Dewarping Book Page Spreads Captured with a Mobile Phone Camera

Chelhwon Kim
Electrical Engineering Department
University of California, Santa Cruz
Santa Cruz, CA, US
chkim@soe.ucsc.edu

Patrick Chiu, Surendar Chandra
FX Palo Alto Laboratory
Palo Alto, CA, US
chiu@fxpal.com, chandra@fxpal.com

***Abstract*—Capturing book images is more convenient with a mobile phone camera than with more specialized flat-bed scanners or 3D capture devices. We built an application for the iPhone 4S that captures a sequence of hi-res (8 MP) images of a page spread as the user sweeps the device across the book. To do the 3D dewarping, we implemented two algorithms: optical flow (OF) and structure from motion (SfM). Making further use of the image sequence, we examined the potential of multi-frame OCR. Preliminary evaluation on a small set of data shows that OF and SfM had comparable OCR performance for both single-frame and multi-frame techniques, and that multi-frame was substantially better than single-frame. The computation time was much less for OF than for SfM.**

Keywords-document capture, document analysis, dewarping, mobile phone camera, book scanning

I. INTRODUCTION

Using portable devices to capture images of documents is a fast and convenient way to "scan" documents. Being able to use the compact capture device on-site is an important benefit in many scenarios. For example, students can use them to copy pages from books in a library, without potentially damaging the book spines when copying with a flat-bed copier. Another example is the digitization of documents in storage, in which bounded or loose paper records are often in too poor a condition to be used with flat-bed or V-bed book scanners without damaging them.

Compared with the results produced by flatbed scanners, these photos of documents taken with portable devices suffer from various issues including perspective distortion, warping, uneven lighting, etc. These defects are visually unpleasant and are impediments to OCR (optical character recognition). This paper focuses on the problem of dewarping page spread images of a book captured by a hi-res mobile phone camera.

We built an app for the iPhone 4S, which has an excellent camera, to capture a sequence of frames (8 MP, 2 fps). To capture a page spread, the user simply sweeps the device across the open book, similar to taking a video (see Fig. 1). From the sequence of frame images, we estimate the 3D information. We have implemented both optical flow (OF) and structure from motion (SfM) algorithms. The output of this step is a disparity map which encodes the depth information. Then we leverage the dewarping module in our previous system (where the disparity map was obtained from a stereo camera) [7]. This dewarping algorithm uses a 3D cylindrical model. An overview of the pipeline is illustrated in Fig. 2.

Making further use of the sequence of frame images, we consider a multi-frame OCR approach to improve the OCR performance. The idea is based on the observation that the left and right pages may be in better focus and not cropped off in different frames as the phone camera sweeps across the page spread at a non-uniform velocity.

We performed a preliminary evaluation to compare the OF and SfM algorithms in terms of OCR performance and computation time. We also compared multi-frame OCR with single-frame OCR using the middle frame image to see whether the improvement is substantial. The results are reported in detail below.

II. RELATED WORK

Existing research systems have been developed that relies on special 3D cameras or mounting hardware. The Decapod system [15] uses two regular cameras with special mounting hardware. Our previous system [7] uses a consumer-grade

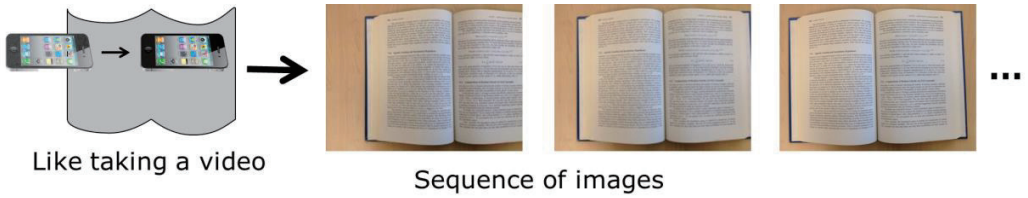


Fig. 1. Capturing a page spread of a book.

