High-Quality Capture of Documents on a Cluttered Tabletop with a 4K Video Camera

Chelhwon Kim
University of California, Santa Cruz
Santa Cruz, CA, USA
chkim@soe.ucsc.edu

Patrick Chiu
FXPAL
Palo Alto, CA, USA
chiu@fxpal.com

Henry Tang
FXPAL
Palo Alto, CA, USA
tang@fxpal.com

ABSTRACT
We present a novel system for detecting and capturing paper documents on a tabletop using a 4K video camera mounted overhead on pan-tilt servos. Our automated system first finds paper documents on a cluttered tabletop based on a text probability map, and then takes a sequence of high-resolution frames of the located document to reconstruct a high quality and fronto-parallel document page image. The quality of the resulting images enables OCR processing on the whole page. We performed a preliminary evaluation on a small set of 10 document pages and our proposed system achieved 98% accuracy with the open source Tesseract OCR engine.

Categories and Subject Descriptors
I.7.5 [**Document and Text Processing**]: Document Capture – *document analysis, Optical character recognition (OCR), scanning.*

Keywords
Document capture; tabletop system; computer vision; reconstruction; image processing; video processing; OCR

1. INTRODUCTION
In multimedia communication and collaboration, people desire to exchange and interact with physical objects such as paper documents as they would with their digital counterparts. Tabletops augmented with video cameras or projectors is a way to bridge the physical and digital worlds. In particular, research systems for working with paper documents have been developed over the years (e.g. [11], [6], [7]). These systems dealt with problems such as user interaction, selection of a small region for capture, and document page tracking. However, the problem of capturing a whole document page located anywhere on a cluttered tabletop has not be adequately addressed. The ability to provide a high-quality image of a page would enable a more seamless transition between the physical and digital worlds.

In the digital world, OCR (Optical Character Recognition) is a powerful tool for analysis and understanding of the document content. Having high-quality images is essential for OCR. This allows one media (text) to be extracted from another (video). With the text extracted via OCR, there are many possible applications including: indexing & search systems, document repositories, and language translation.

DocEng '15, September 08 - 11, 2015, Lausanne, Switzerland

© 2015 ACM. ISBN 978-1-4503-3307-8/15/09…$15.00

DOI: http://dx.doi.org/10.1145/2682571.2797074

One limitation of previous systems is that they used low resolution video cameras. For example, CamWorks [11] and FACT [7] used VGA or HD video cameras (640 x 480 and 960 x 720 respectively), which can capture only small parts of the document page at high resolution. With recent technology like 4K video cameras (4096 x 2160), it is more feasible to tackle the problem since a document page (8.5" x 11") at 300 dpi (a standard resolution for OCR) has similar pixel resolution (3300 x 2250).

Even with sufficient resolution, a captured image may suffer from noise and geometric distortion. We present some methods to improve the image reconstruction. In particular, we developed a multi-frame stop-motion capture technique along with an image fusion process.

To cover the entire tabletop, the camera must be able to point at any part of the surface. We mount the camera overhead on a robot turret which has pan and tilt servos. It is automatically controlled to sweep over the surface for document detection and capture.

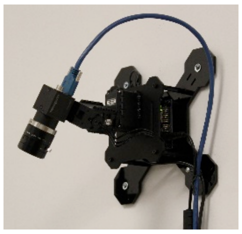
For more realistic scenarios, the document pages must be detected on a tabletop that is cluttered with other objects (e.g. coffee mugs, pens, etc.). To deal with this, we use on a text probability map computed from a training set of document images.

In this paper, we propose a system that employs a two-stage approach for detection and high-quality capture of paper documents placed anywhere on a cluttered tabletop. We evaluated our system using an open source OCR Engine (Tesseract [14]) and obtained 98% accuracy.

