Mosaicing of Camera-captured Document Images

Jian Liang [a], Daniel DeMenthon [b], David Doermann [b]

[a] *Jian Liang is with Amazon.com; Seattle, WA; USA.*

[b] *Daniel DeMenthon and David Doermann are with University of Maryland; College Park, MD; USA.*

Preprint submitted to Elsevier *11 June 2008*

Abstract

In this paper we present a method for composing document mosaics from camera-captured images. We decompose the complexity of solving the 8-dof transformation between image pairs into two problems, that is, rectification and registration. This is achievable under a key assumption that sufficient text content forms orthogonal texture flows on the document surface. First, perspective distortion and rotation are removed from images using the texture flow information. Next, the translation and scaling are resolved by a Hough transformation-like voting method. In the image composition part, our contribution is a sharpness based selection process which composes a seamless and blur free mosaic for text content. Experiments show that our approach can produce an accurate, sharp, and high resolution mosaic of a full document page from small image patches captured by a camera with various zooms and poses.

Key words: Camera-based document analysis, image mosaicing, image registration

1 Introduction

Digital image mosaicing has been studied for several decades, starting from the mosaicing of aerial and satellite pictures, and now expanding into the consumer market for panoramic picture generation. Its success depends on two key components: image registration and image blending. The first aims at finding the geometric relationship between the to-be-mosaiced images, while the latter is concerned with creating a seamless composition.

Email address: jliang@amazon.com (Jian Liang).

Many researchers have developed techniques for the special case of document image mosaicing [3,7,8,12,14,15,11,10]. The basic idea is to create a full, *frontal* view of a document page, often too large to capture during a single scan or in a single frame, by stitching together many small patches.

If the small images are obtained through flatbed scanners [3,12], image registration is less challenging because the overlapping part of two images differ only by a 2D Euclidean translation (plus slight rotation, if any), and there is no perspective distortion. When cameras are used, it is still possible to work within similar conditions. In some of the reported work the user is simply asked to point the camera straight at the document plane [7,14]. Others reinforce it with hardware support. For example, Nakao et al. [8] attach a downward looking video camera to a mouse such that displacements among images can be derived from mouse movement. In [15] Zappala et al. fix a downward looking camera overhead, and move a document on the desktop, which essentially mimics a scanner. The main weakness of either scanners or fixed cameras is their poor portability.

With portable cameras, perspective distortion may exist in the images. Registration is still possible. For example, feature point matching is a common approach in general image registration that is robust against projective transformation. There are affine invariant feature point detectors specially designed for text documents [13]. However, registration by itself does not remove the projectivity. Usually, for document mosaicing, the perspective distortion should be removed.

Motivated by structure-from-motion methods, Sato et al. moved from still image cameras to video cameras [11,10]. Their prototype system has an on-

line stage which tracks feature points across frames and generates a mosaic preview, and an off-line stage which refines the 3D reconstruction and final mosaic. The on-line stage essentially estimates the extrinsic camera parameters, i.e., pose or projectivity; the intrinsic parameters are irrelevant, as long as they are constant. In practice, this translates into using a fixed zoom. One disadvantage of video cameras in terms of document mosaicing is their limited resolution and motion blur.

Figure 1 shows two patches of a document captured by a camera with different zoom and poses. These two images differ in their perspective, resolution, brightness, contrast, and sharpness. Although many methods have been proposed for image registration ([9,4], to name a few), samples in Figure 1 still are challenging because of large displacement, small overlap, significant perspective distortion, and periodicity of printed text which presents indistinguishable texture patterns everywhere. For example, we were unable to get any meaningful results from global registration technique such as the Fourier-Mellin transform [9] whenever the overlapping area accounts for less than one fourth of each image. In terms of local feature point detection, we tested a general detector (PCA-SIFT [4]) with two robust estimators (Graduated Assignment [2] and RANSAC). The result is unsatisfactory because of the large number of outliers (above 90%) in the result from PCA-SIFT. Figure 2 shows the matched feature points found by PCA-SIFT for three pairs of images.

For example, in Figure 2(a), where PCA-SIFT is applied to two outdoor scenery images, most of the matches are correct, as shown by the fact that their connection lines have roughly the same length and direction. There are only a few incorrect matches and they stand out clearly. Figure 2(b) shows two document image patches with the same displacement and scale difference.

However, the percentage of incorrect matches is significantly higher because the periodicity of text lines and characters makes feature points less distinguishable from one another. Figure 2(c) shows the matched points between the two images in Figure 1. In this case, the incorrect matches are so overwhelming that it is very difficult to identify any good matches at a glance. Overall, this example shows that while it is easy to locate feature points in document images, it is more difficult to find good matches under perspective distortion and with small overlapping areas.

