Advances in Interactive Video Scanning of Paper Documents

Stuart Taylor, Chris Dance, William Newman, Alex Taylor,
Mauritius Seeger, Michael Taylor, Tony Aldhous
Xerox Research Centre Europe, 61 Regent Street,
Cambridge, CB2 1AB, United Kingdom
first.last@xrce.xerox.com, mjt@cre.canon.co.uk, Anthony.Aldhous@Smallworld.co.uk

Abstract

This paper describes the design and evaluation of CamWorks, a system that employs a video camera as a convenient means of capturing from paper sources during reading and writing. The user can view a live video image of the source document alongside the electronic document in preparation. We describe a novel user interface developed to support selection of text in the video window and several new techniques for segmentation, restoration and resolution enhancement of camera images. An evaluation shows substantially faster text capture than with flatbed scanning.

Introduction

People from virtually every branch of professional work author electronic documents while referring to paper documents such as books, articles and reports. Despite efforts to develop paperless authoring environments, there is little evidence that the use of paper is diminishing. Studies of authoring [Haas 1996, O'Hara and Sellen 1997] suggest that paper offers a degree of flexibility and ease of navigation that is not found in tools for on-line access to source material. It may be some time before these drawbacks of on-line tools are overcome: in the meantime authors need more efficient ways of working with paper sources. The *CamWorks* system described in this paper illustrates how efficient authoring can be achieved by using a live video image of the source, captured by a digital camera, displayed alongside the electronic document in preparation. Figure 1 shows the system in use.

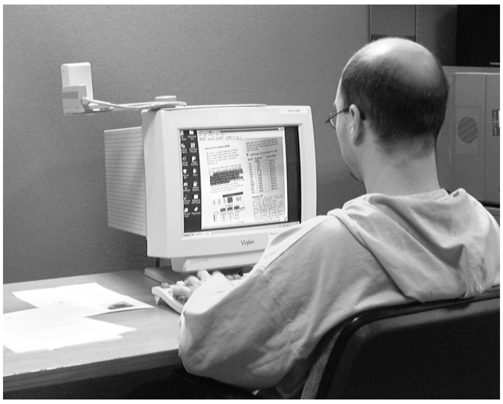


Figure 1: The CamWorks system in use

Previously suggested systems for using video cameras to capture document images such as the DigitalDesk [Wellner 1991, Wellner 1993] and its successors [Ishii and Kobayashi 1992, Ishii Kobayashi and Arita 1994] employ a projector to transform a desk surface into a space for manipulating paper and electronic documents. In contrast, CamWorks augments the electronic domain of word processing and spreadsheets with live video images of source documents. We believe this is a more practical and efficient way of using cameras to assist authors.

Previous studies of authoring clearly indicate that authoring tools must support the use of paper-based sources but they leave a number of questions unanswered that must be addressed in order to formulate requirements for the design of efficient authoring systems. We have therefore undertaken studies of our own in order to gain a better understanding of authors' needs for document capture tools. Our first study [O'Hara *et al* 1998] highlighted the need for efficient ways of capturing text while reading. The analysis of a follow-on study, which focused on the use of sources during writing, identified important requirements for the support of source material capture:

?? *Capture support is needed whether the author's focus is on reading or writing.* Authors need to be able to take notes while reading, and to make references to sources while composing text.

? *There is a need to capture small segments of text.* In some application domains, a large proportion of the segments copied are one sentence or less in length.

?? *Small-segment capture should not interrupt the flow of reading or writing.* Otherwise the author loses considerable time in returning to their place and re-establishing their context.

? *Accuracy of capture is important, but not always essential.* Accuracy may not be important at the early stages of authoring because the material may be discarded from later drafts.

Although present day scanning devices provide accurate document capture, they do not meet the other requirements sufficiently to enable them to serve as efficient authoring tools. The flatbed scanner is not widely used by authors since it provides a cumbersome interface for capturing small amounts of text: the document must be moved from the place of reading and the user must wait for a time-consuming pre-scan. Various handheld scanners have been developed, but they are usually too small to capture a wide stripe of the page, and so require several passes over the document.

We regard the present situation as unsatisfactory, but believe digital cameras can provide a route to a solution. In particular, cameras require no desk-space and can provide face-up real-time scanning, so supporting rapid capture from documents at the user's normal place of reading.