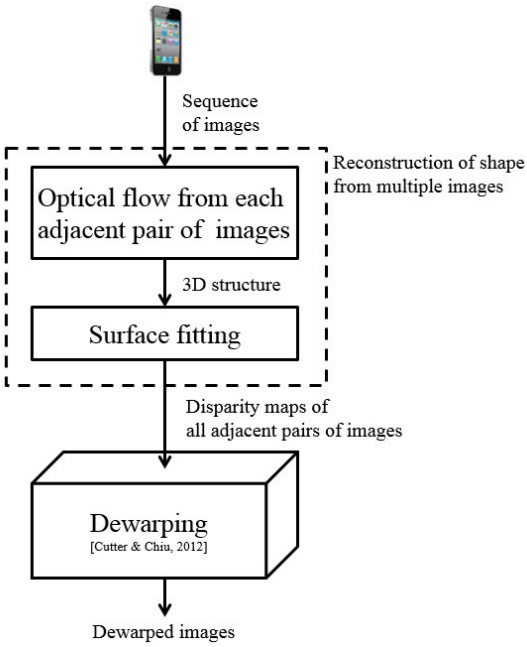


Fig. 2. Pipeline of system.

compact 3D stereo camera (Fujifilm Finepix W3). The dewarping method in our system is based on a cylindrical model, which for non-3D images performed the best (though the difference was not statistically significant) in the *Document Image Dewarping Contest at CBDAR 2007* (see [8], [14]).

Other 3D capture devices include structured light, which can sense highly accurate 3D information but requires more complicated apparatus. An example system is [4].

While it is possible to dewarp a book page image from a single photo taken with a non-3D device, the techniques to compute the 3D information are more specialized. Approaches include detecting content features like curved text lines or page boundaries and then applying a 3D geometric model to dewarp the image (e.g. [5], [6], [8], [9]).

Using video to capture documents is perhaps the approach that is the most related to our present work. With standard video formats, the frame image resolution is limited (VGA at 0.3 MP, HD at 2 MP) and performing OCR is problematic. In contrast, our app captures frames at much higher resolution (8 MP).

An early system, Xerox XRCE CamWorks ([11], [18]), has a video camera mounted over a desk to capture text segments from flat documents. It applied super-resolution techniques and OCR was evaluated on simulated images but not on actual camera images.

The NEC system [10] uses a VGA webcam and a mobile PC to capture video of a flat document or a curved book page spread. The user sweeps over the document in a back-and-forth path in order to cover the document and an image mosaicing method is applied to reconstruct an image of the whole document. The mosaicing uses a structure from motion algorithm that tracks Harris corner feature points. OCR was not performed nor evaluated.

Our system also uses a structure from motion algorithm that tracks "Good Features To Track" (GFTT) feature points [16]. In addition, we implemented a simpler optical flow algorithm. The high resolution allows us to use optical flow because a single sweep can capture the whole image and mosaicing is not need. Mosaicing requires a global coordinate system that SfM computes but OF does not. With OF, it suffices that only adjacent pairs of frames share a consistent coordinate system.

III. Computing and Dewarping the 3D Structure

We proceed to describe our implementation of two methods to compute the 3D structure: optical flow (OF) and structure from motion (SfM). In both, the features that are tracked are GFTT feature points [16]. Another option for feature points is the popular SIFT points; however SIFT points are not specifically designed to be tracked like the GFTT points. We also perform camera calibration to model the camera's geometry and correct for the lens distortions, which depends on the individual iPhone 4S device. The algorithms for GFTT and camera calibration are available in the OpenCV [3] computer vision library. The output of these OF and SfM methods is a disparity map that encodes the depth information, which are then fed into the dewarping module in the pipeline (see Fig. 2).

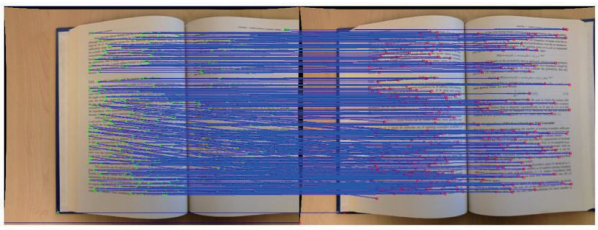


Fig. 3. Identifying corresponding feature points.

