRR = 96.13%

CRR = 22.27%   CRR = 91.32%

CRR = 23.44%   CRR = 79.66%

Figure 6. Unwarped document images. Left three are warped images and right three are flattened images. CRR stands for 'Character Recognition Rate'. OCR results are from ScanSoft OmniPage.

In scans of thick bound volumes, the distorted area is not large enough to justify an expensive surface estimation process. Therefore the shape-free methods in [15] is appropriate. However, our method could do a better job of recovering the text line directions using the major texture flow field E_M when figures, tables, and other non-character connected components are present.

- Cylinder shaped pages of opened books

  In this case, the vertical character strokes coincide with the rulings, so we can find the vertical vanishing point V_v using both the minor texture flow field and the estimated projected rulings. By numerical integration we can find two contours that follow the major texture flow. It can be shown that the four intersections of any two lines passing through V_v and the two contours are coplanar and they are the four vertices of a rectangle. These four points lead to a horizontal vanishing point V_h, and together with V_v they determines f_0. The shape of the page is determined, too, once f_0 is known. The unrolling process would be the same as that for general developable surfaces.

6. Conclusion

In this paper we propose a method for flattening curved documents in camera captured pictures. We model a document page by a developable surface. Our contributions include a robust method of extracting texture flow fields from textual area, a novel approach for projected ruling detection and vanishing point computation, and the usage of texture flow fields and projected rulings to recover the surface. Our method is unique in that it does not require a calibrated camera. The proposed system is able to process images of document pages that form general developable surfaces, and can be adapted for typical degenerated cases: planar pages under perspective projection, scans of thick books, and cylinder shaped pages of opened books. To our knowledge this is the first reported method that can work on general warped document pages without camera calibration. The restoration of a frontal planar view of a warped document from a single picture is the first part of our planed work on camera-captured document processing. An interesting question that follows is the adaptation to multiple views. Also, the shape information can be used to enhance the text image, in particular to balance the uneven lighting and reduce the blur due to the lack of depth-of-field.

References

[1] M. S. Brown and W. B. Seales. Image restoration of arbitrarily warped documents. *IEEE Transactions on Pattern Analysis and Machine Intellegence*, 26(10):1295–1306, October 2004.

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

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

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

[5] J. M. Francos and H. H. Permuter. Parametric estimation of the orientation of textured planar surfaces. *IEEE Transactions on Image Processing*, 10(3):403–418, Mar 2001.

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

[7] W. L. Hwang, C.-S. Lu, and P.-C. Chung. Shape from texture: Estimation of planar surface orientation through the ridge surfaces of continuous wavelet transform. *IEEE Transactions on Image Processing*, 7(5):773–780, May 1998.

[8] T. Kanungo, R. M. Haralick, and I. Phillips. Global and local document degradation models. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 730–734, 1993.

[9] D. C. Knill. Contour into texture: Information content of surface contours and texture flow. *Journal of the Optical Society of America A*, 18(1):12–35, Jan 2001.

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

[11] M. Pilu. Extraction of illusory linear clues in perspectively skewed documents. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages 363–368, 2001.

[12] M. Pilu. Undoing paper curl distortion using applicable surfaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages 67–72, 2001.

[13] Ø. D. Trier and T. Taxt. Evaluation of binarization methods for document images. *IEEE Transactions on Pattern Analysis and Machine Intellegence*, 17(3):312–315, 1995.

[14] 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*, pages 240–246, 2004.

[15] 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*, volume 1, pages 589–593, 2003.

[16] Z. Zhang, C. L. Tan, and L. Fan. Restoration of curved document images through 3d shape modeling. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 10–15, 2004.