

# Multi-View Document Rectification using Boundary

Yau-Chat Tsoi and Michael S. Brown  
School of Computer Engineering, NTU  
Nanyang Avenue, Singapore 639798  
{tsoi0002,msbrown}@ntu.edu.sg

## Abstract

We present a new technique that uses multiple images of bound and folded documents to rectify the imaged content such that it appears flat and photometrically uniform. Our approach works from a sparse set of uncalibrated views of the document which are mapped to a canonical coordinate frame using the document’s boundary. A composite image is constructed from these canonical views that significantly reduces the effects of depth distortion without the blurring artifacts that is problematic in single image approaches. In addition, we propose a new technique to estimate illumination variation in the individual images allowing the final composited content to be photometrically rectified. Our approach is straight-forward, robust, and produces good results.

## 1. Introduction

### 1.1. Motivation

Imaging printed materials is an active task of libraries, museums, and large scale commercial ventures, with the goal of making printed content digitally accessible. The use of cameras in such efforts is rapidly becoming the accepted method of digitization [10]. One drawback to camera imaging is that the documents are not pressed flat when imaged. This can result in the document’s printed content appearing distorted in the captured image. Post-processing algorithms are therefore necessary to correct this distortion and rectify the printed content.

The degree of distortion correction needed is dictated by the use of the extracted content. For modern text-based documents, distortion correction is performed to the necessary level to make the imaged content suitable for OCR. If printed content beyond text is desired, for example, artwork, or books of antiquity, then distortion correction is needed to produce high-quality images that capture the “look and feel” of the original material. This paper targets this latter class of imaged materials. For such cases, two types of

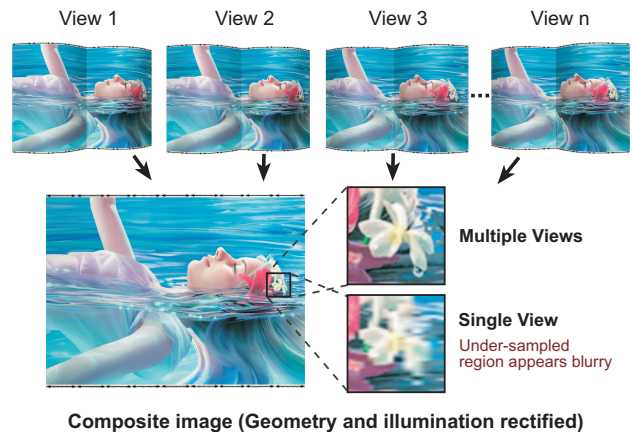


Figure 1. [Our approach] Multiple views of a document are composited to produce an output with rectified geometry and illumination. This multi-view approach reduces blurring in regions of high-curvature which is problematic for single-view approaches.

distortion must be rectified. The first is the geometric distortion caused by the materials shape; the second is effects from non-uniform illumination.

Several vision-based approaches have been proposed recently to address geometric distortion (e.g. [2, 3, 6, 8, 13, 16, 17]). A few of these approaches also include photometric correction [16, 17]. These existing approaches vary in the way they rectify the content, from cylindrical-model based approaches [3], to more general geometric structures [6, 8, 17], to non-parametric approaches that use 3D information [2, 13, 16].

All of these existing approaches have a common limitation that they operate from a single image. As a result, pixel under-sampling in regions of high-curvature on the document’s surface appear blurry in the restored image. The effects of such blurring is shown in Figure 1. For text-based documents this can often be ignored, since the final output is converted to a binary representation for use with an OCR engine. For non-textual items targeted in this paper, these blurry regions are undesirable and techniques to correct this problem are warranted.