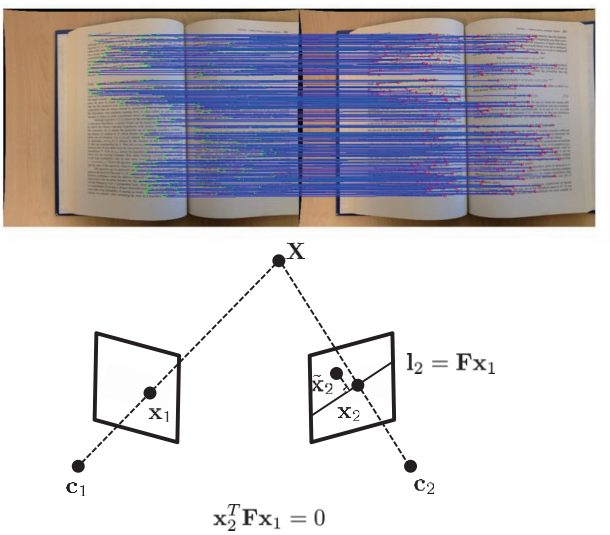


Fig. 4. Removing outliers using epipolar geometry.

A. *Optical Flow*

First, for each pair of sequential frame images, the corresponding feature points are matched. An example is shown in Fig. 3.

Next, the outliers are removed using epipolar geometry between two frames, which is described in the following equation

$$\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1 = 0,$$

where $\mathbf{F}$ is the fundamental matrix, $\mathbf{x}_1$ and $\mathbf{x}_2$ are homogeneous coordinates of the projected points of 3D point $\mathbf{X}$ onto the first and second image plane respectively. From this equation, we can map $\mathbf{x}_1$ to a line $\mathbf{l}_2 = \mathbf{F}\mathbf{x}_1$ in the second image. In other words, the projected point $\mathbf{x}_2$ on the second image plane always lies on the line. However, we cannot guarantee that all pairs of corresponding feature points satisfy this epipolar constraint due to noise in the image measurements and error in the optical flow matching method.

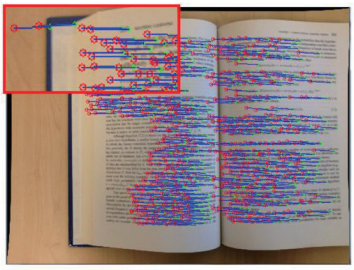


Fig. 5. Optical flow disparities (upper-left corner shows a closeup).

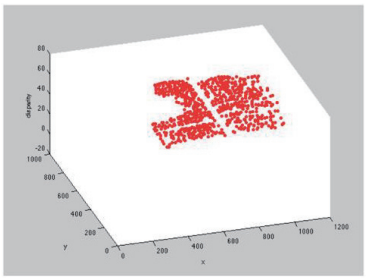


Fig. 6. Recovering shape information.

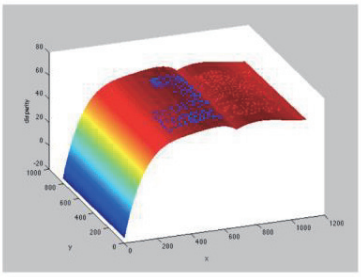


Fig. 7. Surface fitting.

Fig. 8. Disparity map with document region localized.

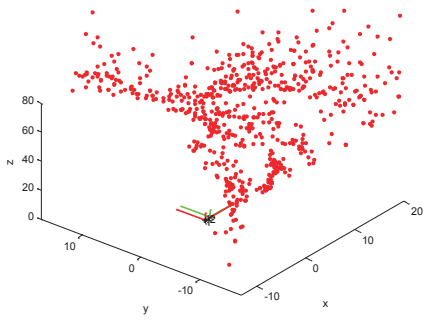
Therefore, to identify outliers among them, we calculate the orthogonal distance from the matching point in the second image, $\tilde{\mathbf{x}}_2$ to $\mathbf{l}_2$ (see Fig. 4), and if the distance is beyond a certain threshold then the pair of corresponding points is considered as an outlier. Fig. 4 shows the remaining inliers.

Computing disparities from optical flow is accomplished by looking at the displacements of the tracked feature points. The points on the book page spread at different depths will have different displacements (Fig. 5), and these disparities can be used to recover the shape of the page spread (see Fig. 6). Each dot in Fig. 6 represents a pair of corresponding points in the 3D space, where (x, y) are the image coordinates of the feature point in the first image, and z is the displacement of the tracked feature point in the second image with respect to the corresponding feature point in the first image. The recovered 3D points are clustered into two groups on each page; currently this is done manually by labeling the location of the book spine. This process can be automated by applying a clustering algorithm. A surface model is fitted to each cluster of 3D points using a 4-th order polynomial equation. See Fig. 7.