Our goal is to handle images such as those in Figure 1. Our method removes perspective distortion and registers images. While it is possible to first register images, then remove perspectivity, we found that once perspectivity is first removed, registration becomes easier. In order to estimate 3D structure information and then to remove perspective, a key assumption is that the document consists of sufficient text content which forms two orthogonal texture flows on the surface. In a certain sense, it is a structure-from-texture (or texture flow) method. First we remove perspective distortion and rotation in each image using the orthogonal texture flows. This step leaves a 3-dof transformation (a translation and a scaling) between any two overlapping views. Next we find feature point matches using PCA-SIFT. Although outliers still dominate, we are able to filter them out efficiently with a Hough transform-like voting method. After cross-correlation block matching, we obtain a refined registration between two images, where the perspective distortion is already removed.

With respect to image blending, there are three possible problems that have not been well addressed for document images. Conventional blending computes the weighted average in an overlapped area, i.e., $f = a_1 f_1 + a_2 f_2$, where f_1 and

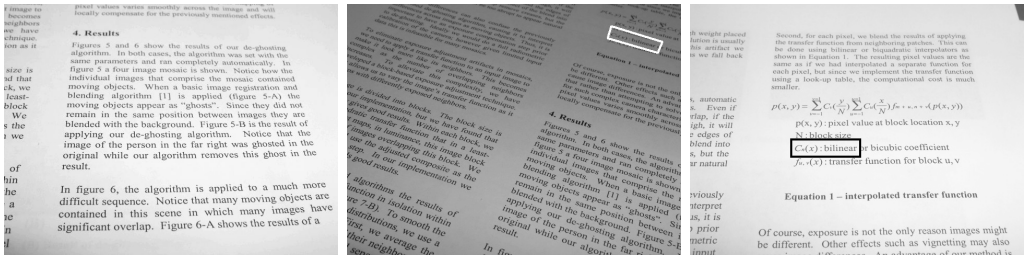
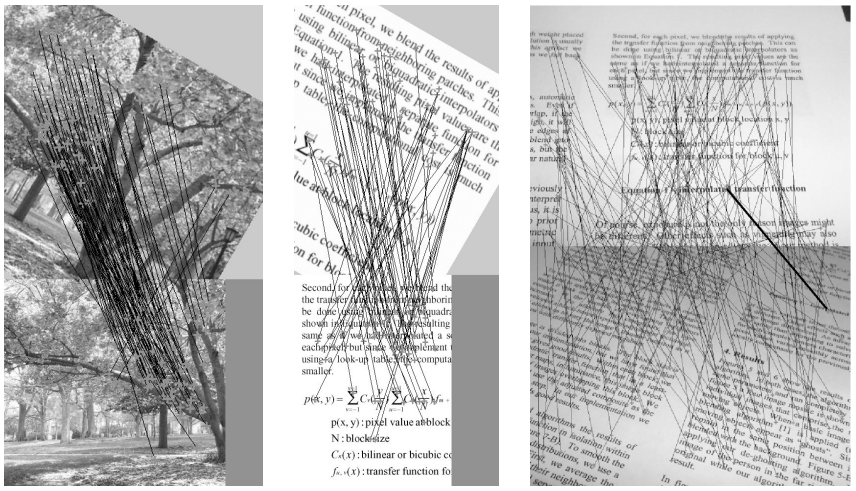


Fig. 1. Image patches of the same document captured from various positions. The same word in two views is marked by overlaid boxes.



(a) (b) (c)

Fig. 2. Match points found by PCA-SIFT. (a) Two sub-images with different scale and rotation generated from a scenery image. (b) Two sub-images from a document image with the same scaling and rotation as in (a). (c) Two camera-captured images of a document page. A thick black line shows one correct match.

f_2 are pixel values from two images, a_1 and a_2 are two weights that sum to 1. By varying the weights, one achieves a gradual transition from one image to another across the overlapping area. Other more sophisticated methods exist, which are essentially variations of weighted averaging [1]. Though averaging usually works well for general images, it is not optimal for document images.

First, the averaging methods treat only the overlapping area. They do not

address the overall uneven lighting across images. Second, registration may have errors. In mis-registered areas, weighted averaging would result in so-called 'ghost' images. Third, two images may have different sharpness because of different resolution, noise level, zooming, out-of-focus blur, motion blur, or lighting change. Weighted averaging essentially reduces the sharpness of the sharper image by blending a blurred image into it. Figure 3 shows the shortcomings of averaging method. For general scenery or portrait images, a certain amount of lighting variation and blurring is acceptable and 'ghosts' can be softened by blurring. However, for document images, viewers and OCR algorithms expect sharp contrast between text and background and minimum lighting variation. Therefore, averaging does not represent the optimal way of creating document mosaics.