Despite the convenience of camera scanning, the quality of the resulting images tends to be far lower than with conventional scanning devices. This is largely because cameras function under much less constrained conditions and so the images are subject to spatially varying illumination, blur and geometric distortions. Although some of these conditions could be controlled by the use of additional hardware such as copy-stands, this would detract from the convenience of camera scanning. Our approach has therefore been to tackle these problems by applying image restoration techniques.

Another major source of poor image quality with today's digital video cameras is their low resolution. However, the resolution available from digital still cameras is steadily improving and the bandwidth necessary to use these resolutions for video is gradually becoming available. Despite ongoing increases in processor speeds, the expectation of increased resolutions coupled with the requirement of rapid user feedback has meant that all the image restoration procedures developed must be very efficient.

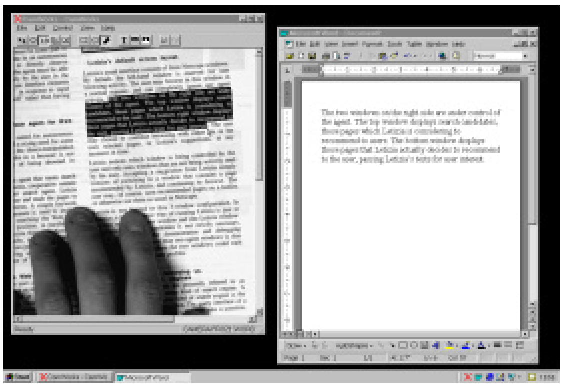
In the next section, we explain how the requirements that emerged from our studies influenced the design of the CamWorks user interface. We then describe the text segmentation techniques that support the selection of word sequences, and outline how the selected images are enhanced and binarised. An evaluation of CamWorks is presented, showing that it offers significant performance advantages over flatbed scanning.

The CamWorks User Interface

A video camera mounted over the desk can deliver an image of a paper document to the user, who can then quickly select and copy material to other electronic documents. There are, however, a number of aspects of the user interface that need to be addressed, including whether to use a live or static image, and what types of selection methods to support.

We experimented with both live and static video images, and found that a static image was often sufficient for selection purposes. However, we noted that considerable time was spent in positioning the document in relation to the scanning region of the device. A live video image could help the user in this positioning task, and thus reduce disruption. Our basic design strategy for CamWorks has therefore been to provide a continuously updated video image of the document under the camera.

Currently available scanning software typically restricts the user to selecting a rectangular region of the page, but short text segments often have non-rectangular outlines, such as that shown in Figure 2. We therefore provide *content-based* selection methods. The user can select a single word, or a sequence of words, in a manner identical to word-processor text selection. This requires rapid real-time segmentation of the image, using a technique we describe below. To allow the user to select larger regions, for example when copying a diagram, we have also implemented a set of rectangular region selection functions.



The two windows on the right side are under control of the agent. The top window displays search candidates, those pages which Letizia is considering to recommend to users. The bottom window displays those pages that Letizia actually decided to recommend to the user, pausing Letizia's tests for user interest.

Figure 2. The CamWorks user interface. Note the selected text in the CamWorks window, which has been copied into Microsoft Word™.

In addition to copying selected regions as images, the user can also have the selection copied as text. This is achieved by first binarising the selected region and then passing the binary image to ScanSoft's TextBridge™ OCR engine. The resultant text is then placed on the Windows® clipboard ready to be pasted into another application, for example Microsoft® Word.

CamWorks Image Processing

In this section we describe the image processing components needed to support the word-to-word selection and OCR interface.

Skew Detection

The skew of a document image is the angle of the text lines to the image raster direction. We have developed an algorithm that reliably estimates the skew angle to within 0.25 degrees on camera images. This procedure takes only a few tenths of a second[1] and so causes no significant delay for the user. Our method, is based on that described in Bloomberg

[1] Measured on a 200MHz Pentium.

[Bloomberg, Kopec and Dasari 1995].

To reduce the influence of lighting variations, we perform the skew calculation on binarised images. A window is defined, centred at the mouse click, where we expect to find a few fragments of text lines. We compute the horizontal *projection profile (i.e.* the sum the number of black pixels within each row) of this window. The profile of skewed text has a lower variance than that of non-skewed text. Thus, one way to detect the skew angle is to rotate the image through a range of angles in small increments and pick the angle that maximises the profile variance.

