Rectification and 3D Reconstruction of Curved Document Images

Yuandong Tian and Srinivasa G. Narasimhan
The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
Email: {yuandong, srinivas}@cs.cmu.edu
Website: http://www.cs.cmu.edu/~ILIM

Abstract

Distortions in images of documents, such as the pages of books, adversely affect the performance of optical character recognition (OCR) systems. Removing such distortions requires the 3D deformation of the document that is often measured using special and precisely calibrated hardware (stereo, laser range scanning or structured light). In this paper, we introduce a new approach that automatically reconstructs the 3D shape and rectifies a deformed text document from a single image. We first estimate the 2D distortion grid in an image by exploiting the line structure and stroke statistics in text documents. This approach does not rely on more noise-sensitive operations such as image binarization and character segmentation. The regularity in the text pattern is used to constrain the 2D distortion grid to be a perspective projection of a 3D parallelogram mesh. Based on this constraint, we present a new shape-from-texture method that computes the 3D deformation up to a scale factor using SVD. Unlike previous work, this formulation imposes no restrictions on the shape (e.g., a developable surface). The estimated shape is then used to remove both geometric distortions and photometric (shading) effects in the image. We demonstrate our techniques on documents containing a variety of languages, fonts and sizes.

1. Introduction

Over the past thirty years, optical character recognition (OCR) technology has matured to achieve very accurate results. Using OCR, printed books can be digitized rapidly into electronic form that can be easier to store, retrieve and edit. However, the document images input to OCR are required to be taken without distortion, i.e., the document must be planar with text lines being horizontal and straight. Any distortion significantly reduces the accuracy of OCR.

Traditionally the image of the document is acquired using a flat-bed scanner. While this is perfect for a single sheet of paper, forcibly flattening books (especially if they are old and precious) is not desirable. In order to address this problem, several vision systems estimate the distortion and rectify the image of the document. Some systems rely on additional and precisely calibrated hardware such as stereo cameras [12, 15], laser scanners [1], or structured light projectors [3, 2] to measure the 3D deformation in the documents. While these systems have demonstrated accurate results, they are more expensive and less portable and hence have not found widespread application. Other systems aim to reduce distortion by analyzing a single captured image of the document. The idea is to infer the distortions from the changes in scale and orientation of text lines and the foreshortening of text fonts. While these systems are cheap and flexible, estimation of the 3D deformation and rectification reliably from a single image is a challenging task.

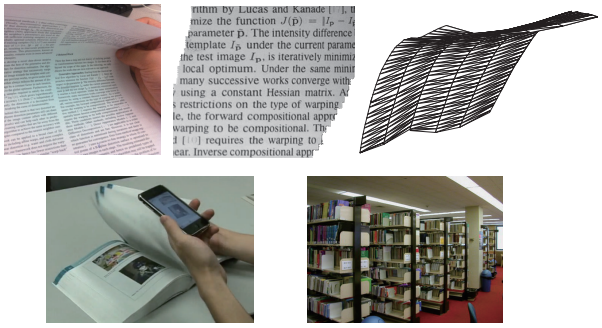


Figure 1. **First row:** From a single image of smoothly curved document as input, our methods compute the 3D shape of the document and a rectified image of text with no warping or shading effects. **Second row:** Two potential applications of our system: mobile text scanning and book digitization. (Images from Google).

In this paper, we follow the latter trend and build a vision system that reconstructs the 3D shape from a single image of curved document and rectifies the image (Fig. 1). We first estimate the 2D distortion (warping) grid in an image by exploiting the line structure and stroke statistics in text documents. This estimation consists of two main steps: text lines are automatically identified and densely traced, and the text orientation is determined at every location in the image. This approach does not rely on more noise sensitive operations such as image thresholding and character segmentation [4, 16], and does not rely on any *a priori* knowl-

edge of the font sizes, types or alphabet.

Unfortunately, knowing just the 2D distortion grid is not sufficient to rectify foreshortening and shading effects in the document. For this, we present a novel formulation of shape-from-texture to estimate the 3D deformation from the 2D distortion grid. In most documents, we observe that the 2D image grid can be regarded as a perspective projection of a *3D parallelogram mesh*. This observation allows us to solve an otherwise under-constrained reconstruction problem exactly using Singular Value Decomposition (SVD) (up to a global scale). Our reconstruction approach can be applied to general smooth surfaces and not restricted to simple parametric surfaces such as cylinders or developable surfaces as in [4, 18, 8, 9, 17]. Using the 3D shape, we present algorithms to unwarp the text document and remove shading effects under general and unknown lighting conditions.

Our system assumes that the image contains only text of the same font type and size. As shown in Fig. 1, our system has many applications. One example is mobile-OCR. Many smartphones have high-resolution cameras and can be used to image documents anywhere. Ideally, one just takes pictures of a note, notice, bulletin, receipt, or a book page and the application automatically converts the image to text. Another example is high-speed book scanning. In this scenario, a high-speed camera is used to record a book whose pages are being flipped through, and then the recorded video frames are rectified and assembled to obtain the textual content. This reduces the scanning time dramatically and can strongly impact several digital library initiatives.