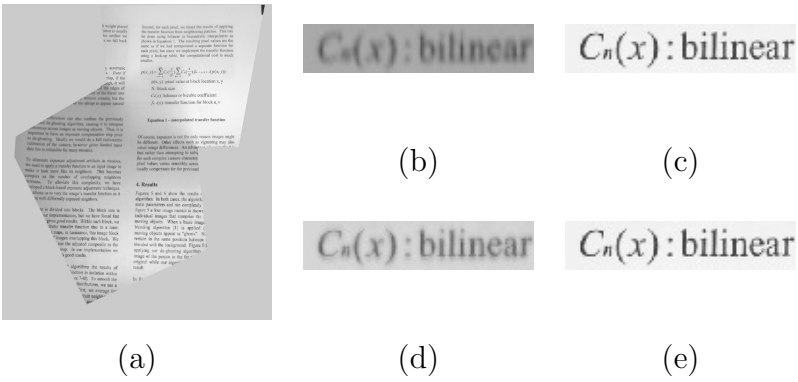


Fig. 3. Challenges for seamless image blending. (a) For two document patches with uneven lighting, their weighted average results in inconsistent contrast across the composite image. (b) A small portion of the darker image. (c) The same portion of the lighter image. (d) Weighted averaging result of (b) and (c) extracted from (a). (e) Our selective image blending result.

We treat the inconsistency of lighting by localized histogram normalization, which balances the brightness and contrast across two images as well as within each image. Then in the overlapped area, we perform a component level selective image composition which preserves the sharpness of the printed markings,

and ensures a smooth transition near the overlapping area border.

A shorter version of our work has appeared in [5]. In this paper we present our method in full details and provide more experimental results. Our prototype system can be illustrated by the pseudo-code in Figure 4. In the next sections we describe the three steps in details.

Input: two camera-captured document images, A_0 **and** B_0
Output: mosaic J_1 **free of perspective and rotation**
STEP 1: GEOMETRIC RECTIFICATION
Detect directions of text lines and vertical strokes;
Compute the vanishing points of text lines and vertical strokes;
Compute the homography from the vanishing points;
Remove the perspective and rotation using the homography: $A_0 \rightarrow A_1$, $B_0 \rightarrow B_1$, where A_1 and B_1 are free of perspective and rotation;
STEP 2: IMAGE REGISTRATION
Adjust the contrast by local histogram normalization: $A_1 \rightarrow A_2$, $B_1 \rightarrow B_2$;
Find feature point matches using PCA-SIFT: $(A_2, B_2) \rightarrow M_0$, where M_0 is a set of matched points;
Find the correct matches M_1 ($\in M_0$) and the scale r between A_2 and B_2 using compatible group voting: $M_0 \rightarrow (M_1, r)$;
Scale B_2 by r: $B_2 \rightarrow B_2'$;
Based on M_1, compute an initial registration H_0 between A_2 and B_2';
Using H_0, find a set of dense matched points M_2 between A_2 and B_2' using cross-correlation matching.
Using M_2, compute the final registration H_1 between A_2 and B_2';
STEP 3: SEAMLESS COMPOSITION
Compute the "sharpness" maps of A_2 and B_2': $A_2 \rightarrow S_A, B_2' \rightarrow S_B$, where S_A holds the sharpness measure of each pixel in A_2, and so is S_B;
Using H_1, composite an initial mosaic J_0 of A_2 and B_2' by averaging their overlapping part;
Find connected component set C in the binarized version of J_0;
For each element c of C do:
  Sum up the "sharpness" of all pixels in c: $S_A \rightarrow s_A^c, S_B \rightarrow s_B^c$;
  If $s_A^c > s_B^c$, copy all pixels in c from A_2 to final mosaic J_1; otherwise, copy from B_2';
End for
Fill other parts of J_1 by averaging A_2 and B_2';

Fig. 4. Workflow of procedure for mosaicing camera-captured documents

2 Document Image Rectification

Our mosaicing approach removes perspective distortion and rotation in document images in a step called geometric rectification. This step was described in great detail in [6]. Here we only provide a brief description for completeness. Please refer to [6] for implementation details. First, we detect text line direction and vertical character stroke direction in the image, using local projection profile analysis and directional filter respectively. Figure 2 shows the detected text line and vertical stroke directions superposed on the original text. Second, we find the vanishing points of these two groups of orthogonal directions using SVD decomposition. From these vanishing points we can estimate the focal length of the camera, the document plane orientation, and finally the homography that maps the two vanishing points to infinity at east and north. The result of geometric rectification is a document image that is free of perspective distortion and is rotated so that all text lines are horizontal. See [6] for details.