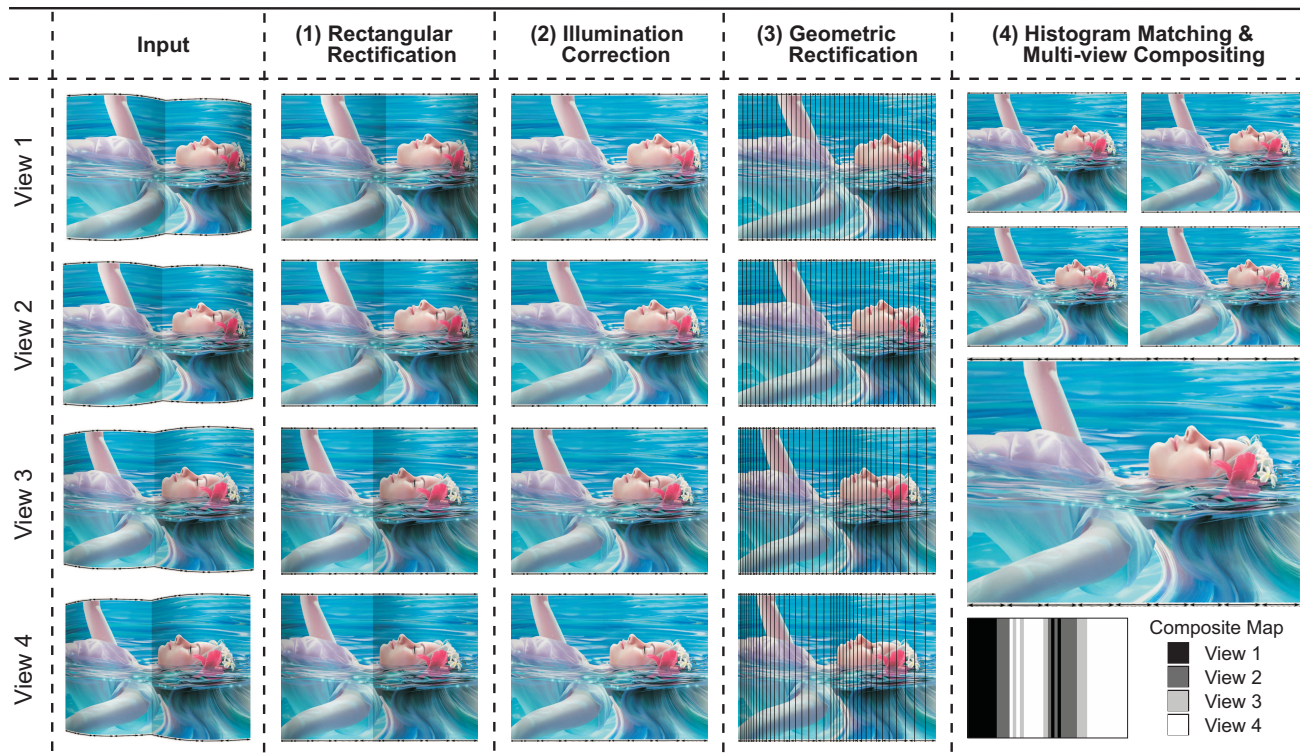


Figure 2. [Overview] Our framework has four phases: 1) Rectangular rectification; 2) Illumination correction; 3) Geometric rectification; 4) Histogram matching and multi-view compositing to build the final output.

## 1.2. Contribution and Related Work

We present a multi-view technique to rectify geometric and illumination artifacts in imaged documents. Our approach works by transforming the multiple views into a common coordinate frame based on the observed document boundaries. Working in this common coordinate frame, the best local sampling in the multiple views over the document’s surface can be determined. Compositing these views based on the sampling allows us to correct the effects from depth distortion without the need for explicit 3D information. Moreover, this allows the final output to avoid blurring artifacts present in single view approaches with inherent under-sampled regions. Our approach targets documents that can be modeled as ruled surfaces. Such ruled-surface models can represent the most common types of document distortion, including binder curl, folds, and their combinations. Exploiting the nature of the ruled surface, we also present a novel technique for estimating illumination directly from the individual views, allowing non-uniform illumination to be rectified.

Outside the scope of document restoration work, our compositing procedure is related to some degree to image mosaicing. Most notable is an approach to mosaic a sweep-camera described in Peleg and Herman [12], mosaicing slit images with 3D plenoptic function by Shum *et al* [15] and the mosaicing of views with relief geometry by Lhuillier *et al* [7]. Our approach, however, is more specific in its focus

and the restrictive nature of the document’s structure allows us to tailor a solution for document correction. Moreover, these mosaicing techniques do not require that their scene content be uniformly illuminated.

Several techniques for stitching multiple camera images of a document have been proposed, for example, work by Liang *et al* [9] and Mancas-Thillou and Mirmehdi [11]. These approaches reduce blurring effects from under-sampling, but are restricted to planar text documents and do not address illumination artifacts. In regards to document restoration work, we use the boundary interpolation discussed by Tsoi and Brown [17] to initially rectify our images, however, this previous single-view approach requires a 3D pattern to remove depth distortion and the photometric correction relies on white-borders in printed content. Our multi-image approach overcomes these previous limitations, avoiding the need for a 3D pattern and white-border, and does not suffer from image blurring.

## 2. Framework Overview

Figure 2 shows an overview of our proposed correction approach which is divided into four phases: (1) rectangular rectification; (2) illumination correction; (3) geometric rectification; (4) histogram matching and multi-view compositing. Each phase is discussed individually in the following sections.

## 2.1. Rectangular Rectification

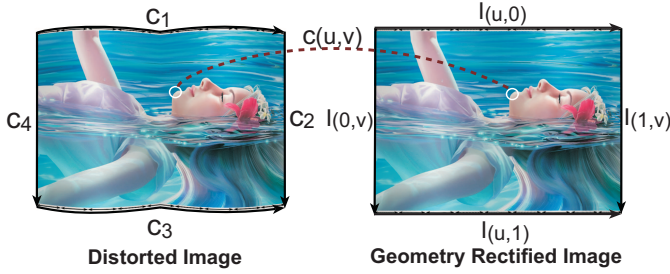


Figure 3. The document’s printed content is rectified to a rectilinear coordinate frame based on its observed four boundaries.

All input views of the document are rectified into a canonical coordinate frame using boundary interpolation proposed by [17]. Note that this procedure does not completely correct geometric distortion, as it cannot compensate for depth distortion. This can be clearly seen in Figure 2, where each view after rectangular rectification has compressed image regions related to depth and view angle. Solving this problem is described in Section 2.3. Rectangular rectification serves only as our starting point and helps simplify subsequent processing.

