Camera-Based Document Image Mosaicing

Abstract

In this paper we present an image mosaicing method for camera-captured document images. Our method is unique in not restricting the camera position, thus allowing greater flexibility than scanner-based or fixed-camera-based approaches. To accommodate for the perspective distortions introduced by varying poses, we implement a two-step image registration process that relies on accurately computing the projectivity between any two document images with an overlapping area as small as 10%. In the overlapping area, we apply a sharpness based selection process to obtain seamless blending across the border and within. Experiments show that our approach can produce a very sharp, high resolution and accurate full page mosaic from small image patches of a document.

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.

Many researchers have developed techniques for the special case of document image mosaicing [2, 5, 6, 8, 9, 10]. The basic idea is to create a full 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 [2, 8], image registration is somewhat easier because the overlapping part of two images differ only by a 2D Euclidean transformation. However, if the images are captured by cameras, the overlapping images differ by a projective transformation. Virtually all reported work that we are aware of on document mosaicing using cameras impose some restrictions on the camera position to avoid perspective distortion. Some of them simply ask the user to point the camera straight at the document plane [5, 9]. Others require hardware support. Nakao et al [6] attach a video camera to a mouse, facing down at the document page. While a user drags the mouse across the page, a sequence of pictures are taken, and registered pairwise with the help of mouse movement. In [10] a overhead camera is fixed facing down while the document is moved on the desktop. While hardware support reduces projective transformations to Euclidean transformations, it defeats one purpose of using cameras, which is to achieve flexibility and convenience.

Our goal is to get back flexibility and convenience, so that users can take pictures from any position. Fig. 1 shows some example images that we are able to process. While there have been hundreds of papers written on image registration ([7, 3], to name a few), the images in Fig. 1(a,b) are still very difficult to register because the displacement is large, the overlapping areas are small (around 10%), the perspective distortion is significant, and the periodicity of printed text presents similar texture patterns everywhere. The Fourier-Mellin registration method [7] did not work well on them. We have also tried robust estimators (RANSAC and SoftAssign [1]) with a feature points detector (PCA-SIFT [3]), which failed because the periodicity of text leads to large number of outliers (up to 90%) in feature point matches.

Fig. 1 also reveals three other problems in image blending that have not been well addressed. First, the lighting is inconsistent between two images, which is common for consumer grade cameras with inaccurate auto-exposure feature and on-camera flash. 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 f_2 are pixel values from two images, a_1 and a_2 are two weights that sum up to 1. By varying the weights one can achieve a gradual transition from one image to another across the overlapping area. There are other more sophisticated methods, but while they may work for natural scenes, they are not optimized for document images. Fig. 1 (c) shows the result of a simple blending method. Although other sophisticated methods may do better inside the overlapping area, the point is that it only affects the overlapped area; the overall lighting variation is still not satisfactory.

Second, there could be errors in registration. If so, weighted averaging would result in double or 'ghost' images. Third, two images have different sharpness, either because of different resolution due to camera zooming, or because of out-of-focus blur in either image (see Fig. 1(d,e)). Weighted averaging is only adding blurring in one image to

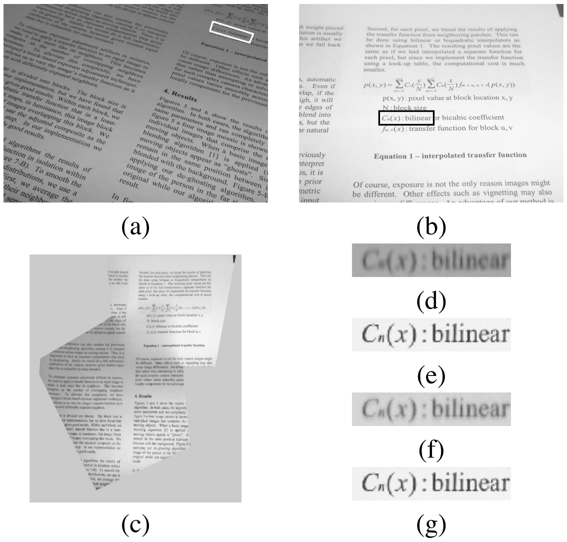


Figure 1. Challenges for camera-based document image moscaicing. (a,b) Camera captured images. (c) Result of perspective rectification, image registration, and conventional weighted average blending. (d) Rectified small portion of (a). (e) Rectified small portion of (b). (f) Weighted averaging result of (d) and (e) extracted from (c). (g) Our selective image blending result.

sharpness in the other. For natural scene photos, when registration error is small, visual appearance will generally be acceptable. However, for document images, any small offset and blurring is noticeable and affects visual recognition of black markings on white paper.