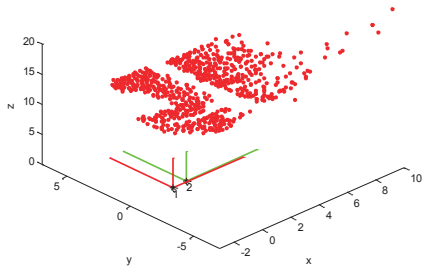
From this surface model, a disparity map is generated by mapping the depth (z-coordinate) to a grayscale value. Finally, the document region is localized within the image using an image segmentation algorithm; a good algorithm is GrabCut [13], which is available in OpenCV. In order to apply GrabCut, some background pixels must be identified and one way to do this is to sample pixels around the edge of the image and eliminate those that are similar to the center area of the image. An example of the resulting disparity map is shown in Fig. 8.

B. *Structure from Motion*

The first step is to initialize the 3D structure and camera motion from two sequential frames as follows: we first set the first camera matrix $\mathbf{P}_1=\mathbf{K}[\mathbf{I}_{3x3}|\mathbf{0}_{3x1}]$ to be aligned with the world coordinate frame, where $\mathbf{K}$ is the camera calibration matrix. Next, we identify the corresponding points between those two frames and estimate the fundamental matrix $\mathbf{F}$ using RANSAC algorithm. This is available in OpenCV library. The fundamental matrix is used to remove outliers as described above. Then, the essential matrix is computed by $\mathbf{E}=\mathbf{K}^T\mathbf{F}\mathbf{K}$. Once we have determined the essential matrix, we can recover the camera pose (rotation $\mathbf{R}$ and translation $\mathbf{t}$) for the second frame with respect to the first camera frame [17]. Then $\mathbf{P}_2$, the camera matrix for the second frame, can be easily obtained by multiplying the camera calibration matrix $\mathbf{K}$ by the camera pose for the second frame $[\mathbf{R}|\mathbf{t}]$. Lastly, we

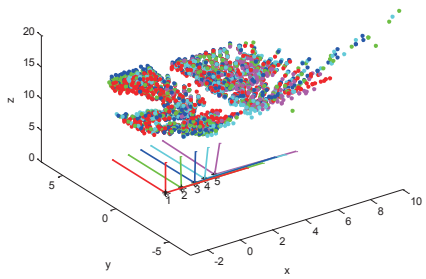


(a) ill-conditioned structure

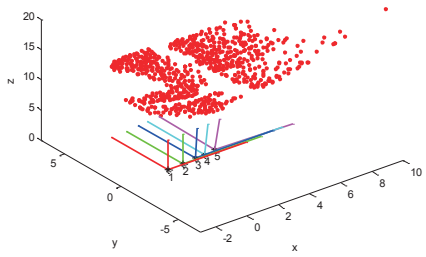


(b) well-conditioned structure

Fig. 9. Initial 3D structure.



(a) 3D structures for 5 frames



(b) combined 3D structure

Fig. 10. Structure from motion: after 5 frames.

estimate the 3D point structure from the 2D corresponding points and $\mathbf{P}_2$ through triangulation [17].

In practice, the algorithm for the fundamental matrix might not produce a well-conditioned initial 3D structure due to noise in the image measurements. Therefore, we add a step to reject ill-conditioned structures. An example of an ill-conditioned initial 3D structure is shown in Fig. 9a. The criterion of rejection is based on the prior knowledge that the shape of a book spread page is almost always two slightly curved surfaces that are not too far from a plane. Therefore, we first detect a dominant plane using RANSAC from the generated 3D structure, and then calculate the orthogonal distance for each 3D point to the plane. If the average distance is less than a predefined threshold then we accept the pair of frames, or reject it and check the next pair of frames. The threshold can be fixed under an assumption that the distance between the camera and the target is almost consistent across different users. Fig. 9b shows a well-conditioned 3D structure from the selected pair of frames.