Figure 3 shows the basic idea of rectification using boundaries. The four boundary curves of the distorted document, denoted as  $c_1, c_2, c_3, c_4$ , are extracted from the input image. Each curve is represented as a piecewise natural cubic spline (NCS), where  $c_i = (x(t), y(t))$  is parameterized by  $t$  with range between 0 and 1. Each spline is specified by a set of 2D points  $\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n\}$  and their corresponding knot values  $\mathbf{T} = \{t_0, t_1, \dots, t_n\}$ .

The 2D points  $\mathbf{P}$  are acquired by detecting image coordinates along the boundary, while the corresponding knot values  $\mathbf{T}$  can be approximated by their 2D arc-length position, defined as follows:

$$t_j = \begin{cases} 0 & \text{if } j = 0 \\ \frac{1}{L} \sum_{1 \leq k \leq j} d(\mathbf{p}_{k-1}, \mathbf{p}_k) & \text{if } 1 \leq j \leq n \end{cases}, \quad (1)$$

where  $d(\mathbf{p}_{k-1}, \mathbf{p}_k)$  denotes the distance between the points, and  $L$  is the total length of the curve. Given  $\mathbf{P}$  and  $\mathbf{T}$  the coefficients of the NCS can be uniquely computed as described in any numerical methods book (e.g. [14]).

Assuming the imaged content is rectangular, a transformation,  $c(u, v)$ , from the input view to a new rectified image parameterized by  $u$  and  $v$ , where  $u, v \in [0, 1]$ , can be formulated using Coons patch interpolation [4] as follows:

$$\begin{aligned} c(u, v) = & [1 - u \quad u] \begin{bmatrix} c_4(v) \\ c_2(v) \end{bmatrix} \\ & + [c_1(u) \quad c_3(u)] \begin{bmatrix} 1 - v \\ v \end{bmatrix} \\ & - [1 - u \quad u] \begin{bmatrix} c_1(0) & c_4(1) \\ c_2(0) & c_3(1) \end{bmatrix} \begin{bmatrix} 1 - v \\ v \end{bmatrix}. \end{aligned} \quad (2)$$

Given  $c(u, v)$ , the input views,  $I_{input}(x, y)$ , can be rectified to a new image,  $I(u, v)$ , by the following  $I(u, v) = I_{input}(c(u, v))$ . Such boundary interpolation is modeling the document’s 3D surface as an opposite-boundary ruled surface, where the top and bottom image curves,  $c_1$  and  $c_3$  correspond to the 3D directrix and director curves. After rectification, image columns, parameterized by  $v$ , correspond to individual rulings on the 3D surface. The full details of this relationship is outside the scope of this paper and we refer the reader to [17] for further details. For our purposes, the alignment of surface rulings to image columns allows us to assume that 3D surface normals for each image column are the same. This convenient alignment will be exploited by our illumination rectification and multi-view compositing.

## 2.2. Illumination Rectification

After rectifying the multiple views to be rectilinear, illumination distortion is addressed. Our correction algorithm operates on the luminance component of the HSV colorspace and is denoted as  $I_V$ . Our goal is to remove shading artifacts present in  $I_V$  by normalizing intensity changes across image columns caused by illumination.

### 2.2.1 Assumptions

Illumination effects on the imaged scene are modeled via the intrinsic image model [1]:

$$f = ir = (\mathbf{l} \cdot \mathbf{n})r, \quad (3)$$

where  $f$  is an image point,  $i$  is the illumination falling on a point in the scene and  $r$  is the reflectance of the point in the scene. The value  $i$  can be decomposed using a Lambertian shading model that considers illumination falling on a 3D point to be the dot product of  $\mathbf{l}$ , which is a vector representing the direction and magnitude of the light source, and  $\mathbf{n}$  is the unit surface normal at the point. Note that this could also be extended to the sum of light sources falling at a single point.

Assuming that contribution from the lighting  $\mathbf{l}$  is reasonably constant between adjacent surface rulings, illumination change from one image column to the next can be attributed to a change in the surface normal direction. Since normals are constant for the entire ruling, intensity differences of



image pixels of homogeneous printed content between two image columns can be modeled as a simple scale change,  $\alpha$ , that corresponds to the effects of surface normal change between the columns. Thus, illumination change between image points of *homogeneous printed content* can be modeled as:

$$i_1 r = \alpha i_2 r \quad \text{for which} \quad \alpha = \frac{i_1}{i_2}, \quad (4)$$

where  $i_1$  and  $i_2$  denote the illumination falling on two surface points mapped to adjacent image columns that are assumed to have the same reflectance value,  $r$ .

### 2.2.2 Illumination Change Estimation

Based on our assumption, the illumination change between *all* successive  $k$  columns in the image can be modeled as per-column scale factors,  $\alpha_k$ . To estimate these  $\alpha$  values from Equation 4, it is necessary to find pixels with the same  $r$  between columns. To do so, we consider adjacent column pixels with sufficiently low image-gradients to be considered similar content. The following procedure is then used to estimate the  $\alpha_k$ . First, per pixel intensity ratios between columns, represented by image  $R(u, v)$ , are computed, where:

$$R(u, v) = \begin{cases} 0 & \text{if } u = 1 \\ \frac{I_V(u, v)}{I_V(u-1, v)} & \text{if } 1 < u \leq w \end{cases}, \quad (5)$$

where  $w$  is the width of the image  $I_V(u, v)$ .