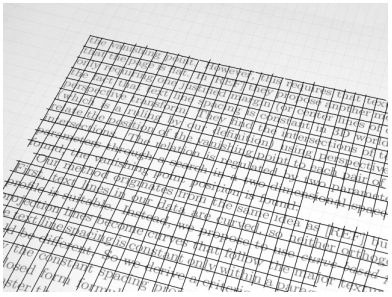
End 

Fig. 5. Text line and vertical stroke directions found in an document image

Figure 6 shows how the two rightmost images in Figure 1 are transformed by the rectification (followed by a local histogram normalization described in Section 4).

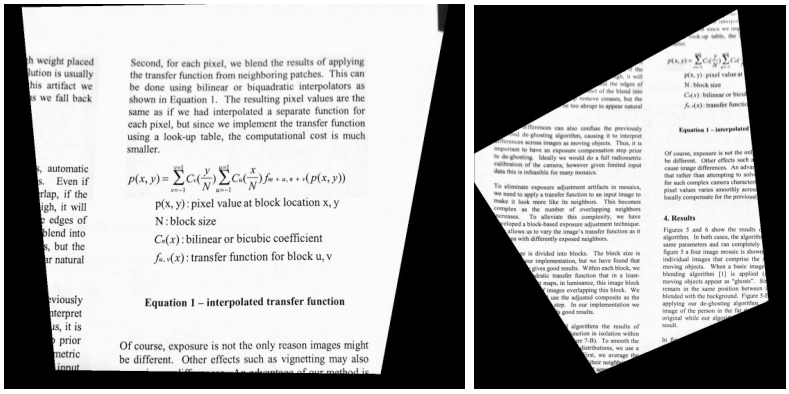


Fig. 6. Document patches after rectification and local histogram normalization.

3 Document Image Registration

Although projectivity has been removed after geometric rectification, small overlap, large displacement, and periodicity of texture are still challenging for common registration methods. For example, the Fourier-Mellin registration still fails because of insufficient overlapping, and PCA-SIFT still gives a lot of false matches that defeat Graduated Assignment and make RANSAC ineffective. However, we are able to filter out the outliers using a Hough transform-like voting mechanism, since we know only a translation and a scaling remain to be found.

First, let us assume the scale is known. Suppose two images (called A and B) are placed within the same coordinate system after proper scaling, and the true translation of image B with respect to image A is (x_0, y_0). Let $\{p_i\}_{i=1}^N$ be the feature points in image A, and $\{q_i\}_{i=1}^N$ be the matched points in image B. If p_i and q_i are a correct match, we have $q_i - p_i = (x_0, y_0)$, and an inequality otherwise. We compute all the displacements between matched points, i.e., let $q_i - p_i = (x_i, y_i)$. We have $(x_j, y_j) = (x_k, y_k)$ (we say that they are *compatible*), where j and k denote any two correct matches. Meanwhile, the probability

of having $(x_s, y_s) = (x_t, y_t)$, where either s or t denotes an incorrect match, is extremely low assuming incorrect matches are randomly distributed across the image. We group the matches with equal displacement (within a certain quantization bound) into compatible groups. Ideally, all correct matches are assigned to one group, while each incorrect match constitutes a group of its own. Hence, the correct matches are the matches in the largest group, and their displacements represent the correct translation. In practice, due to the quantization in the histogram used in our compatibility test (see below), some incorrect matches that have similar displacements may be placed in the same group. Even so, the sizes of such groups are highly unlikely to exceed the size of the group of correct matches.

If the scale estimate deviates from the correct value, the compatibility measure among correct matches will degrade. A small scale error can be absorbed by the histogram quantization. As the error increases, the group of correct matches will eventually split. Given a completely incorrect scale, the displacement distribution of correct matches will be as random as incorrect matches, so the largest compatible group will split into single-match groups. In summary, the largest compatible group is generated when the scale is correct.

Based on the above analysis, searching for the largest compatible group of matches as a function of scale can simultaneously solve the problems of finding 1) the correct matches, 2) the correct scale, and 3) the correct translation between two images. The specific procedure is as follows:

(1) For every scale s in a quantized range, construct the compatible groups and let $g(s)$ be the largest.
(2) Select s^* which maximize $|g(s)|$ and s^* is the correct scale.

(3) Find all matches in $g(s^*)$, and compute the mean of their displacements, which is the correct translation.

The scale range is quantized on a logarithmic scale. For a given scale, we use a 2D histogram of the match displacements in x and y to find the compatible groups. We divide the 2D displacement space into bins, and the displacement of each match falls in one bin. To address quantization error at bin boundaries, we smooth the 2D histogram by a 3×3 averaging kernel. Then, the bin with the most votes is the largest compatible group. The optimal bin size should be proportional to the average position error of the correctly matched feature points. We use an empirical value, i.e., 1/20 of the image diagonal length as the bin size. In practice, we find that the sensitivity of the method to this parameter is low (see below).