Our proposed registration method for two overlapping views consists of two steps. First we remove perspective distortion and relative rotation of individual views using text lines and vertical character strokes detected in document images. This step removes perspective foreshortening and rotation, and leaves only a translation and a scaling between the two views. Then, we find feature point matches between views using PCA-SIFT. Although outliers still dominate, we are able to filter them out efficiently. After refining the transformation with cross-correlation block matching results, we can obtain a very accurate registration for the image pair.

We treat the inconsistency of lighting by localized histogram normalization, which balances the brightness and contrast across two images as well as within each. 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.

2. Document Image Registration

The first step of our image registration method is to remove perspective foreshortening and rotation between two images of the document, and leave only a three-parameter transformation combining a translation and a scaling between the two images. We achieve this by removing the perspective distortion in both images. The basic idea is to detect text line directions and vertical character stroke directions, find their vanishing points, and the homography that maps the vanishing points back to infinity (see [4]).

Ideally, the two resulting perspective-free images now should only differ by a translation and a scaling. Although projectivity is gone, large displacement, small overlap, and periodicity of texture still prevent common registration methods from succeeding. For example, the Fourier-Mellin method still fails, and PCA-SIFT still gives a lot of false matches that defeat SoftAssign and make RANSAC impractical. However, we propose a novel way of filtering out the PCA-SIFT outliers at this point, taking advantage of the fact that the transformation between the two images now depends only of two translation parameters and one scaling parameter.

Let's first assume that we know the scale. Suppose that two images are placed within the same coordinate system after proper scaling, and the true translation of image 2 with respect to image 1 is (x_0, y_0). Let $\{p_i\}_{i=1}^N$ be the feature points in image 1, and $\{q_i\}_{i=1}^N$ be the matched points in image 2. If p_i and q_i are a correct match pair, we should have $q_i - p_i = (x_0, y_0)$, and inequality otherwise. Since we do not know (x_0, y_0), nor the correct matches, this equation itself does not lead to anything. However, if we compute all the displacements between matched points, i.e., let $q_i - p_i = (x_i, y_i)$, we will have $(x_j, y_j) = (x_k, y_k)$ (we say that they are *compatible*), where j and k denote any two correct matches. In the meantime, the probability of $(x_s, y_s) = (x_t, y_t)$ where either s or t denotes an incorrect match is extremely low assuming that q_s or q_t is randomly distributed across the image. Therefore, the true translation is the one supported by the largest group of compatible match pairs. After that, it is straightforward to find the correct matches.

If the page is not at the same scale in the two images, the compatibility between correct matches will degrade and so will the support for the true translation vector. When the scale is totally wrong, the distribution of the correct matches will be as random as the incorrect ones, so that the size of the largest group of compatible matches will diminish to one or two matches.

Therefore, we can search for the best scale and translation by looking for the largest group of compatible matches. For a given scale, we compute a 2D histogram of the displacement vectors, and each matched pair contributes one vote in one of the bins. The optimal bin size should be proportional to the average position error of the correctly matched feature points. In practice, we find that this is not very critical. We use 1/20 of the image diagonal length. Fig. 2 shows the sizes of the first and second largest groups of compatible matches found in 2D histograms for different scales. The results are based on images in Fig. 1. The highest peak in the solid curve identifies the correct scale. The largest group size remained at 2 when scale is totally wrong because PCA-SIFT repeated a pair of matched points in its output. At the correct scale, the second largest group (only 3 counts) is much smaller than the largest group (12 counts). This shows that the aggregation of correct matches is good, which in turn means robustness against noise. The figure also shows that when the scale is a little bit off to the right, some matches in the largest group get moved to the second largest group at the neighboring bin, which confirms our statement that the compatibility among correct matches degrades. As a side benefit, the ratio between the two curves may serve as a reliability index for the identified scale.

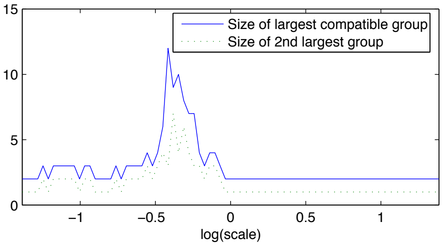


Figure 2. 2D histogram peak values vs. scales

Once the scale is known, we use the corresponding 2D histogram to find the correct matches aggregated in the largest group. Using this group we compute an initial projective transformation between the two images, and map one into the other, as shown in Fig. 3(a). Because good matches tend to reside near the center part of the overlapped region, the transformation is inaccurate around the border area. We further refine the transformation using cross-correlation block matching. This results in a dense and accurate matched point set covering the whole overlapped area, which is used to compute the projective transformation (see Fig. 3(b)).

3. 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 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.