Next, salient content edges in the image,  $I_V$ , are computed using a gradient-based edge detector, e.g. *Canny*, to form a mask,  $M(u, v)$ , consisting of either 0 or 1, where 1 represents an edge. Given the ratio image,  $R$ , and edge mask,  $M$ , we would like to find the scale-factor  $\alpha_k$  that minimizes the error between  $\sum_{v=1}^h \|(1 - M(k, v))(\alpha_k I(k, v) - I(k-1, v))\|_2$ , i.e. select  $\alpha_k$  that gives the minimum error for homogeneous pixels over the entire column  $k$ . While a least squares fit would be the mean  $\alpha$  for column  $k$ , because our computed mask only gives candidate pixels assumed to have similar content and is subject to outliers, we find the median a more robust choice. As a result, we compute the  $\alpha_k$  as follows:

$$\alpha_k = \begin{cases} 1 & \text{if } k = 1 \\ (\text{median}_{\substack{1 \leq v \leq h \\ M(k, v) = 0}} R(k, v)) \cdot \alpha_{k-1} & \text{if } k > 1, \end{cases} \quad (6)$$

where  $h$  is the height of  $I_V(u, v)$ . One special case that must be handled is in situations where an entire column has an edge in the mask  $M$ . This can occur in the document where it is bound or has crease lines from folds. In this situation, we simply skip this column and apply our procedure to the first none edge column found. Missing  $\alpha$ s for the edge columns are interpolated from their neighbor columns.

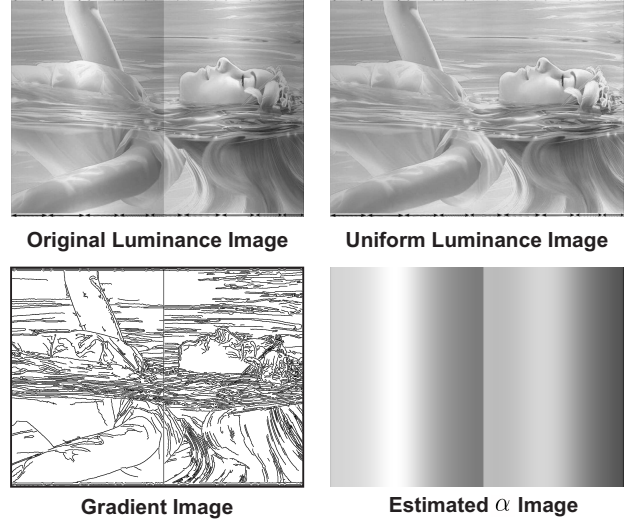


Figure 4. (Left column) Original luminance image and gradient image, (edge mask). (Right column) Corrected luminance and estimated per-column  $\alpha$  image.

### 2.2.3 Uniform Luminance Image

Given the scaling factors  $\alpha_k$ , the corrected luminance image  $I_V^{new}$  can be computed by dividing each image column in the original luminance image  $I_V$  by the  $\alpha_k$ s, such that:

$$I_V(k, \cdot)^{new} = \frac{I_V(k, \cdot)}{\alpha_k}, \quad (7)$$

where  $I_V(k, \cdot)$  represents all pixels at column  $k$ , over all image columns. Figure 4 shows the results of this procedure.

## 2.3. Geometric Rectification

As previously mentioned, the rectified single views suffers from noticeable geometric artifacts due to the document's non-planar structure. To address this problem, the multiple views are compared to find the best local sampling over the document's 3D surface from the various views. This allows regions of less sampling from one view to be replaced by better sampling from another view. Compositing sections of these views together, we produce a final output image that reduces the geometric distortion. Assumptions made about the nature of these multiple views are discussed in Section 4.

### 2.3.1 Column Registration and View Normalization

Assume that all the  $n$  input images,  $I_1, I_2, \dots, I_n$ , have undergone rectangular rectification and illumination correction. We now want to find a mapping between each images' columns to its corresponding location in the other views as shown in Figure 5.



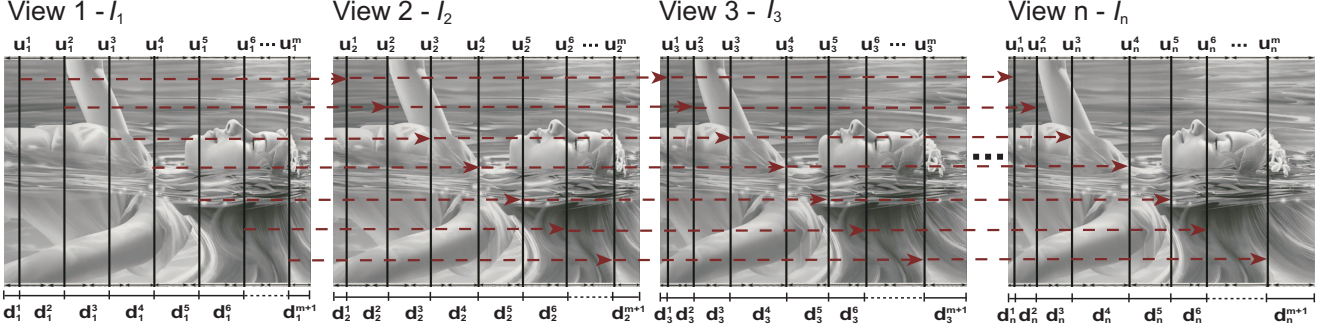


Figure 5. Example of column matching between  $n$  canonical frames. (Left to right)  $m$  image columns obtained from view 1 at positions  $u_1^1, u_1^2, \dots, u_1^m$  are matched to their corresponding positions in the other frames,  $I_j$ .