2. Related work

Estimation of 2D document warping. Several approaches preprocess images using techniques such as binarization [4], connected component analysis [17, 7] or character segmentation [16] to estimate 2D warping. Previous line tracing techniques require pre-segmentation of each character [16], a global text line shape model (e.g. cubic spline [5]), or manual input of starting points of text lines [14]. These methods may miss many lines in the document. In contrast, we estimate the 2D distortions using domain knowledge in the form of the line structure and stroke statistics that is common to most text documents. Our self-similarity measure, scale estimation and line resampling steps do not miss lines, do not rely on thresholding and segmenting or identifying specific languages and fonts, and works even on low-resolution images.

3D reconstruction. To make 2D warp estimation more stable, many previous works assume a strong shape model, e.g., part of a cylinder [4], piecewise cylinder [18] or a developable surface [8, 9, 17]. In this paper, since the usage of domain knowledge leads to a better estimation of text warping, fewer assumptions are needed to reconstruct a broader class of 3D shapes. Most previous shape-from-texture works start by estimating the local differential quantities that the 3D shape projects onto the captured image, e.g. projected tangent angle [13], texture distortion [10] and foreshortening [6]. Since they all minimize non-linear objective functions, the estimation is not guaranteed to produce the global optimum [6]. In contrast, we formulate shape-from-texture in the specific context of text document images as a homogeneous least square problem, in which the globally optimal solution can be obtained using SVD.

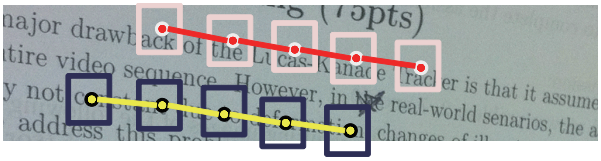


Figure 2. The self-similarity measure used for line tracing. Local patches extracted along the text line direction are correlated.

3. Estimation of document image warping

We define warping in a document image as a two dimensional coordinate system with one coordinate along the text line direction and the other across the text lines. For convenience, we call the former *horizontal* lines and the latter *vertical* directions. In this section, we present a series of steps to accurately trace and identify the text lines using a self-similarity measure that works for different sizes and types of fonts and different alphabets. Next, we estimate the vertical directions (or text orientation) by exploiting local stroke statistics in the text. Compared to previous works [4, 8, 9], our methods use explicit domain knowledge to better estimate the two dimensional warping in document images.

3.1. Horizontal text line detection

We begin by tracing an initial set of text lines, called *seed lines*, across the document image from randomly selected starting points, based on an image self-similarity measure. Then these seed lines are resampled and refined using dynamic programming. We describe each of the steps below.

3.1.1 Line tracing using self-similarity measure

Fig. 2 illustrates the concept of self-similarity measure: the patches extracted from a set of points along a text line are similar to each other in terms of an image metric such as normalized correlation. This property holds for different languages, font types/sizes, illumination and resolution of document images, and thus can be used for line tracing. Unlike the procedure in [14], our measure is invariant to the choice of the starting point for tracing lines.

But how do we determine the scale (or size) of the patch in self-similarity measure? For this, we study how the *mean gradient magnitude* (MGM) changes over image scale. We compute an image pyramid by successively downsampling

the original document image and for each level of the pyramid, we compute the MGM. We observe that the MGM initially increases during downsampling, since uniform 2D regions (inter-line whitespace) shrink more than 1D edges. However, the MGM starts to decrease at a scale where neighboring edges of letters/characters start to merge. This creates a peak as shown in Fig. 3(a). The location of the peak thus is directly related to the *characteristic scale* of the fonts in document images.

The text line tracing is done on the image downsampled to the characteristic scale. Starting from a random location x_0, we extract the patch centered at x_0, explore the patches at nearby locations $\{x_0 + (s\cos\theta_i, s\sin\theta_i)\}_{i=1}^{m}$ from x_0, and pick the one which is most similar to the current patch, measured by normalized correlation. Here s is the step and m is the number of angles to be explored. We repeat this process until the tracing trajectory reaches the boundary of the text region. We trace in both directions to cover the entire text region. The resulting lines are sorted from the top of the image to the bottom (Fig. 3(b)).