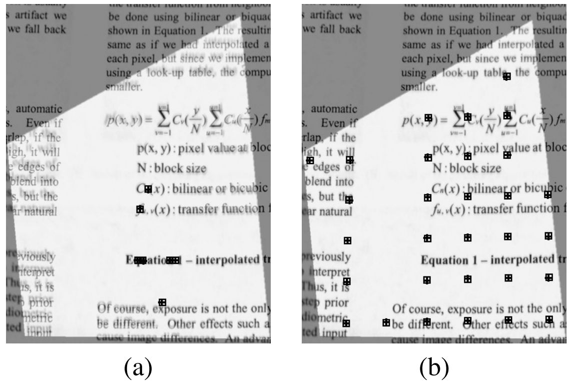
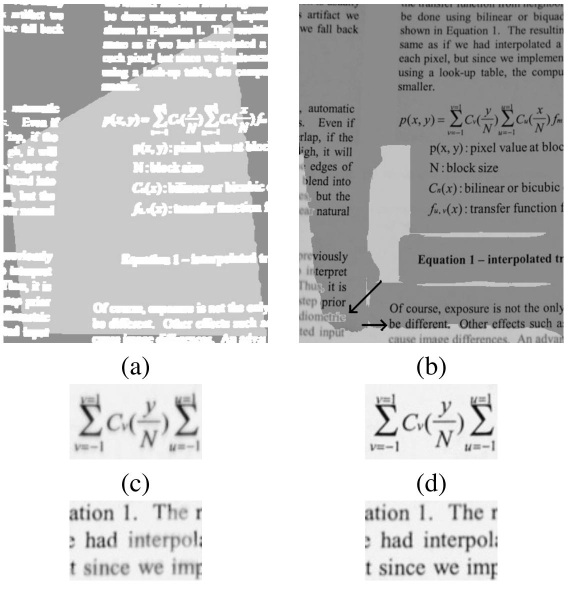


Figure 3. Image registration results. (a) Registration by correct PCA-SIFT matches shows misalignment. Squares and crosses indicate the matched points. (c) Registration by block matching results is very accurate.

The second problem is registration error and the third is different sharpness of patch images. Our solution to both of them is *selective image blending*, i.e., for each pixel, we choose the value from the image with better sharpness. We measure sharpness at a pixel by the local average of the gradient magnitude. For each pixel in the overlapped region, if it is 'sharper' in the first image, we say the pixel decision is '1', or '2' otherwise. If we apply pixel level decision directly, it may cut words or even characters into pieces. In practice we find that it is more desirable to keep each word whole. Therefore we aggregate the pixel decisions at the component level. More specifically, we compose an averaging image for the overlapped area, then binarize it and find its connected components. We dilate each component to cover a larger area. This ensures that areas that may contain 'ghost' images are merged into their nearest components. The dilation is made larger in the horizontal direction such that each word is more likely to become a single component. All the pixels inside a component vote with their pixel level decision, and the majority vote is taken as the component decision. If the decision is '1', every pixel in the component is copied from image 1; otherwise, they are all taken from image 2. As a result, most whole words are taken from the image that has the sharpest version of that word. For the background pixels, there is no danger of 'ghost', so we use the pixel level decisions directly.

Fig. 4 illustrates the process of selective image blending and the results. In Fig. 4(a), most components are single words. In Fig. 4(b) the two arrows indicate how the component level decision avoids cutting words. A large area of light gray is embedded in the dark gray in Fig. 4(b), which is fine since that is in the white background area, so any decision will do. The comparison between (c) and (d), as well as between (e) and (f) shows that sharpness is preserved. Also note that boundaries between mosaiced images are eliminated.



(a) (b)

(c) (d)

(e) (f)

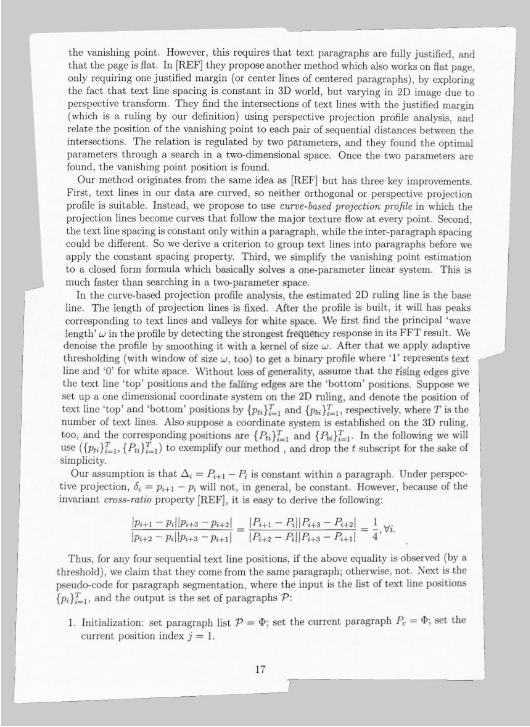
Figure 4. Selective image blending. (a) Connected component map where letters are dilated and connected. The overlapped region is in light gray, background in dark gray, and components in white. (b) The binary selection decision map distinguished by dark and light gray. (c,e) Weighted averaging result. (d,f) Selective image blending result.