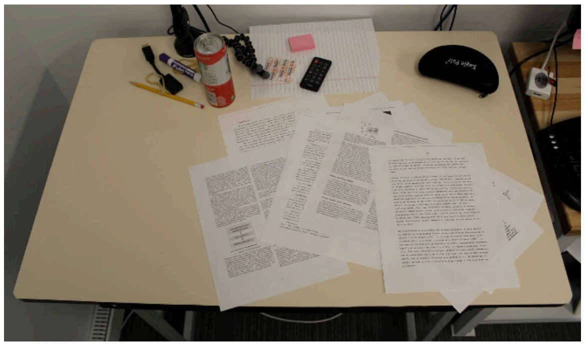
2. RELATED WORK
CamWorks [11] is a research system that employs a video camera mounted over an author's desk. It supports capturing text segments on the page selected by the user. The user interface shows the video image of a page of the document, where the user can select material to be copied. The small regions selected are captured with a very low-resolution video camera (640x480).

The FACT system [7] used a video camera (960 x 720) to track a pen tip for user interaction with the contents of a document page. It does not capture high quality document images, instead it takes a different approach by linking a paper document in the camera view to an online version of the document. The link is determined by matching feature points on images of the document pages.

A video based document tracking system [6] was developed to track documents on a desktop surface. It is able to identify the document page objects by detecting when the page is brought into the camera's view and placed on the desk. The camera resolution was 1024x768 and OCR was not performed.



(a)



(b)

Figure 1. System setup: (a) the camera is attached to a robot turret mounted above the table, (b) tabletop with documents and cluttered with other objects.

In summary, none of the existing video camera based tabletop systems can cover an entire table surface and capture a full document page at sufficient quality for OCR.

3. SYSTEM OVERVIEW

We installed a 4K camera (Point Grey Flea 3) mounted on a pan-tilt robot turret (with two Dynamixel servos) above a table to look for documents placed anywhere on the cluttered tabletop. See Figure 1. The camera has resolution 4096 x 2160. The distance of camera from the center of the table is determined so that we have a 300 dpi image with approximately 20 pixels x-height to achieve optimal OCR performance.

Our proposed system has two stages. First, the system scans the tabletop to detect text areas by moving the camera along a predefined path while it stops and captures a part of the tabletop surface at grid positions. Figure 3a shows the captured images. These tabletop images are converted into a text probability map by using a pre-computed text probability histogram and an image-stitching method based on known camera orientations recorded at the positions (Figure 2b). This gives text blobs which indicate the locations of document papers on the tabletop surface.

The second stage is to capture each document page. From the detected text blobs, the system generates scan paths (blue lines in Figure 2b) whereon the camera's viewpoint moves and captures a sequence of frames (Figure 2c). The camera uses a stop-motion technique: it stops moving when it captures a frame. The captured frames are fused together by a conventional image-stitching method (Figure 2d). Finally, the system corrects the perspective distortion in the fused document image by finding a quadrilateral formed by two horizontal lines along the text lines and two vertical lines along the vertical paragraph margins (Figure 2e).

4. TECHNICAL DETAILS

4.1 Document Paper Detection

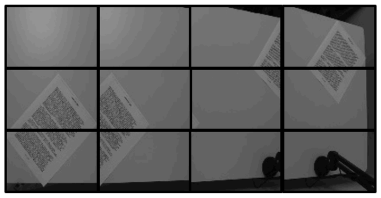
We use the "Bag of Keypoints" method [2] to detect text regions in the captured image. Instead of building a classifier for {text, non-text}, we compute a histogram of text occurrences based on SIFT keypoint features and use this to compute the probability map of a new image. More precisely, we first extract 128-dimensional SIFT descriptor vectors [8] from the document page images in a training set. Then we cluster these SIFT descriptor vectors into a set of visual words (i.e. clusters) using the hierarchical k-means clustering algorithm [10]. We compute a histogram of occurrences of visual words based on the number of SIFT descriptor vectors assigned to each cluster. The normalized histogram of occurrences of visual words in the training dataset serves as a probability model which provides the probability of text occurrences given the quantized image features in an image. This normalized histogram is pre-computed offline.