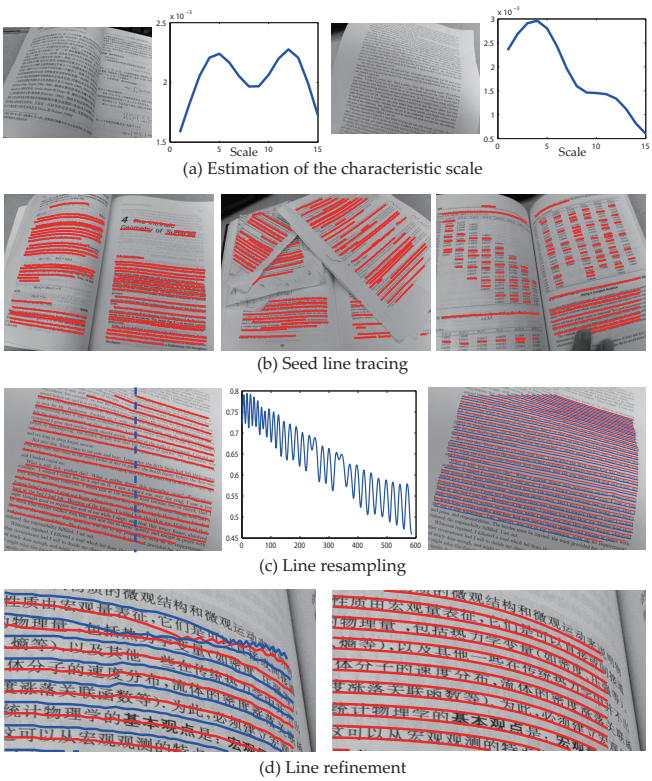
3.1.2 Resampling traced lines

Let $L \equiv \{l_1, l_2, \ldots, l_K\}$ be the *seed lines* traced and sorted as described above. Since the seed lines start from randomly selected points in the image, they typically skip text lines and may contain duplicate tracings for a single text line. A naive but inefficient approach would be to trace from every pixel of the image and pick a comprehensive set of text lines that cover the text region. Instead, using the fact that the directions of neighboring text lines are likely to be similar, we can interpolate the sparse set L to obtain a dense set $L' \equiv \{\tilde{l}_1, \tilde{l}_2, \ldots, \tilde{l}_{K'}\}$ where $K' > K$.

From this dense set L', our goal is to pick exactly one tracing for each text line and inter-line whitespace. For this, we consider the *mean pixel intensity* (MPI) computed on each interpolated tracing $\tilde{l}$. $MPI(\tilde{l})$ is low on dark text lines and high on whitespaces. Therefore, from the top to the bottom of the document image, the MPIs of L' depicts a sinusoidal profile, and the local extremes of this profile (i.e., $MPI(\tilde{l}_i) > MPI(\tilde{l}_{i\pm1})$ or $MPI(\tilde{l}_i) < MPI(\tilde{l}_{i\pm1})$) yield the desired set of tracings, one for each text line and one for inter-line whitespace.

3.1.3 Line refinement

Each of the above set of tracings passes near the center of the text line or inter-line white space. However, this is not accurate enough for estimation of warping and rectification. To refine these tracings, we first identify the top and bottom of every text line by interpolation (as show in the rightmost figure of Fig. 4). This is because they are easier to localize than the line centers. Then we maximize the following objective function for the interpolated



(a) Estimation of the characteristic scale

(b) Seed line tracing

(c) Line resampling

(d) Line refinement

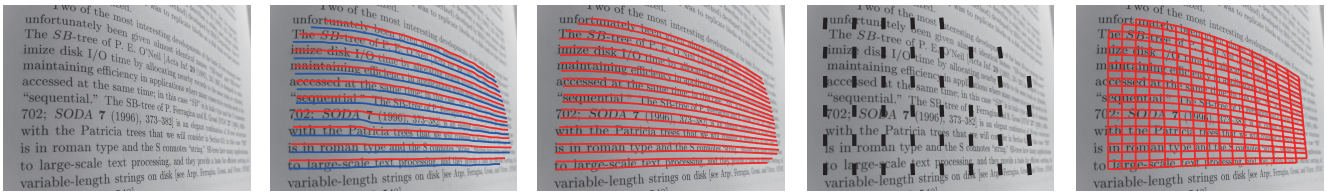
Figure 3. Workflow of horizontal text line tracing. **(a)** The mean gradient magnitude (MGM) on each level of the image pyramid, computed by successively downsampling the document image. The first peak of MGM can be used as a characteristic scale of the text. **(b)** Line tracings from random starting points on document images. The tracing performs well in both text regions and white spaces. **(c)** *Left:* A set of tracings are chosen, called "seed lines"; *Middle:* Mean pixel intensities computed along densely interpolated seed lines. The centers of text lines and white spaces correspond to the local extremes of the mean pixel intensities; *Right:* Then the top and bottom of the text lines (blue and red) are estimated, **(d)** and are refined by optimizing Eqn. 1.

top or bottom tracing that is represented by a set of points $l = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$:

$$E(\delta y_1, \delta y_2, \ldots, \delta y_n) = \sum_{i=1}^{n} \phi_i(\delta y_i) + \lambda \sum_{i=1}^{n-1} \psi_{i,i+1}(\delta y_i; \delta y_{i+1}) \quad (1)$$

where $\{\delta y_i\}$ are the vertical shifts of the point (x_i, y_i). The first term $\phi_i(\delta y_i)$ measures the log-likelihood of a shifted point $(x_i, y_i + \delta y_i)$ being at the true top or bottom boundary of the text line. The second term $\psi_{i,i+1}(\delta y_i, \delta y_{i+1})$ is a smoothness measure that penalizes sharp changes in the tangents of the tracing. The shifts δy_i are bounded by adjacent text lines in order to avoid intersection of tracings.

Although the objective function is nonlinear, it can be solved exactly using dynamic programming in linear time. As shown in Fig. 3(d), the result of the above steps is an accurate identification and tracing of horizontal text lines.



(a) Input image (b) Line tracing/resampling (c) Line refinement (d) Vertical estimation (e) Coordinate grid

Figure 4. Estimation of document image warping. **(a)** The original curved document image; **(b)** Horizontal text line tracing and resampling (Section 3.1.1-3.1.2); **(c)** Text line refinement (Section 3.1.3); **(d)** Estimation of vertical text orientation using local stroke statistics (Section 3.2); **(e)** The 2D coordinate grid of the image warp obtained using horizontal tracings and text orientation.