This task is performed by first extracting a sequence of  $m$  image grayscale columns from the first image  $I_1$  at positions  $\Psi_1 = \{u_1^1, u_1^2, \dots, u_1^m\}$ . These image columns are obtained by uniformly sampling along the  $u$  direction. Next, each of these image columns are matched to their corresponding locations in the other images. Normalized Cross Correlation (NCC) is used for matching to compensate for slight illumination differences in the various views. The matching of column,  $a$ , in image 1, denoted as  $u_1^a$ , to its location in image  $j$  is defined as:

$$c(k) = \frac{\sum [I_1(u_1^a, \cdot) - \overline{I_1(u_1^a, \cdot)}][I_j(k, \cdot) - \overline{I_j(k, \cdot)}]}{\sqrt{\sum [I_1(u_1^a, \cdot) - \overline{I_1(u_1^a, \cdot)}]^2 \sum [I_j(k, \cdot) - \overline{I_j(k, \cdot)}]^2}},$$

where  $1 \leq a \leq m$ ,  $k$  is the column index of  $I_j$ ,  $\overline{I_1(u_1^a, \cdot)}$  and  $\overline{I_j(k, \cdot)}$  denote the mean value of all pixels in the columns  $I_1(u_1^a, \cdot)$  and  $I_j(k, \cdot)$  respectively.

Allowing the  $k$  to span a pixel range from the previous matched location for  $u_1^{(a-1)}$  onwards, the maximum score  $c(k)$  is taken as the best matched column. This matched position for  $u_1^a$  will be denoted as  $u_j^a = k$ . This process is repeated for all image segments in  $\Psi_1$ . After the column registration, we obtain  $n$  sequences of column correspondences,  $\Psi_1, \Psi_2, \dots, \Psi_n$  for the  $n$  input views. These matched columns allow the best sampling of the document among each view to be determined.

“Best sampling” is defined to be the maximum pixel distance spanned by successive matched columns in each view. Let  $D_j = \{d_j^1, d_j^2, \dots, d_j^{m+1}\}$  be a sequence of distance values measured by the successive column positions found in each input  $I_j$ . Each scalar value  $d_j^l$  for  $1 \leq l \leq m+1$  represents the number of pixels available between the two successive columns positions. These distances are computed as:

$$d_j^l = \begin{cases} u_j^1 & \text{if } l = 1 \\ u_j^l - u_j^{l-1} & \text{if } 1 < l \leq m, \\ W - u_j^m & \text{if } l = m + 1 \end{cases}, \quad (8)$$

where  $W$  is the width of the image (see Figure 5). With this metric, we can calculate the pixel coverage by each view for each segment; e.g., the view that covers the most pixels for segment  $l$  can be found by checking the set of values,  $d_1^l, d_2^l, \dots, d_n^l$ . The one with the maximum value is taken as the best candidate for the current target segment, i.e.  $\arg \max_{1 \leq j \leq n} d_j^l$ .

## 2.4. Histogram Normalization and Multiview Compositing

The final image is generated by concatenating the column spans with the best pixel coverage selected from individual views. This will produce an image with depth distortion significantly reduced and will not suffer from image blurring. While the illumination of each view has been rectified, this rectification is not global across the multiple views. As a result, compositing image segments from the various views will result in noticeable illumination differences between segments. Thus, we first normalize the histograms of all the views and then perform compositing.

To normalize the illumination among the views, the view that contributes the largest number of pixels to the final output is chosen as the target image whose histogram all other images should map to. Such histogram mapping is described in any standard image processing text (e.g. [5]).

After histogram normalization, the final composite image is produced by copying successive maximum spanned segments for the  $n$  views. We show this as a composite map as shown in Figure 2 and in the following result section.

## 3. Results

The effectiveness of our approach is demonstrated on a synthetic example and two real examples. The synthetic model is created using OpenGL using a textured 3D model with shading (and without shading) and provides a means of comparing our results against ground truth. The real examples demonstrate the usefulness of our method on examples representative of a “real world” digitization setup.