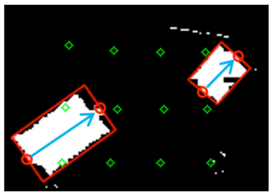
For each of the tabletop images (Figure 2a), we compute a text probability map based on the trained probability model and the SIFT feature points extracted from the image. Specifically, we assign a 2D Gaussian kernel density to each feature point location, weighted by the normalized frequency of the visual word corresponding to the feature descriptor. The corresponding visual word is determined by a fast k-Nearest Neighbor search algorithm [10]. Finally, we sum up all the Gaussian kernels to produce the final probability map.

All the computed text probability maps are stitched together by estimating relative shift for each tabletop image to one particular image based on homography induced by the relative camera orientations based on the pan-tilt parameters. The final binary map is computed by thresholding the stitched text probability map and applying a morphological filter (Figure 2b). The text blobs in the binary map are detected by the connected component labeling (red rotated rectangles in Figure 2b). As each text blob indicates the candidate location of a document paper on the tabletop, we define a camera scan path over the document paper by selecting the top and bottom center of the text blob (red circles) and generate a list of camera capture points on a line joining the two points (blue arrows). For each point, we recover the corresponding pan and tilt values (camera orientation) by interpolating (or extrapolating) the known nearby data points that are the locations of the aligned image tiles in the map (green circles in Figure 2b). Finally, a sequence of images captured at the list of points (Figure 2c) is processed to reconstruct a document image.

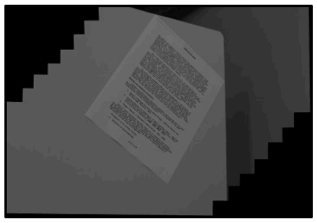
4.2 Reconstruction

The captured multiple document images are stitched together by the general image-stitching method described in [13]. We first align all images by estimating relative shift for each frame to one particular frame. Since the images contain sufficient textures on text regions we can use point feature based homography. Homography produces the relative shift of pixels with sub-pixel accuracy so the interpolation is necessary when the pixels in each frame are brought to the reference image pixel grid. The aligned pixels at the same grid pixel location of the reference image are combined by a simple arithmetic mean. Finally, a sharpening filter is applied to the combined image to remove blurring.


a. Scan tabletop (3 x 4 images) to find documents


b. Stitched text probability map and text blob detection

c. Capture a sequence of high-resolution document images


d. Reconstructed document image

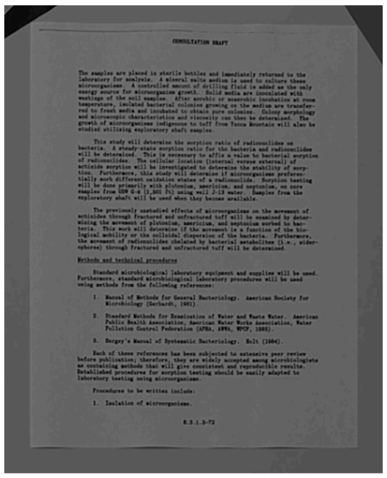

e. Rectified document image

Figure 2. Overview of process.

4.3 Perspective Rectification

The reconstructed image still has perspective distortion so we correct it and recover the fronto-parallel view of the document page. Our correction method finds homography between a source quadrilateral formed by two horizontal lines along the text lines and two vertical lines along the vertical paragraph margins and its corresponding target quadrilateral (upright rectangle) similar to [1] and [3].

We first detect line segments along the text lines by the probabilistic Hough transform algorithm [9] (red line segments in Figure 3a), and the end points of the detected line segments are used to estimate two vertical lines along the vertical paragraph margins using the RANSAC based line-fitting algorithm (See blue and green lines in Figure 3a.). We use these two fitted lines as the left and right sides of the source quadrilateral.