3.2. Text orientation estimation using local stroke statistics

The alphabet in many languages, such as English, Chinese and Hindi, contain vertical strokes (e.g., "b", "d", "k" and "l" in English). This property can be exploited to estimate the vertical direction at each location of the text region. The vertical text direction along with the horizontal line tracing of the previous section constitutes the coordinate grid for the warping of the deformed document.

As shown in [8, 9], the stroke statistics can be captured by locating the peaks in the edge orientation histogram of a local region. However, it is nontrivial to find the right scale of such local regions. Small scales have good localization but can be unstable due to other interfering strokes in the letters, whereas a large scale is more stable but with poorer localization. Instead of simply smoothing over local estimations [9], we provide a formulation robust to interfering strokes and achieves stable estimation even in small scales.

Let Ω be the set of all the image pixels and $m(\mathbf{x})$ and $\theta(\mathbf{x})$ be the gradient magnitude and orientation at pixel $\mathbf{x}$. We partition Ω into M *overlapping* local regions $\{R_1, R_2, \ldots, R_M\}$. Our goal is to find $A \subseteq \Omega$ that ideally contains only the vertical strokes in the image, so that within each region R_i, the gradient orientations of A are similar. Once A is obtained, vertical direction can be estimated stably even in small scales. Mathematically, we optimize the following objective:

$$\begin{aligned} J(A) &= \sum_{i=1}^{M} J_i(A_i) = \sum_{i=1}^{M} J_i(A \cap R_i) \\ J_i(A_i) &= \sum_{\mathbf{x}\in A_i} m(\mathbf{x})(\theta(\mathbf{x}) - \bar{\theta}_{A_i})^2 - \beta \sum_{\mathbf{x}\in A_i} m(\mathbf{x}) \end{aligned}$$

where $A_i = A \cap R_i$ and $\bar{\theta}_{A_i} = \frac{\sum_{\mathbf{x}\in A_i} m(\mathbf{x})\theta(\mathbf{x})}{\sum_{\mathbf{x}\in A_i} m(\mathbf{x})}$ is the weighted average of local gradient orientations. The first term of $J_i(A_i)$ penalizes the weighted variance of gradient orientations of A_i. The second term is a regularization term to avoid the trivial solution $A = \emptyset$.

To solve this intractable combinatorial optimization, we introduce intermediate variables θ_i (the local dominant orientations) for each region R_i and write $J(A)$ as minimization of the following function $J'(A; \{\theta_i\})$ over $\{\theta_i\}$:

$$J'(A; \{\theta_i\}) = \sum_{i=1}^{M} J'_i(A_i; \theta_i)$$

where $J'_i(A_i; \theta_i) = \sum_{\mathbf{x}\in A_i} m(\mathbf{x})(\theta(\mathbf{x}) - \theta_i)^2 - \beta \sum_{\mathbf{x}\in A_i} m(\mathbf{x})$. Obviously $\min_A J(A)$ is equivalent to $\min_A \min_{\{\theta_i\}} J'(A, \{\theta_i\})$. We obtain a solution by alternatively minimizing θ_i and A_i for each region while fixing the region $A - A_i$ (A excludes A_i) and other variables θ_j ($j \neq i$). For each region R_i, we initialize θ_i as the perpendicular direction to the estimated horizontal text lines. An example result is shown in Fig. 5. The entire workflow of estimating the two dimensional warping of the document image is summarized in Fig. 4. We emphasize that accurate warping estimation is crucial for 3D reconstruction.

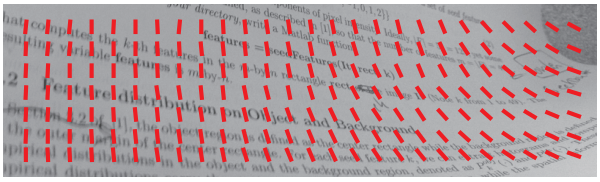


Figure 5. Example of text orientation estimation by Section 3.2.

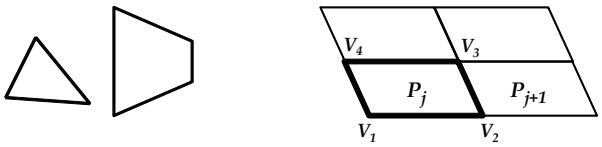


Figure 6. **Left:** Depth information can be extracted from a perspective projection of a 3D parallelogram using its foreshortened edges. This is impossible for a triangle. **Right:** We assume each grid cell is a parallelogram in 3D space, which gives 3 linear constraints: $\mathbf{V}_1 + \mathbf{V}_3 - \mathbf{V}_2 - \mathbf{V}_4 = 0$. With four unknowns, the parallelogram can be reconstructed up to a global scale. With more constraints than unknowns, estimating the depths of a grid with shared vertices is a well-defined problem.

4. Reconstruction from a single image

Using the 2D warping, we can make the text line horizontal and text orientation vertical. However, this is not

sufficient to rectify the document image due to the following two reasons. First, a pure geometric rectification cannot remove the shading on the images. Second, due to the depth variation, the foreshortening effects along the text lines cannot be correctly rectified as shown in Fig. 8(c). In this paper, we address these two problems by first estimating the 3D deformation of the curve document from only the 2D warping. Then the foreshortening effects can be rectified by using the depth variation along the text lines. The shading can be removed by computing surface normals of the 3D deformation and by assuming a reflectance model (e.g. Lambertian) for the document.