An alternative method for computing the fundamental matrix is to use a non-linear optimization technique (e.g. [1]). This might improve the accuracy of the camera pose, but it requires more complicated processing.

Now we have an initial 3D point structure and consider how to use a new frame to update it. Let us assume that the 3D point structure for (i-1)-th frame is already known and we have tracked the existing corresponding points from the (i-1)-th frame to the i-th frame. As we described above, we remove outliers from the tracked points using epipolar geometry. The remaining tracked points and the corresponding 3D points are used to estimate the new camera pose for i-th image $\mathbf{P}_i$ by minimizing the projection error $e = \sum_j \left\| \mathbf{x}_j^{(i)} - \mathbf{P}_i \mathbf{X}_j \right\|^2$, where $\mathbf{x}_j^{(i)}$ is the j-th tracked 2D point in the i-th image and $\mathbf{X}_j$ is the corresponding j-th 3D point. Given this estimated camera matrix $\mathbf{P}_i$ and the tracked points in the i-th frame, we recalculate the 3D point structure through triangulation. We iterate the above process throughout the sequence of frames. Fig. 10a shows the 3D point structures for each iteration and camera pose frames with different colors. To get a single 3D structure from all the frames' 3D structures, we combined them by simple averaging (Fig. 10b). The final 3D structure still has outliers as can be seen from the right most corner of the structure in Fig. 10b. In order to deal with this, we perform the surface fitting algorithm with RANSAC.

From the surface model, a disparity map is generated for each frame as described above in the optical flow method.

Another option for combining all the 3D structures is to use bundle adjustment (e.g. [20]). The advantage is that it might improve the accuracy of the camera poses and the 3D structures. Since in our application, the camera motion is very simple (basically linear), the improvement may be small. The disadvantage of using bundle adjustment is that it requires more processing.

C. Cylindrical Model

For completeness, we give a brief summary of how the cylindrical model is used with the disparity map to do the dewarping; for more details refer to [7]. First, from a disparity map, two depth profiles perpendicular to the spine are extracted from the top and bottom halves of the page spread by averaging over their respective halves. These profiles form the skeleton of the cylindrical model. To facilitate the rendering of the dewarped image, rectangular meshes are employed. A mesh vertex point on the cylindrical

model can be mapped to a vertex point in the dewarped image by flattening it using its arclength along the cylindrical surface to push it down and outward from the spine. Points inside each mesh rectangle are then interpolated based on the rectangle's vertices.

IV. Multi-frame OCR

By *single-frame OCR*, we mean using one frame to OCR the left and right pages of a page spread. Typically, the middle frame in the sequence of frame images can be used, because both pages of the book spread are usually in view with the camera held in landscape orientation.

By *multi-frame OCR*, we mean using more than one frame for doing the OCR. The idea is that the left page is more likely to be better captured in the early frames and the right page in the later frames. Some frames may also be in better focus than others.

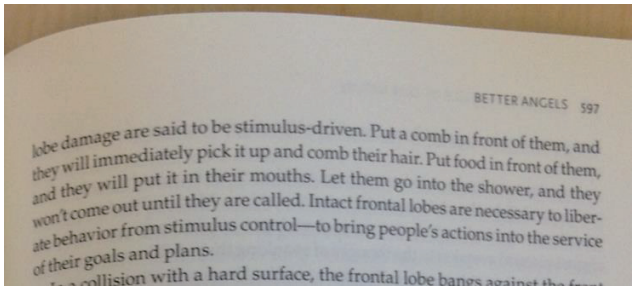
To study the potential of multi-frame OCR, we compared the best OCR scores for the left and right pages over multiple frames to the OCR scores of the middle frame. These results are reported below.

For single-frame OCR and multi-frame OCR, a separate condition is whether the frame images have been dewarped.

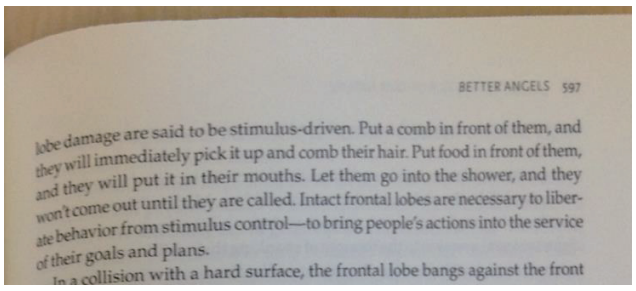
V. Preliminary Evaluation

To compare OF vs. SfM, non-dewarped vs. dewarped, and single-frame vs. multi-frame, we did a preliminary evaluation on a small set of data based on OCR.