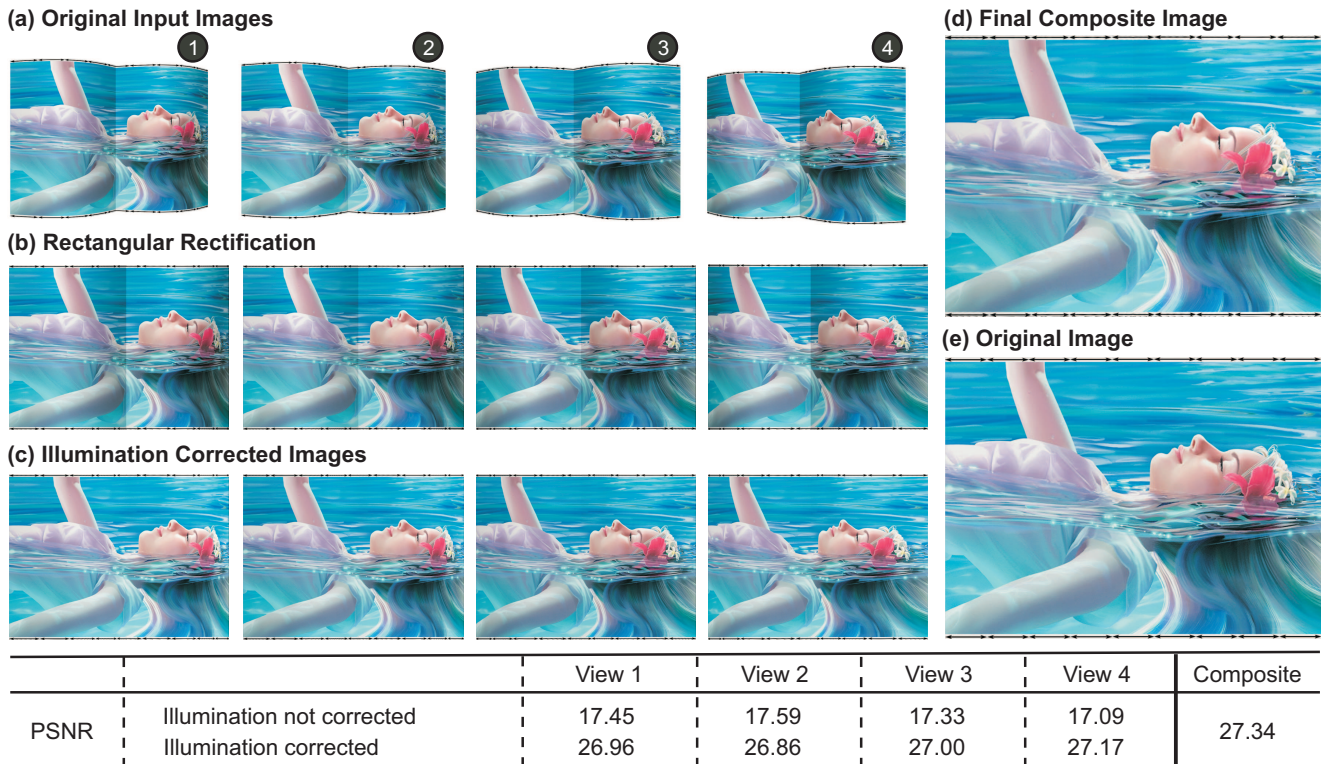


Figure 6. Results of our algorithm on a synthetic model representing distortion from binding. (a) Input images captured at four different views. (b) Rectangular rectification. (c) Illumination corrected. (d) Final composite image produced by stitching the best local pixel coverage from each views. (e) Original texture image used for the 3D model. The table shows the PSNR of the individual views compared with the synthetic model with shading turned off. We see our illumination correction significantly improves the PSNR.

### 3.1. Synthetic Result

Figure 6 shows the results on the synthetic example. A pattern of equally spaced markers is inserted at the top and bottom of the example. This allows geometric distortion to be easily observed. For example, the markers are unequal for the rectangular rectified images. However, the spacing is much more uniform in the composite view generated by our approach. We can compute the actual pixel differences for our composite image. The mean spacing of the inserted markers are approximately 110 pixels apart, which is only 4 pixels off of the actual spacing of 113 for the  $1024 \times 768$  image. This is an error of only 0.04.

For shading correction, we compute the peak signal-to-noise ratio (PSNR) for images with and without illumination correction. To provide ground truth, a reference image is generated by applying the rectangular rectification of each view with OpenGL shading turned off.

The table in Figure 6 shows the quantitative results for the PSNR. Significant PSNR improvements are obtained for shading corrected images versus those without shading correction. The original image used to texture the 3D surface and to simulate the printed content is shown. The final composite image generated by our algorithm is virtually identical.

### 3.2. Real World Examples

Results are demonstrated on two representative examples. Although ground truth is not available for these examples and the results are arguably subjective, the final output images generated by our approach are undoubtedly more desirable than any of the input images.

Figure 7 shows an example of a large three-page fold-out from an art book. Figure 7(a) shows the three input views. Figure 7(b) shows the corrected results using rectangular rectification. It is apparent that these corrected images suffered from noticeable depth distortion. This can be easily observed by looking at the uneven size of the folded pages. Figure 7(c)-(d) shows images with illumination corrected and the composite map. The final composite image is shown in Figure 7(e). Not surprisingly, for this particular example with page folds, the composite contributions are aligned with the page segments of the image based on each views angle to the individual segments.

Figure 8(a) shows another example of printed artwork from a large multi-page fold-out, captured in four different views. Figure 8(b) shows the rectangular rectification. Figure 8 (c, d and e) shows the corrected images with no shading, composite map and final composite image respectively. Again the geometric distortion in the single views

vs. the final output can be observed by the spacing of the individual pages.

## 4. Discussion

Our results demonstrate that our multi-view approach can effectively rectify geometric and illumination distortion. Our implementation assumes that the camera's line of sight is reasonably orthogonal to the material and foreshortening of rectified image columns is negligible. We also assume that the images are taken at roughly the same distance away from the document. Having one view that is significantly zoomed will result in distortion in the composite image. While such assumptions seem restrictive, they are typical of a working library or archive imaging setup, especially if physical housing or gantries are used to mount the cameras.