Without any assumptions, 3D reconstruction from a single image is an under-constrained problem with more unknown variables than constraints. In this work, we assume (1) the camera projection is perspective and (2) each cell of the 2D warping coordinate grid is a parallelogram in 3D space. The second assumption is reasonable because (a) the surface can be assumed to be locally planar or rigid if grid cells are sufficiently small, as demonstrated in recent work [11], and (b) for most *undistorted planar* documents, the text lines are parallel and so are local vertical text directions, thus forming a parallelogram grid.

But why not use a triangle mesh as in the work of Taylor et.al [11]? As shown in Fig. 6, the equilateral property in a parallelogram makes it possible to estimate its depth up to a global scale from a single perspective view. This is in contrast to [11] in which three camera views are required to reconstruct a 3D triangle up to a "flip" ambiguity.

We now formulate the problem of reconstructing a 3D parallelogram mesh from a 2D warping grid. Consider the illustration in Fig. 6. We denote the 3D coordinates of the i-th grid vertex as $\mathbf{V}_i = (X_i, Y_i, Z_i) = (x_i Z_i, y_i Z_i, Z_i)$, where (x_i, y_i) is its 2D coordinates. For simplicity, focal length is assumed to be 1 and center of projection is at the origin. Let $\{P_j\}_{j=1}^{N_p}$ denote the parallelograms where $P_{j,1:4}$ are the four vertices in counter-clockwise direction. The necessary and sufficient condition that the four vertices form a parallelogram is $\mathbf{\Delta}_{P_j} \equiv \mathbf{V}_{P_{j,1}} + \mathbf{V}_{P_{j,3}} - \mathbf{V}_{P_{j,2}} - \mathbf{V}_{P_{j,4}} = \mathbf{0}$. Thus we minimize the following objective:

$$Q(Z_1, Z_2, \ldots, Z_n) = \sum_{j=1}^{N_p} \|\mathbf{\Delta}_{P_j}\|^2 \qquad (2)$$

To avoid the trivial solution of $\mathbf{Z} \equiv [Z_1, Z_2, \ldots, Z_n] = \mathbf{0}$, we add a global scale constraint $\|\mathbf{Z}\| = 1$ and solve Eqn. 2 exactly using Singular Value Decomposition (SVD) up to a global scale factor. Each 3D parallelogram brings forward 3 linear constraints, making the problem well-constrained.

For robustness to noise in the 2D grid locations, we add the re-projection errors to relax Eqn. 2:

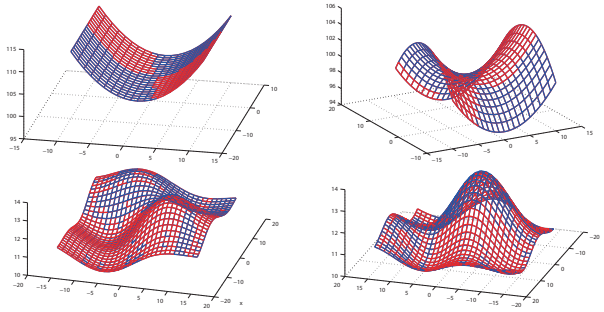


Figure 7. Example reconstructions of synthetic shapes. The ground truth shapes are shown in blue, while reconstructed shapes using Eqn. 3 are shown in red. Our method can reconstruct both ruled and non-ruled surfaces.

Noise level	0	0.001	0.005	0.01	0.05
Errors	0.0012	0.0014	0.0044	0.0085	0.0503

Table 1. Reconstruction errors of randomly generated synthetic 3D shapes under different noise levels. Gaussian noise is added to the 2D projections, with standard deviations as shown. The average side length of grid cells is 1. The relative root-mean-square errors between ground truth and reconstructed 3D shapes are averaged over 100 random shapes for each noise level. The low errors demonstrate the robustness of our approach.

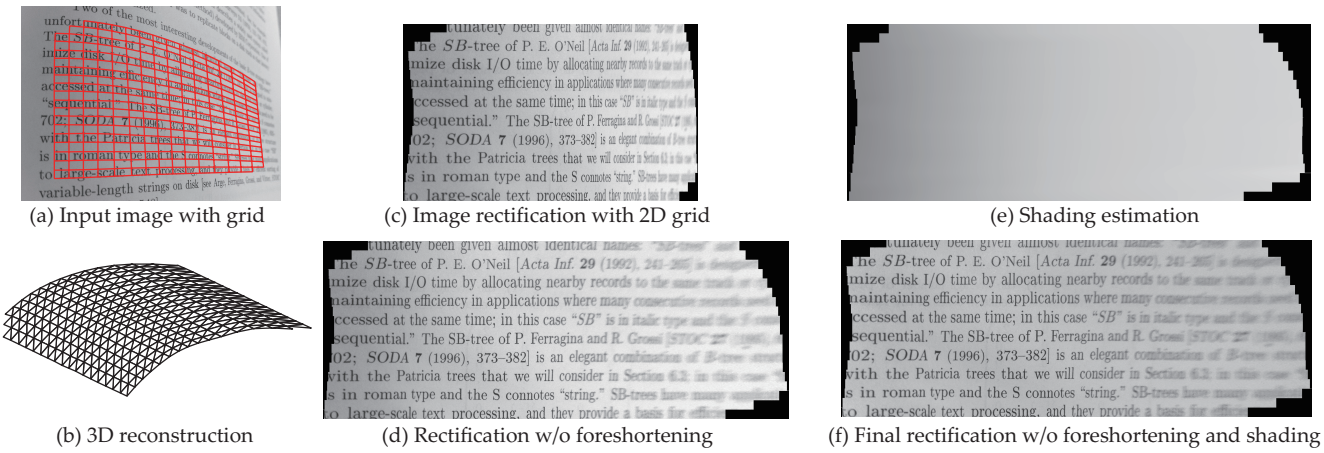
$$Q'(\{\mathbf{V}_i\}_{i=1}^n) = \sum_{j=1}^{N_p} \|\mathbf{\Delta}_{P_j}\|^2 + \alpha \sum_{i=1}^{n} (X_i - x_i Z_i)^2 + (Y_i - y_i Z_i)^2 \qquad (3)$$