Six images of book page spreads were taken with our app on an iPhone 4S camera. The device was handheld (a tripod was not used). The frame image resolution was 8 MP (3264 x 2448). The frame rate used was about 1 fps; we found this to work fine for our processing pipeline even though the frame rate can go as high as 2 fps when capturing 8 MP images.



(a) before dewarping



(b) after dewarping

Fig. 11. Example of a dewarped page spread with the top-right region shown.

Our mobile phone app was implemented in Objective-C, and the code for processing the frame images was implemented in C++ and uses the OpenCV library [3]. The captured images were processed on a desktop PC.

We examined the boundary text lines on the two pages in each page spread: {top-left, top-right, bottom-left, bottom-right}. By a "boundary text line", we mean the text line nearest to the top or bottom of a page that spans more than half the body of the page, so that short lines at the bottom of a paragraph and headers or footers are not considered. The 6 page spreads provides a total of 24 boundary text-lines.

An example of a dewarped page spread is shown in Fig. 11. The frame is the middle frame of the image sequence. The method is OF. The bottom image in the figure is a closeup of the top-right region of the page spread showing several text lines that have been dewarped. There is some inaccuracy near the spine of the book, which is a difficult area to handle due to the steepness of the page and the lack of content for tracking.

For OCR, we use the open-source Tesseract OCR engine [19]. To measure the difference between two text strings, we use edit distance (Levenshtein distance), normalized by dividing by the length of the ground-truth string.

The left and right pages were manually cropped from the images, and each page was processed through the OCR engine. Then the top and bottom boundary text line characters were extracted and the edit distances were computed.

For the single-frame condition, we used the middle frame in the image sequence. For the multi-frame condition, we used the frames at the beginning, middle, and end of the image sequence.

The OCR results show that dewarped was better than non-dewarped, with substantial improvement for multi-frame over single-frame. See Fig. 12. OF and SfM had similar performance for both single-frame and multi-frame. In terms of processing time for computing the 3D structure, OF was much faster than SfM (more than 2x).

VI. Conclusion and Future Work

We presented an application to capture page spread images with a mobile phone, and a processing pipeline that uses either OF or SfM to compute the 3D information along with a cylindrical model to perform dewarping. Our preliminary evaluation indicates that OF might be a better choice than SfM since they had similar OCR performance but OF was much faster. This could be important in future systems when the frame images are processed on the mobile phone.

Another aspect that could be improved in the future is to mitigate the motion blur caused by the sweeping motion of the camera when the user takes the sequence of images. This is somewhat noticeable in the images in Fig. 11. One way to address the blur problem is to apply deconvolution algorithms, which is an active area of research (e.g. [21]). Improvements in mobile phone cameras such as faster lens

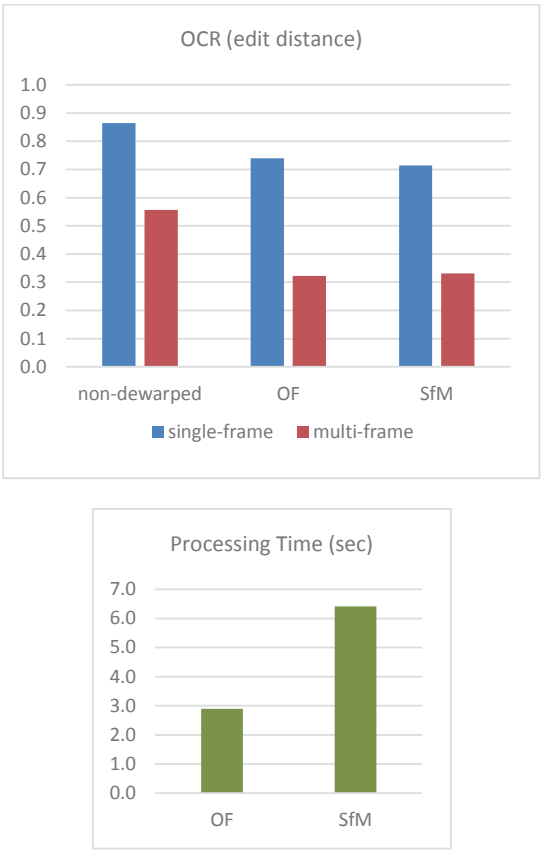


Fig. 12. OCR and processing time results.

and more reliable autofocus systems will also lessen the blurriness.