In order to get the top and bottom sides of the source quadrilateral, we find a rotated rectangle around a text blob detected by the same method described in the Section above on document paper detection. (See red rectangle in Figure 3b.) Then the line segments of top and bottom sides of the rotated rectangle are rotated around their midpoint and aligned with the direction from the midpoint to a vanishing point to which the text line segments converge. The vanishing point can be estimated by RANSAC-based method [5]. Finally, the four sides of the source quadrilateral are found with the fitted two vertical lines. (See blue quadrilateral in Figure 3b).

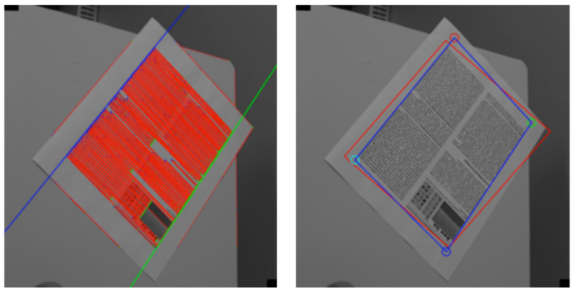

(a) (b)

Figure 3. Perspective distortion correction: (a) text line segments and two vertical lines, (b) source quadrilateral.

The target quadrilateral is estimated by using the width/height ratio of a back-projection of the source quadrilateral onto the table surface in space. To do this, we recover 3D coordinates of four vertexes of the source quadrilateral by finding intersecting points between back-projection rays of the vertexes and the table surface provided that the camera intrinsic matrix and the table surface orientation are estimated. After that, the width/height ratio is then a ratio between an average length of top and bottom sides and the one of left and right sides of the back-projected quadrilateral on the plane. The width is fixed to a certain number of pixels and the height is computed using the computed aspect ratio. After the four vertex correspondences of source and target quadrilateral are computed, the homography is obtained using the Direct Linear Transformation algorithm [4].

5. PROTOTYPE IMPLEMENTATION

The hardware and tabletop setup is described above (Section 3). The software is implemented in C++ and uses the OpenCV library [12]. On a desktop computer (Windows 7, Intel Core i7 3.6GHz CPU with 16GB RAM), execution time for the first stage of detecting documents takes 11 seconds on average. For the second stage of capture and reconstruction of a page, it takes 20 seconds for capturing and fusing the 8 images of resolution 4096x2160 and for correcting the perspective distortion in the fused image.

In the training of the text probability histogram (described in Section 4.1), we used scanned document images in the UNLV-ISR OCR dataset [15]. About 800K SIFT features were extracted from about 800 document page images. Figures were manually removed in the document images. To reduce computation time, we down-sampled the image tiles to 20% of the original size. The width of Gaussian kernel and threshold value used in the text probability map computation are empirically determined to smooth and to threshold the text probability values until the map produces at best one text blob per document.

6. EVALUATION

We performed a preliminary evaluation based on OCR on a small set of 10 document pages. Five of them are from the UNLV-ISR OCR dataset in one-column layout, and five are from papers published in IEEE format in two-column layout. These were printed out on paper using a laser printer.

The tabletop was cleared of cluttered objects so that the pages can be placed systematically on the surface. The pages were placed flat on the tabletop surface without rotation. Each of the 10 pages was placed at three different locations {0cm, 68cm, 92cm} from the center of the camera along the diagonal of the tabletop surface. With a typical large desk size of 60"x30", placing an 8.5"x11" piece of paper at a corner corresponds to the 68cm location; this is the location of maximum distortion on the desk surface. To see how the performance drops off at a farther distance, we tested it at the 92cm distance, which would be beyond a typical desk surface.

For the OCR, we use the open-source Tesseract OCR engine [14]. 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. Then we take the *OCR score* to be 1.0 minus the normalized edit distance (so that zero distance corresponds to 100% accuracy).

The OCR results are shown in Figure 4. Our system achieved 98% OCR accuracy and showed stable results over the three different locations. As a benchmark, we ran the Tesseract OCR engine on the document images obtained by a generic flatbed scanner, and the result was 99% accuracy. Comparing to the benchmark shows that our system, despite the geometric distortions and uncontrolled lighting conditions, can perform almost as well as a flatbed scanner.