where $\mathbf{V}_i = (X_i, Y_i, Z_i)$ is the estimated 3D location of the i-th grid vertex and α is the regularization constant. Eqn. 3 can also be solved exactly using SVD. Note that globally optimal solution is attained without initial guess of $\mathbf{Z}$.

There are a few special cases, e.g. plane and cylinder, in which the local parallelogram assumption is strictly true. In general, even if this assumption is only approximately true (because of local curvature), minimizing Eqn. 2 (or Eqn. 3) still gives very good estimations of a broad class of 3D shapes, including many non-ruled surfaces, as shown in Fig. 7. Fig. 9 shows a synthetic example in which text is mapped onto a sphere and projected back to the image plane. Using the methods in Section 3.1, we build the 2D coordinate grid and apply Eqn. 3 to obtain the 3D reconstruction. Note that, in principle, such surfaces cannot be reconstructed by previous approaches [8, 9, 17].

We quantitatively evaluate the 3D reconstructions obtained on a set of smooth surfaces randomly generated using 20 radial basis functions. Table 1 shows the relative root-mean-square errors between the ground truth and the 3D shape estimations. Gaussian noise is added to the 2D projections, with standard deviations as shown in the table. The average side length of grid cells is set to 1. The low errors demonstrate the robustness of our approach.

Note that more constraints could be incorporated into the



(a) Input image with grid (c) Image rectification with 2D grid (e) Shading estimation

(b) 3D reconstruction (d) Rectification w/o foreshortening (f) Final rectification w/o foreshortening and shading

Figure 8. 3D reconstruction and image rectification. **(a)** Original image with the 2D coordinate grid (Section 3); **(b)** 3D reconstruction from a single image (Section 4); **(c)** Image rectification using the 2D coordinate grid. Notice the foreshortening and shading effects. Using 3D information, **(d)** foreshortening can be rectified (Section 5.1) and by exploiting a reflectance model (e.g. Lambertian), **(e-f)** shading can be estimated and normalized to yield an albedo image. (Section 5.2).

optimization framework. A typical example is to enforce grid cells to be not only parallelograms but rectangles (text lines are horizontal and text orientation is vertical). However, such constraints introduce nonlinear terms in the optimization and global optimality can no longer be guaranteed.

5. Image rectification

5.1. Geometric rectification

While the vertical coordinates of the 2D warping grid is well-defined by interleaving text lines and white spaces, the horizontal coordinates along the text lines are not well-defined without depth information. An easy way to build the horizontal coordinates is to sample along the text lines with uniform image distance. This is a perfectly valid sampling for 3D reconstruction, but causes foreshortening effects in rectified text. As shown in Fig. 8(c), while all the text lines are horizontal, regions in the left appear stretched while regions in the right appear squished.

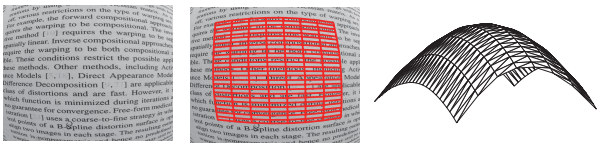
Fortunately, this foreshortening can be rectified using the 3D shape without knowing the font sizes, types and alphabet. Consider a patch within an image grid cell P_i. First we compute the 3D lengths of the two sides a_i and b_i of the parallelogram P_i and their ratio $r_i = a_i/b_i$. Then we warp the patch in P_i from the original image to a rectangle R_i of the same aspect ratio r_i. This is done by estimating a perspective transform that maps 4 corners of parallelogram P_i to the 4 corners of R_i. This process is applied to each grid cell independently. The result is shown in Fig. 8(d).

5.2. Photometric rectification

Using the estimated 3D shape, we can also remove the shading effects on the document image without knowing the prevailing lighting conditions. By assuming a Lambertian reflectance model, the pixel brightness at $\mathbf{x}$ is:

$$I(\mathbf{x}) = \rho(\mathbf{x})(\mathbf{n}(\mathbf{x}) \cdot \mathbf{w}) + \rho(\mathbf{x})A \tag{4}$$

where $\mathbf{w}$ is the unknown direction of lighting, A is the unknown ambient light and $\rho(\mathbf{x})$ is the unknown albedo. The surface normal $\mathbf{n}(\mathbf{x})$ can be computed from the 3D shape. We will further assume that the whitespace between lines (detected as described in Section 3) has uniform albedo. Then, we can set up a linear system of equations for patches in the whitespace to estimate the light direction $\mathbf{w}$, the ambient light A and the whitespace albedo ρ_w. The shading of the entire document image can be removed by computing the albedo image $\rho(\mathbf{x}) = I(\mathbf{x})/(\mathbf{n}(\mathbf{x}) \cdot \mathbf{w} + A)$. An example result is shown in Fig. 8(f).