Other future work includes automating some of the steps in the pipeline. For example, page frame detection algorithms (e.g. [2]) can be applied to crop the left and right pages from the page spread. Image quality assessment algorithms (e.g. [12]) can be applied to select the frames that are likely to produce the best OCR results.

ACKNOWLEDGMENT

This work was done at FX Palo Alto Laboratory. We thank Michael Cutter and David Lee for helpful discussions.

REFERENCES

[1] P. Beardsley, A. Zisserman, D. Murray. "Sequential Updating of Projective and Affine Structure from Motion," *Intl. J. of Computer Vision* (23), No. 3, Jun-Jul 1997, pp. 235-259.

[2] S. Bukhari, F. Shafait, T. Breuel. Border noise removal of camera-captured document images using page-frame detection. *Proc. CBDAR 2011*, pp. 102-107.

[3] G. Bradski. "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[4] M. Brown, W. Seales. "Image Restoration of Arbitrarily Warped Documents," *IEEE TPAMI*, vol. 26, Oct. 2004, pp. 1295-1306.

[5] M. Brown, Y.-C. Tsoi. "Geometric and shading correction for images of printed materials using boundary," *IEEE Trans. Image Processing*, vol. 15, Jun. 2006, pp. 1544-1554.

[6] H. Cao, X. Ding, C. Liu. Rectifying the bound document image captured by the camera: A model based approach. *Proc. ICDAR 2003*, pp. 71-75

[7] M. Cutter, P. Chiu. Capture and dewarping of page spreads with a handheld compact 3D camera. *Proc. DAS 2012*, pp. 205-209.

[8] B. Fu, M. Wu, R. Li, W. Li, Z. Xu, C. Yang. A model-based book dewarping method using text line detection. *Proc. CBDAR 2007*, pp. 63-70.

[9] J. Liang, D. DeMenthon, D. Doermann. "Geometric rectification of camera-captured document images," *IEEE TPAMI*, vol. 30, Apr. 2008, pp. 591-605.

[10] N. Nakajima, A. Iketani, T. Sato, S. Ikeda, M. Kanbara, N. Yokoya. Video mosaicing for document imaging. *Proc. CBDAR 2007*, pp. 171-178.

[11] W. Newman, C. Dance, A. Taylor, S. Taylor, M. Taylor, T. Aldhous. CamWorks: a video-based tool for efficient capture from paper source documents. *Proc. Intl. Conf. on Multimedia Computing and Systems (ICMCS 1999)*, pp 647-653.

[12] X. Peng, H. Cao, K. Subramanian, R. Prasad, P. Natarajan. Automated image quality assessment for camera-captured OCR. *Proc. ICIP 2011*, pp. 2669-2672.

[13] C. Rother, V. Kolmogorov, A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. *Proc. Siggraph 2004*, pp. 309-314.

[14] F. Shafait, T. Breuel. Document image dewarping contest, *CBDAR 2007*.

[15] F. Shafait, M. Cutter, J. van Beusekom, S. Bukhari, T. Breuel. Decapod: a flexible, low cost digitization solution for small and medium archives. *Proc. CBDAR 2011*, pp. 41-46.

[16] J. Shi, C. Tomasi. Good features to track. *Proc. CVPR 1994*, pp. 593-600.

[17] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.

[18] M. Taylor, C. Dance. Enhancement of document images from cameras. *SPIE Conf. on Document Recognition V*, 3305, 1998, pp. 230-241.

[19] Tesseract OCR. http://code.google.com/p/tesseract-ocr.

[20] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon. Bundle adjustment – a modern synthesis. *Proc. ICCV 1999*, pp. 298-372.

[21] L. Xu, J. Jia. Two-phase kernel estimation for robust motion deblurring. *Proc. ECCV 2010*, pp. 157-170.