Another important aspect is the document's boundary extraction which is not elaborated on in this paper. While integral to the initial rectangular rectification, approaches for finding document boundaries are not particularly novel and are more suitable as an image processing course project. In our situation which is typical of real-world imaging setups, black or blue paper is placed behind each page during imaging. This simplifies boundary extraction to be a combination of thresholding and edge extraction using techniques such as a morphological gradient.

For our real-world examples, normal room lighting was used, which agrees with our assumption of slow varying lighting. Also, our real-world examples were from materials with non-gloss paper and therefore did not appear specular. High-gloss paper does cause problems due to specularities, however, this can be lessened using diffused light sources. We note in addition, that the slow varying illumination assumption is quite common and is the basis for other filtering for illumination correction, such as classic *homomorphic filtering* (see [5]). We have compared our results with homomorphic filtering and found that ours is far superior at preserving the actual printed content, as homomorphic filtering tends to wash out the filtered image.

Lastly, we note that our approach requires each view to see the entire document. This allows proper rectification into the rectilinear coordinate frame. The work in this paper provides the foundation for addressing partial views and such extension is slated for future work. In addition, incorporating super resolution techniques to our framework should be possible given the accurately registered views.

## 5. Conclusion

We have presented a new technique to use multiple views of a document to rectify the geometry and illumination. Our approach has the advantage over previous techniques that it can reduce depth distortion with no blurring artifacts. In

addition, we have introduced a new technique for illumination correction that works directly from the images. Our approach is straight-forward, robust, and produces good results.

## 6. Acknowledgement

We gratefully acknowledge grants ??? for supporting this work.

## References

- [1] H. G. Barrow and J. M. Tenenbaum. *Recovering Intrinsic Scene Characteristics from Images*. Academic Press, 1978.
- [2] M. S. Brown and W. B. Seales. Image restoration of arbitrarily warped document. *IEEE Trans. PAMI*, 26(10):1295–1306, 2004.
- [3] H. Cao, X. Ding, and C. Liu. A cylindrical surface model to rectify the bound document image. In *ICCV*, 2003.
- [4] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 1990.
- [5] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, USA, 2nd edition, 2002.
- [6] N. Gumerov, A. Zandifar, R. Duraiswami, and L. S. Davis. Structure of applicable surfaces from single views. In *ECCV*, 2004.
- [7] M. Lhuillier, L. Quan, H. Shum, and H.T. Tsui. Relief mosaic by joint view triangulation. In *CVPR*, 2001.
- [8] J. Liang, D. DeMenthon, and D. Doermann. Flattening curved documents in images. In *CVPR*, 2005.
- [9] J. Liang, M. DeMenthon, and D. Doermann. Camera-Based Document Image Mosaicing. In *ICPR*, 2006.
- [10] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents: A survey. *International Journal on Document Analysis and Recognition*, 7(2-3):84–104, July 2005.
- [11] C. Mancas-Thillou and M. Mirmehdi. Super-resolution text using the teager filter. In *First International Workshop on Camera-Based Document Analysis and Recognition*, 2005.
- [12] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *CVPR*, 1997.
- [13] M. Pilu. Undoing paper curl distortion using applicable surfaces. In *CVPR*, 2001.
- [14] W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery. *Numerical Recipes in C++: the art of scientific computing*. 2002.
- [15] H. Y. Shum and L. W. He. Rendering with concentric mosaics. In *SIGGRAPH*, pages 299–306, 1999.
- [16] M. X. Sun, R. G. Yang, Y. Lin, G. Landon, B. Seales, and M. S. Brown. Geometric and photometric restoration of distorted documents. In *ICCV*, 2005.
- [17] Y. C. Tsoi and M. S. Brown. Geometric and shading correction of imaged printed materials: A unified approach using boundary. In *CVPR*, 2004.