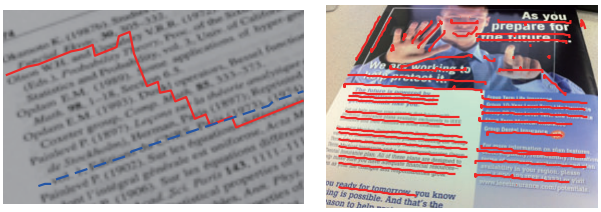


(a) (b) (c)

Figure 9. Example 3D reconstruction of text printed on a sphere. **(a)** The input image; **(b)** The estimated 2D coordinate grid (Section 3); **(c)** 3D reconstruction using Eqn. 3.

6. Experimental Results

We have applied our methods to documents with a wide variety of languages, font sizes and types, and challenging deformations. In order to demonstrate the ease of use, all the images were captured by an iPhone 4 camera. The focal length ($f = 2248$ pixels) is calibrated automatically within a few seconds using the *Theodolite app*. Fig. 12 shows representative results for curved documents written in English,



(a) (b)

Figure 10. Failure cases in line tracing. **(a)** Tracing with large step ($s = 5$) yields line skipping (red solid line). Tracing with small step ($s = 3$) tends to suffer less from line skipping (blue dotted line). **(b)** Line tracing on complicated text layout with images. The algorithm tends to follow straight lines in non-text region.

Chinese and Hindi. From a single image, our system automatically reconstructs the 3D shape and rectifies the document given a few user input specifying the image region for line tracing. The third and fourth columns show accurate removal of both the foreshortening and the shading effects. Fig. 11 shows the histograms of the white spaces in the documents before and after photometric correction. The narrow peaks demonstrate the accuracy of our system.

Failure cases. Fig. 10 shows several failure cases of line tracing. A large step (s large. See Section 3.1.1) in line tracing often yields line skipping, while a small step gives better results but runs slower. Besides, tracing is not working in non-text regions.

Performance. Our un-optimized MATLAB code takes 2-3 minutes to process an image (2592x1936) on Intel Core 2 (2.4GHz). The most time-consuming step is line refinement while others are fast. We are working on a C reimplementation on iPhone 4. The book imaging application requires fast capture but the processing can be done off-line.

There are several avenues of future work. We wish to extend our system to handle more general documents with images, text, and illustrations and handle non-smooth deformations such as folds, creases and tears. We also wish to build a rapid book scanning system using a high-speed camera that captures the images of quickly flipping pages.

Acknowledgements: This work was supported in parts by ONR grants N00014-08-1-0330 and N00014-11-1-0295 and Okawa Foundation Research Grant.

References

[1] A. A. Based, M. Pilu, and M. Pilu. Deskewing perspectively distorted documents: An approach based on perceptual organization. *HP White Paper*, 2001. 377

[2] M. Brown and W. Seales. Document restoration using 3D shape: a general deskewing algorithm for arbitrarily warped documents. In *ICCV*. IEEE Computer Society, 2001. 377

[3] M. Brown and W. Seales. Image restoration of arbitrarily warped documents. *PAMI*, 2004. 377

[4] H. Cao, X. Ding, and C. Liu. Rectifying the bound document image captured by the camera: A model based approach. *ICDAR*, 2003. 377, 378

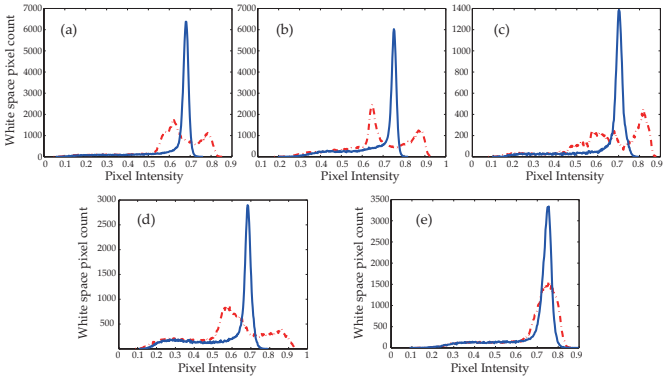


Figure 11. Histograms of white spaces in the documents before (red dotted lines) and after (blue solid lines) shading removal. The five histograms correspond to the five document images from top to bottom in Fig. 12. The intensity distributions on the computed albedo images are significantly narrower after shading removal.