We also evaluated a *single-frame* method without fusion. This only corrects the perspective distortion in a single image frame of document page using the same approach in the proposed system. However, since a single frame can cover only a small field of view of document page depending on its distance from the camera, we generate a virtual single frame with a larger field of view that covers a whole document page. We do this by concatenating two or three frames using the same stitching method as in the reconstruction step (Section 4.2), except that we take only a single pixel from the aligned pixels sitting on the same pixel location in the reference image instead of averaging them. For the sake of fair comparison, we applied the same sharpening filter to the concatenated image. The Single-frame method achieves 98% OCR performance for documents at the center of the table (0cm), but its performance drops down to 90% as the distance of document increases to the farthest location (92cm).

7. CONCLUSION & FUTURE WORK

We presented a novel system for capturing high-quality document page images on a cluttered tabletop. It goes beyond existing systems by providing sufficient resolution and quality for full-page OCR. Our preliminary evaluation shows that it can achieve a high rate of accuracy.

For future work, one direction is to explore continuous capture of the tabletop over a long period of time in a natural setting. This requires better handling of multiple document pages on the tabletop. Occlusion of the document pages by hands or other objects is another problem for further work.

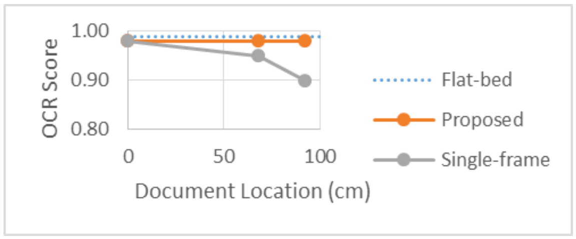


Figure 4. OCR performance.

8. REFERENCES

[1] Clark, P., Mirmhedi, M. Rectifying perspective views of text in 3D scenes using vanishing points, *Pattern Recognition,* 36: 2673–2686 (2003).

[2] Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C. Visual categorization with bags of keypoints. *Proc. of ECCV Intl. Workshop on Statistical Learning in Computer Vision* (2004).

[3] Dance, C.R. Perspective estimation for document images. *Proceedings SPIE Document Recognition IX (2002).*

[4] Hartley, R., Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press (2003).

[5] Hwangbo, M., Kanade, T. Visual-inertial uav attitude estimation using urban 643 scene regularities. *Robotics and Automation (ICRA 2011)*, pp. 2451–2458.

[6] Kim, J., Seitz, S., Agrawala, M. Video-based document tracking: Unifying your physical and electronic desktops. *Proceedings of UIST '04*, pp. 99-107.

[7] Liao, C., Tang, H., Liu, Q., Chiu, P., Chen, F. FACT: fine-grained cross-media interaction with documents via a portable hybrid paper-laptop interface. *Proceedings of ACM Multimedia '10*, pp. 361-370.

[8] Lowe, D.G. Distinctive image features from scale invariant keypoints. *Intl. Journal of Computer Vision*, 60: 91–110 (2004).

[9] Matas, J., Galambos, C., Kittler, J.V. Robust detection of lines using the progressive probabilistic Hough transform. *CVIU* 78 1, pp 119-137 (2000)

[10] Muja, M., Lowe, D.G. Fast approximate nearest neighbors with automatic algorithm configuration. *Intl. Conf. on Computer Vision Theory and Applications (VISAPP 2009).*

[11] Newman, W., Dance, C., Taylor, A., Taylor, S., Taylor, M., Aldhous, T. CamWorks: a video-based tool for efficient capture from paper source documents. *Proc. ICMCS '99*, pp 647–653.

[12] OpenCV. http://opencv.org/

[13] Szeliski, R. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis*. 2, 1 (Jan. 2006), 1-104.

[14] Tesseract. https://code.google.com/p/tesseract-ocr/

[15] UNLV-ISR OCR dataset. https://code.google.com/p/tesseract-ocr/wiki/TestingTesseract