4. Summary

We have implemented a framework for mosaicing camera-captured document images to reconstruct a full page image. Our two-step image registration method can align document images with as little as 10% overlap and severe perspective distortion. We also propose an image blending method that is optimized for document images, which addresses the inconsistent lighting, 'ghost' image, and varying sharpness problems.

We have applied our algorithm in full A4 page document mosaicing experiments. One of the results is shown in Fig. 5. The number of patches in our tests varies from four to eight. Due to limited space we cannot show other results here. In all cases, the registration is accurate, and the selective blending creates smooth and seamless results.

In future work we will test our algorithms with curved document images. We would anticipate some local misalignment beyond projective transformation, which we can rectify by local warping.



the vanishing point. However, this requires that text paragraphs are fully justified, and that the page is flat. In [REF] they propose another method which also works on flat page, only requiring one justified margin (or center lines of centered paragraphs), by exploring the fact that text line spacing is constant in 3D world, but varying in 2D image due to perspective transform. They find the intersections of text lines with the justified margin (which is a ruling by our definition) using perspective projection profile analysis, and relate the position of the vanishing point to each pair of sequential distances between the intersections. The relation is regulated by two parameters, and they found the optimal parameters through a search in a two-dimensional space. Once the two parameters are found, the vanishing point position is found.

Our method originates from the same idea as [REF] but has three key improvements. First, text lines in our data are curved, so neither orthogonal or perspective projection profile is suitable. Instead, we propose to use *curve-based projection profile* in which the projection lines become curves that follow the major texture flow at every point. Second, the text line spacing is constant only within a paragraph, while the inter-paragraph spacing could be different. So we derive a criterion to group text lines into paragraphs before we apply the constant spacing property. Third, we simplify the vanishing point estimation to a closed form formula which basically solves a one-parameter linear system. This is much faster than searching in a two-parameter space.

In the curve-based projection profile analysis, the estimated 2D ruling line is the base line. The length of projection lines is fixed. After the profile is built, it will has peaks corresponding to text lines and valleys for white space. We first find the principal 'wave length' ω in the profile by detecting the strongest frequency response in its FFT result. We denoise the profile by smoothing it with a kernel of size ω. After that we apply adaptive thresholding (with window of size ω, too) to get a binary profile where '1' represents text line and '0' for white space. Without loss of generality, assume that the rising edges give the text line 'top' positions and the falling edges are the 'bottom' positions. Suppose we set up a one dimensional coordinate system on the 2D ruling, and denote the position of text line 'top' and 'bottom' positions by $\{p_{ti}\}_{i=1}^{T}$ and $\{p_{bi}\}_{i=1}^{T}$, respectively, where T is the number of text lines. Also suppose a coordinate system is established on the 3D ruling, too, and the corresponding positions are $\{P_{ti}\}_{i=1}^{T}$ and $\{P_{bi}\}_{i=1}^{T}$. In the following we will use $(\{p_{ti}\}_{i=1}^{T}, \{P_{ti}\}_{i=1}^{T})$ to exemplify our method , and drop the t subscript for the sake of simplicity.

Our assumption is that $\Delta_i = P_{i+1} - P_i$ is constant within a paragraph. Under perspective projection, $\delta_i = p_{i+1} - p_i$ will not, in general, be constant. However, because of the invariant *cross-ratio* property [REF], it is easy to derive the following:

$$\frac{|p_{i+1} - p_i||p_{i+3} - p_{i+2}|}{|p_{i+2} - p_i||p_{i+3} - p_{i+1}|} = \frac{|P_{i+1} - P_i||P_{i+3} - P_{i+2}|}{|P_{i+2} - P_i||P_{i+3} - P_{i+1}|} = \frac{1}{4}, \forall i.$$

Thus, for any four sequential text line positions, if the above equality is observed (by a threshold), we claim that they come from the same paragraph; otherwise, not. Next is the pseudo-code for paragraph segmentation, where the input is the list of text line positions $\{p_i\}_{i=1}^{T}$, and the output is the set of paragraphs $\mathcal{P}$:

1. Initialization: set paragraph list $\mathcal{P} = \Phi$; set the current paragraph $P_c = \Phi$; set the current position index $j = 1$.

17

Figure 5. Full A4 page mosaic result.

References

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

[2] 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.

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

[4] J. Liang, D. DeMenthon, and D. Doermann. Flattening curved documents in images. In *Proc. CVPR*, pages 338–345, 2005.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

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