We use PCA-SIFT to find the matches between the two images in Figure 6. Figure 7 shows the number of matches for the first and second largest compatible groups found in 2D histograms as a function of the scale. The highest peak in the solid curve identifies the correct scale. At the correct scale, the second largest group (only three votes) is much smaller than the largest group (12 votes). This shows good aggregation of correct matches. After examination, we found the second largest group resides in a neighboring bin of the largest group, and the three matches are approximately correct. These two groups would merge if the bin size is increased. With different bin sizes we obtain curves slightly different from those in Figure 7. The correct scale is always found.

The figure also shows that when the scale is set slightly larger than the best value, the solid curve drops while the dotted line climbs. This means some

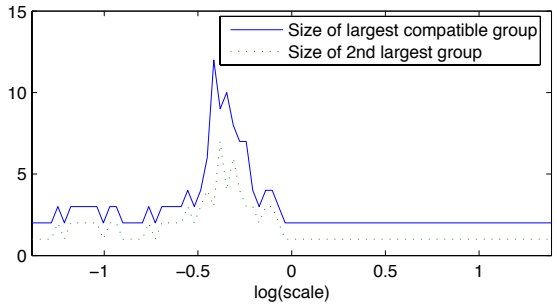


Fig. 7. 2D histogram peak values vs. scales

matches in the largest group shift to the second largest group in the neighboring bin. This confirms that the largest group splits when the scale is not perfect. When the scale differs significantly from the best value, either to the left of right, the solid curve drops to two matches [1] and the dotted curve shows only one match.

Given the best scale, we use the corresponding 2D histogram to find the matches aggregated in the largest group at this scale. Figure 8 shows the correct and incorrect matches.

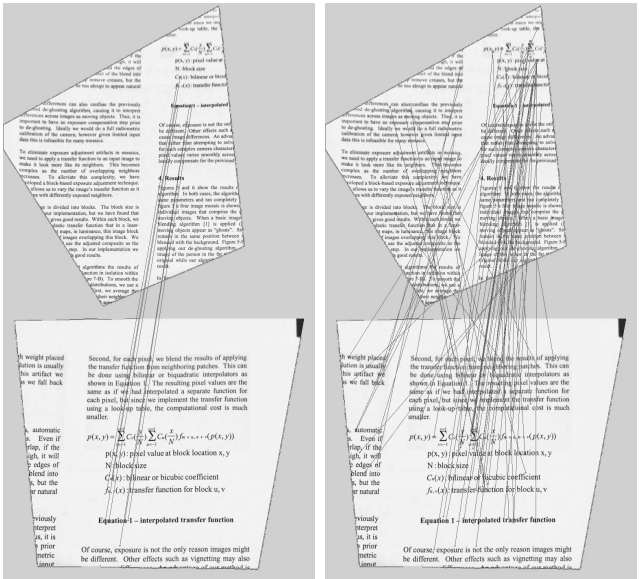


Fig. 8. Correct (left) and incorrect (right) matches in the PCA-SIFT result.

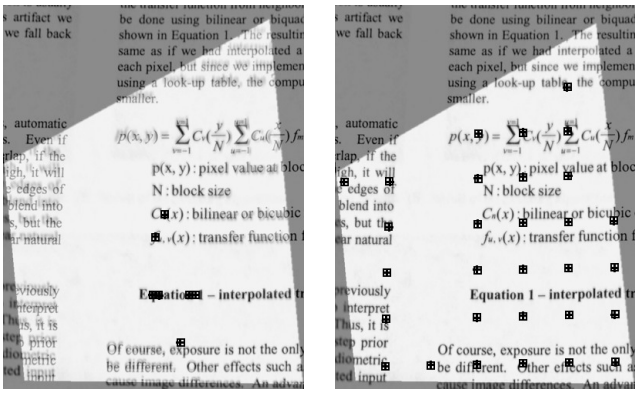
[1] The largest group has two matches because one pair of matched points is duplicated in the output of PCA-SIFT.

In the above analysis, the concept of compatibility groups is similar to the compatibility of matches in [3]. In a broader view, we can see that *voting* lies at the heart of our method, [3] and other RANSAC variations, and many geometric hashing based methods. To deal with outliers, RANSAC relies on the "random" chance of picking a good set, which could be inefficient or ineffective as the chance decreases when outliers dominate; geometric hashing solves the efficiency problem by doing the majority of computation off-line using training data (entire images, not local features). In the area of document mosaicing, our method takes advantage of the fact that image rectification can first remove a great part of the uncertainties, and as a result the voting itself becomes deterministic and efficient.

Taking the correct matches, we compute an initial projective transformation between the two images and map one into the other, as shown in Figure 9(a). However, because good matches tend to reside near the overlapped region's center, the registration is inaccurate near the border. We further refine the registration using cross-correlation block matching. This results in a dense and accurate set of matched points covering the whole overlapped area, which allows us to compute a refined projective transformation (see Figure 9(b)).