[5] H. Ezaki, S. Uchida, A. Asano, and H. Sakoe. Dewarping of document image by global optimization. In *ICDAR*, pages 302–306, 2006. 378

[6] D. Forsyth. Shape from texture and integrability. In *ICCV*, 2001. 378

[7] H. Koo and N. Cho. State estimation in a document image and its application in text block identification and text line extraction. *ECCV*, 2010. 378

[8] J. Liang, D. DeMenthon, and D. Doermann. Unwarping Images of Curved Documents Using Global Shape Optimization. *CBDAR*, 2005. 378, 380, 381

[9] J. Liang, D. DeMenthon, and D. Doermann. Geometric rectification of camera-captured document images. *PAMI*, 2007. 378, 380, 381

[10] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *IJCV*, 23(2):149–168, 1997. 378

[11] J. Taylor, A. Jepson, and K. Kutulakos. Non-rigid structure from locally-rigid motion. In *CVPR*. IEEE, 2010. 381

[12] A. Ulges, C. Lampert, and T. Breuel. Document capture using stereo vision. In *ACM symposium on Document Engineering*, 2004. 377

[13] A. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 1981. 378

[14] C. Wu and G. Agam. Document image de-warping for text/graphics recognition. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 243–253, 2009. 378

[15] A. Yamashita, A. Kawarago, T. Kaneko, and K. Miura. Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system. In *ICPR*, 2004. 377

[16] A. Zandifar. Unwarping scanned image of Japanese/English documents. In *ICIAP*, 2007. 377, 378

[17] L. Zhang and C. Tan. Warped image restoration with applications to digital libraries. In *ICDAR*, 2005. 378, 381

[18] Z. Zhang, C. Tan, and L. Fan. Restoration of curved document images through 3d shape modeling. In *CVPR*, 2004. 378

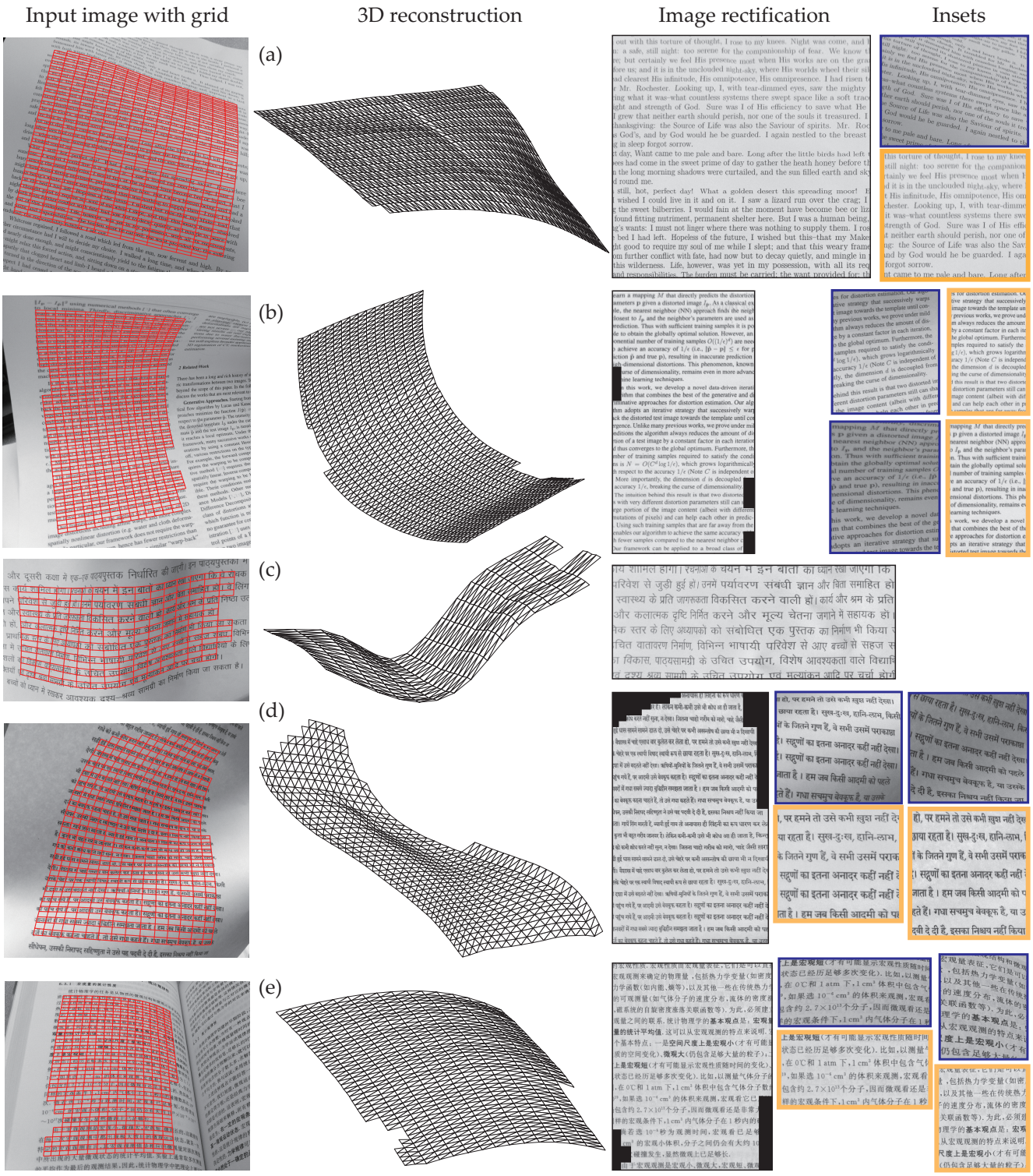


Figure 12. Image rectification and 3D reconstruction from a single curved document image. **First column:** Estimated 2D coordinate grid; **Second column:** 3D reconstruction. **Third column:** Image rectification. **Fourth column:** The insets show comparisons between rectified images (orange rectangles) and original distorted images (blue rectangles). The geometric deformations, text foreshortening and shading effects are all removed by our system. **(Please zoom in to see the details.)**