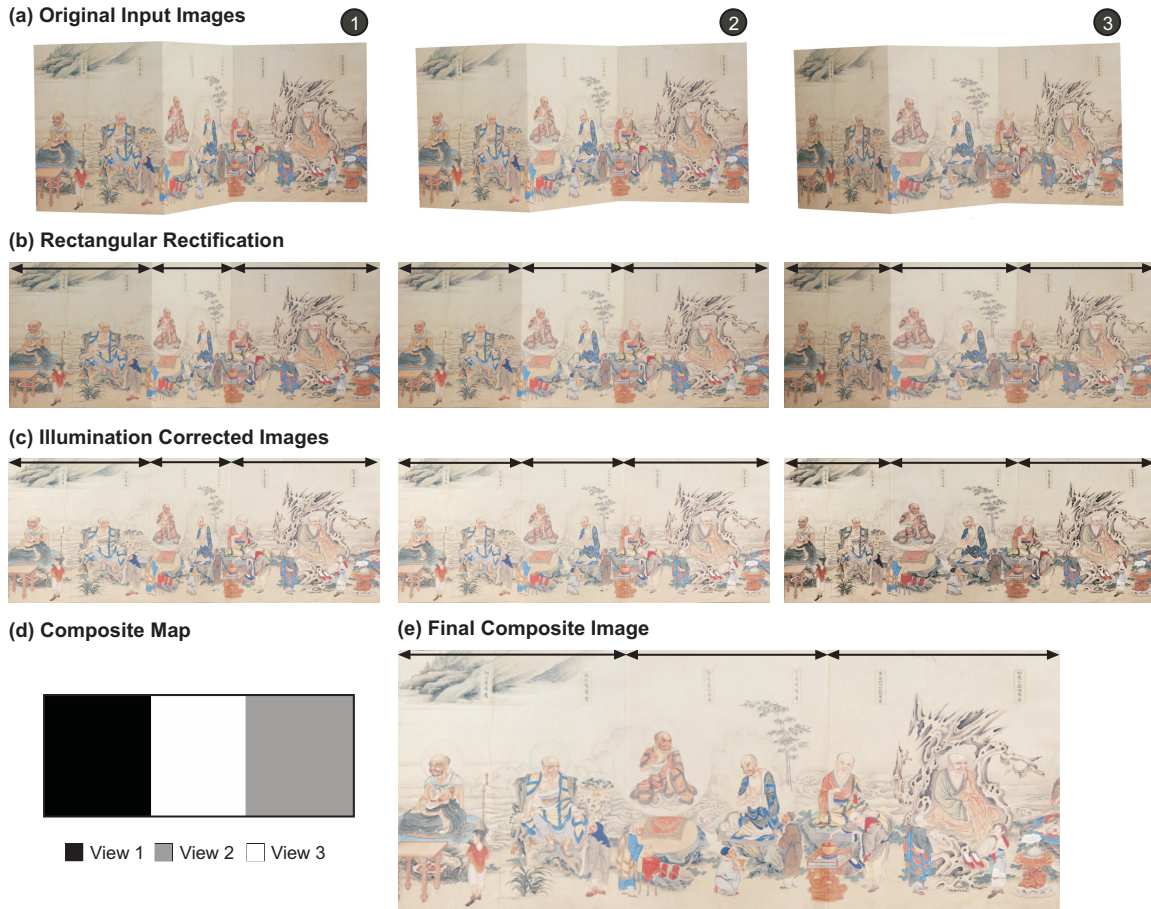


Figure 7. Example of wide three-page fold-out page from an art book. (a) Original input images. (b) Rectified views. (c) Illumination corrected. (d) Composite map. (e) Final composite image.

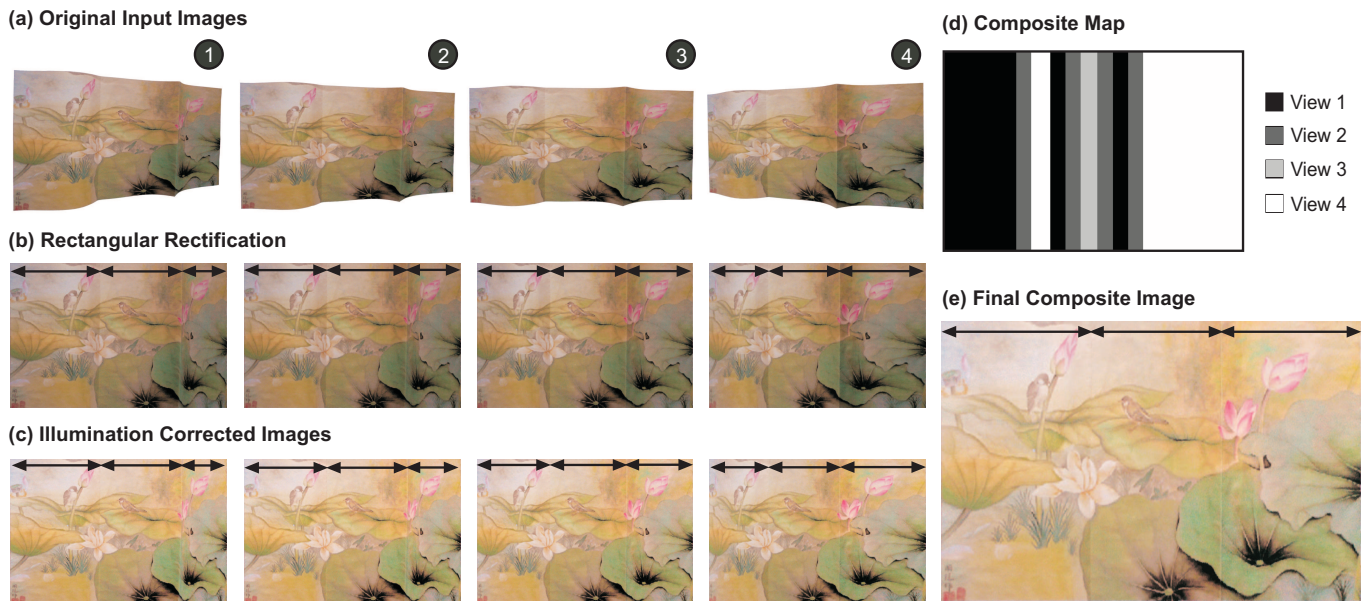


Figure 8. Example of three-page fold-out imaged art. (a) Original input images. (b) Rectified views. (c) Illumination corrected. (d) Composite map. (e) Final composite image.