4 Seamless Composition

As we have stated in the introduction, there are three difficulties in creating a seamless document mosaic. The first is due to inconsistent lighting across two images. Conventional blending does not address overall lighting inconsistency, and it works well for general photos only because people accept lighting changes in natural scenes. However, documents are fundamentally



(a) (b)

Fig. 9. Image registration results where squares and crosses indicate the matched points in two images. (a) Registration using correct PCA-SIFT matches shows misalignment near borders of overlapping region. (b) Registration using additional matches obtained from cross-correlation block matching is very accurate.

binary with black print on white paper, and viewers' eyes are very sensitive to varying shade in documents. Typically, the histogram of a document image is bimodal. Different lighting conditions cause the two modes to shift. One way of balancing the lighting across two document images is to binarize both images. However, binarization introduces artifacts. Instead, we choose localized histogram normalization. The basic idea is to compute the local histogram in a small neighborhood, normalize the histogram such that the two modes are transformed to black and white respectively (or very dark and light gray). Histogram normalization preserves the transition between background and foreground, so the result is more pleasing to view. For documents containing grayscale or color contents, one choice would be to apply segmentation first, then compute histogram normalization parameters in the bimodal areas and estimate these parameters in grayscale or color areas via interpolation/expolation.

The second problem is registration error, and the third is uneven sharpness of

patch images. We solve both with *selective image composition*, i.e., each pixel in the result is chosen from the image with the best sharpness. We measure sharpness in an image by the local average of gradient magnitudes. In the following, the index of the selected image for a pixel is called the *decision* for this pixel.

The pixel-level decisions can be represented by a map in which the same decisions are grouped into regions. The boundaries of decision regions may intersect characters and words. Thus, if we apply pixel-level decisions directly, some characters or words may consist of pieces with different sharpness chosen from different images, which is not desirable. Furthermore, mis-registration tends to break decision regions into small pieces, resulting in 'ghost' images.

Therefore we aggregate the pixel-level decisions at the word level. This requires finding words. To do this, we compose an averaging image for the overlapped area, binarize it, dilate the foreground, and find connected components. The dilation has two effects. First, areas that may contain 'ghost' images are merged into the nearest component. Second, the width of our dilation kernel is set to be larger than its height, so components of a word are more likely to be merged than components from upper or lower text lines. As a result, most connected components contain a word. Next, all the pixels inside a connected component vote with their pixel-level decision, and the majority vote is taken as the component decision. The values for all the pixels of the component are selected from the winning image. This process ensures that 'ghost' images are eliminated and words do not have an uneven sharpness. For background areas (areas that are not included in word regions), the variation of sharpness is not visible, so we use the pixel-level decisions directly (without voting) to assign their values.

Figure 10 illustrates the process of selective image composition and the results. Figure 10(a) shows that most components consist of a single word. Figure 10(b) shows the component-level decision map by two shades of gray. The arrows indicate words that are cut into different parts in pixel-level decision map but not broken in component-level decision map.

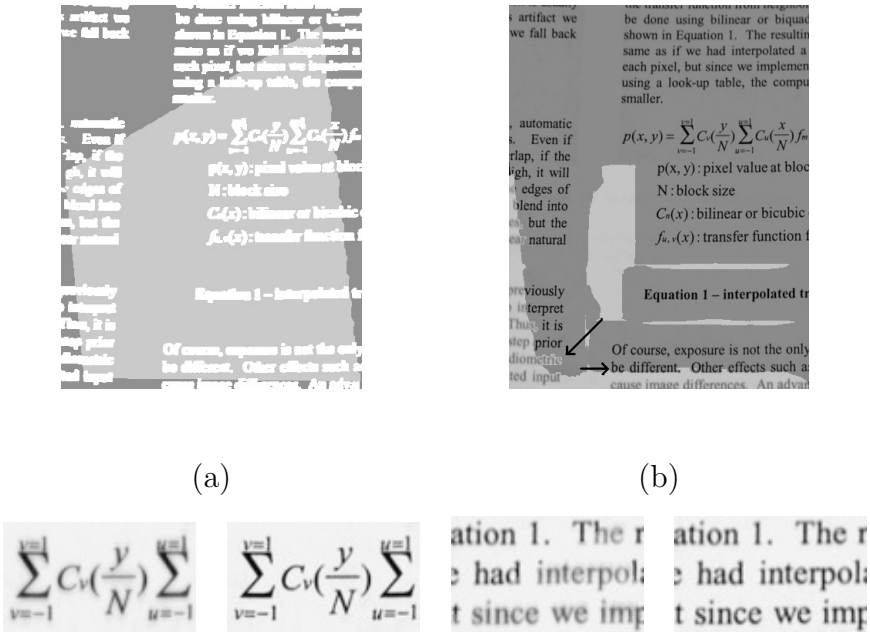
Words may still be broken by the boundaries of the overlapping area (e.g., "previously" and "interpret" in the lower left part of Figure 10(b)). In this case, half of the broken word has better sharpness than the other half. One could select the entire word from the image with lower sharpness to eliminate the difference in sharpness. This choice depends on user preference.

In the background area, the pixel-level decisions result in a large light gray region embedded in dark gray area. This does not create visible differences in the final image because the variation of sharpness in the background is small. In Figure 10, the comparison between (c) and (d) shows that our approach preserves the sharpness. In Figure 10(e), the overlapping area boundary is visible. It is eliminated in Figure 10(f).

5 Experiments

A quantitative evaluation of the rectification step (using synthetic data) is given in [6]. In this paper, we evaluate overall results on real images.

For each test document, we obtained a scan at 300 dpi and use the scan as the ground truth image. We took pictures of the document with a Canon EOS 300D digital camera (6M pixels) and a 28-80mm (35mm equivalence) zoom lens. All camera settings were put in auto-mode. First, we posted the



(a) (b)

(c) (d) (e) (f)

Fig. 10. Selective image composition. (a) Connected components are represented by white. The overlapping area is represented by light gray. (b) The binary selection decision map distinguished by dark and light gray. (c, e) Weighted averaging result. (d, f) Selective image composition result.

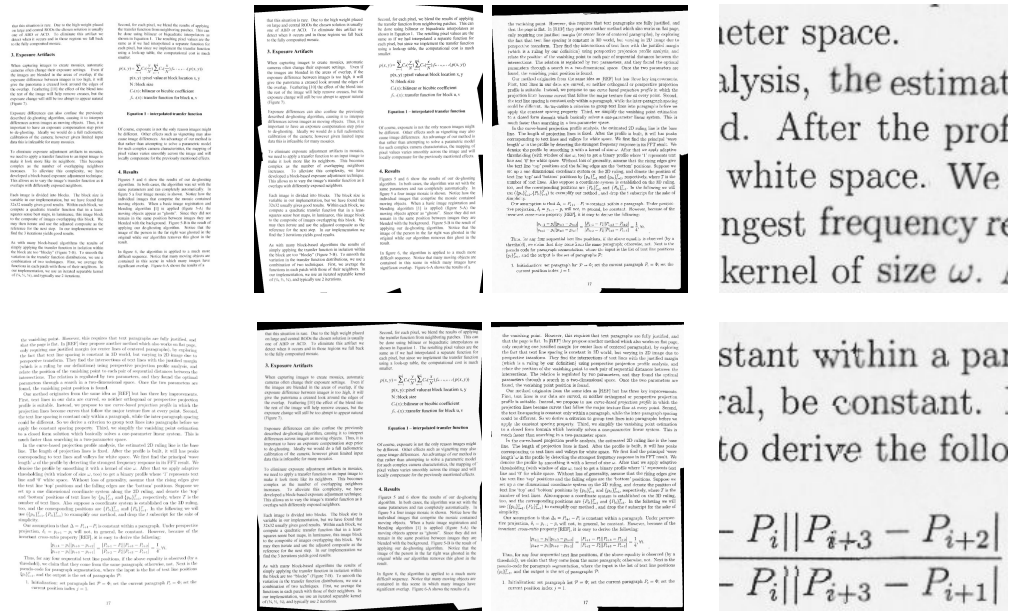
document on a wall, put the camera on a tripod, and carefully calibrated it so that the image is perspective free. We provided sufficient ambient light so that a small aperture (f5.6) could be used without flash. The result image represents, in some sense, the optimal non-mosaic image we can expect from the camera. Then we lowered the ambient light to typical indoor level (which caused the camera flash to go off in many cases), placed the document on a desk, sat at the desk, held the camera in hand and took a set of pictures with various angles and zooms. These image patches are fed to our mosaicing algorithm. Figure 1 contains three example image patches.

In the first round, we collected four documents from scientific journals and conference proceedings. We captured four patches for each document, making

sure that they all overlap each other. The perspective in the patches are kept moderate to minimum. This represents the scenario where the user has some control over the pose but needs higher resolution than a global view. We fed the four patches in all possible orders ($C_4^4 = 24$) to our mosaicing module. During the process, the first image serves as the initial composite, and each patch is registered to the current composite which contains all previous patches.

In the second round, we captured the images with less constraints to evaluate the limits of applicability of our approach. The number of patches varies from eight to twelve. Compared to the first round, the patches cover smaller areas, and perspective is from moderate to high. This simulates the case where a low resolution camera is used and it is difficult to control the document pose. We visually inspected and recorded which patches overlap each other. Then ten random orderings of the patches were generated under the condition that each patch overlaps with at least one patch before it. This ensures that when a patch is fed to the mosaicing module, it overlaps with the composite generated so far.

We find no significant difference between the result in the first round and the second round, except for expected differences of resolution. Figure 11 shows the camera-captured frontal views of two documents, and two composite images for each document. The global views in Figure 11(a) are enhanced by the same local histogram normalization used in document mosaicing. Slight barrel effects are visible, due to the wide lens distortion. All composites in Figure 11(b) have higher resolution and do not show barrel effect, compared to (a). The enlarged views show two portions near the border of underlying patches. There are some fuzziness, misalignment, and "ghost" effects. Nevertheless, the border itself is undetectable.



(a) (b) (c)

Fig. 11. Perspective-free images compared to composite images. (a) Full frontal views captured by a camera. (b) Two composite images for each document. (c) Enlarged portions of composites near borders between underlying patches.

Except for visual inspection which could be subjective, we also compared the mosaics to the global views quantitatively in an overall sense. Since our samples are mostly text documents, we used OCR as the image quality appraiser. For each document, we applied OCR to the digital scan, the camera-captured global view, and all the composites. We used the OCR text from the scan as the ground truth, against which we computed the character and word recognition rates for the global views and composites, respectively. Table 5 shows the number averaged over all documents. The OCR performance on composites is very close to that on the perspective free global view.

The PCA-SIFT used in our experiment is trained using generic image data. We also tried document images as training data. However, no difference was found in their performance in percentage of false alarms.

	Global views	Composite images
Character recognition rate	92.3%	91.0%
Word recognition rate	89.2%	89.5%

Table 1

Average OCR rates of global views and composite views.

In our experiments, computing a mosaic could be very time consuming. Depending on hardware and image size, it may take up to ten minutes. This is partially because our prototype is built in MATLAB and not optimized for speed. The most demanding part is rectification, especially the texture flow computing. The blending part comes second. PCA-SIFT comes third. The registration step is negligible. Overall, the complexity is roughly linear in terms the numbers of pixels.

6 Summary

In this paper we demonstrate a document mosaicing method which deals with severe perspective distortion, large displacement and small overlapping area. The first step, geometric rectification, greatly reduces the complexity of the registration problem. The second step, registration, is robust against large number of outliers found by feature point matching algorithms. The last step, blending, composes a seamless, "ghost" free mosaic with optimal sharpness. While the rectification step only works on text areas in documents, the other two steps can be applied to non-text images without significant modifications.

References

[1] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[2] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. PAMI*, 18(4):377–388, April 1996.

[3] F. Isgrò and M. Pilu. A fast and robust image registration method based on an early consensus paradigm. *Pattern Recognition Letters*, 25(8):943–954, 2004.

[4] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *Proc. CVPR*, volume 2, pages 506–513, 2004.

[5] J. Liang, D. DeMenthon, and D. Doermann. Camera-based document image mosaicing. In *Proc. ICPR*, 2006.

[6] J. Liang, D. DeMenthon, and D. Doermann. Geometric rectification of camera-captured document images. *IEEE Trans. PAMI*, 30(4):291–605, Apr. 2008.

[7] M. Mirmehdi, P. Clark, and J. Lam. Extracting low resolution text with an active camera for OCR. In *Proc. IX Spanish Sym. Pat. Rec. and Image Proc.*, pages 43–48, May 2001.

[8] T. Nakao, A. Kashitani, and A. Kaneyoshi. Scanning a document with a small camera attached to a mouse. In *Proc. WACV'98*, pages 63–68, 1998.

[9] B. S. Reddy and B. N. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Trans. Image Proc.*, 5(8):1266–1271, 1996.

[10] T. Sato, A. Iketani, S. Ikeda, M. Kanbara, N. Nakajima, and N. Yokoya. Mobile video mosaicing system for flat and curved documents. In *Proceedings of 1st International Workshop on Mobile Vision*, pages 78–92, 2006.

[11] T. Sato, A. Iketani, S. Ikeda, M. Kanbara, N. Nakajima, and N. Yokoya. Video mosaicing for curved documents based on structure from motion. In *Proc. ICPR*, volume 4, pages 391–396, 2006.

[12] K. Schutte and A. M. Vossepoel. Accurate mosaicking of scanned maps, or how to generate a virtual A0 scanner. In *Proc. ASCI'95*, pages 353–359, 1995.

[13] M. I. Tomohiro Nakai, Koichi Kise. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In *International Workshop on Document Analysis Systems*, pages 541–552, 2006.

[14] A. P. Whichello and H. Yan. Document image mosaicing. In *Proc. ICPR*, pages 1081–1083, 1998.

[15] A. Zappala, A. Gee, and M. J. Taylor. Document mosaicing. *Image and Vision Computing*, 17(8):585–595, 1999.