In our algorithm, we have found that more accurate results can be obtained by differencing adjacent rows of the profile before the variance is computed. This removes contributions to the variance due to extraneous factors such as variation in text line length. For efficiency, we approximate the rotations by vertical shears. The accuracy of this approximation limits the technique to a ?10 degree range; however, we have found that this is more than enough to account for the skew angles that arise in practice.

Text Block Segmentation

Once the skew angle has been determined, the next stage is to determine the boundaries of the column of text in which the user has clicked. Various techniques for performing page segmentation have been published in the literature, such as identification of streams of white space surrounding blocks of text [Antonocopoulos and Ritchings 1994, Pavilidis and Zhou 1992], or bottom-up grouping by repeated merging of neighbouring connected components [O'Gorman 1993]. In general, however, these techniques are too slow for an interactive system such as CamWorks, and as a result, we have developed a fast localised text block segmenter.

Our algorithm takes advantage of the fact that when the user clicks the mouse to start a selection, they normally click within the block of text to be segmented. This provides the algorithm with a starting point from which to perform the segmentation. We have used a technique based on [Antonocopoulos and Ritchings 1994, Pavilidis and Zhou 1992], which detects the streams of white space that surround the block of text in which the user has clicked. Adopting this approach allows us to perform a localised search, rather than having to make a slower global search.

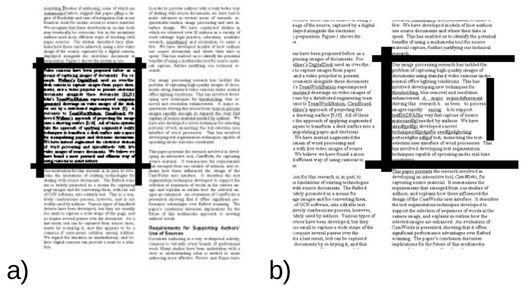


Figure 3: Text block segmentation. a) Complete segmentation, b) Incomplete segmentation

The algorithm initially performs a number of pre-processing steps on the input image, including sub-sampling, high gradient thresholding, threshold reduction and deskewing. Although the segmentation step which follows operates faster on a deskewed image, we have also developed a version of the algorithm which works with skewed images.

The segmentation step proceeds as follows: Starting from the mouse click, searches are made in the four compass directions for streams of white space. For example, moving to the right of the starting point, a search is made for a long vertical stream of white space. Similarly, moving up from the starting point, the algorithm looks for a long horizontal stream. Once four streams have been located, a check is made to ensure that the streams intersect. In the case when they do not, further searches are made and repeated checks for intersection are carried out, see Figure 3. In cases where the text extends beyond the edges of the image, the corresponding sides of the bounding box are set to the image extents.

Word and Line Segmentation

With knowledge of the text column boundaries, it is possible to identify the bounding boxes of the words and hence text lines in the column. Segmentation is performed only in the vicinity of the user's mouse pointer movements to avoid unacceptable delays. The use of local segmentation has meant that we opted for a bottom-up procedure: from connected components to characters, words and finally lines. Our method is essentially the same as the Docstrum method of [O'Gorman and Kasturi 1995], except that we using pre-calculated skew to make the Docstrum run faster. In detail, the steps of our method are as follow:

1. The image is deskewed and binarised. Connected components are detected. A graph of the adjacency relations of connected components is constructed. The nodes of this graph correspond to connected components and two nodes are joined if they are nearest neighbours in one of the four compass directions.
2. Dots of 'i's and accents are grouped with the main bodies of characters. This is achieved by merging vertically neighbouring connected components that are only separated by a small distance.
3. Characters are grouped into words. This is accomplished by modelling the distribution of inter- and intra-word character spaces. In particular, we fit a mixture of two Gaussians to a

histogram of the distances between horizontally neighbouring connected components. Components are considered to be part of the same word if they are closer than the minimum error spacing threshold for the mixture model.

4. Horizontally neighbouring words are grouped into lines. These lines are added to a data structure containing the word bounding boxes that is used to display the selection region.

Binarisation

Whenever the selected region is to be copied as text, its image must first be binarised. For accurate OCR a high-resolution binary image is needed, but video camera images characteristically suffer from low resolution, as well as lighting variations and blur. Consequently, simple thresholding algorithms cannot produce acceptable results. Many such thresholding schemes assume a two-peaked gray-level histogram, with peaks around the foreground and background gray levels. These schemes are unsatisfactory even for negligible amounts of blur because the peak corresponding to the foreground colour is invariably lost. This results from the partial voluming effects of camera pixels and the presence of lighting variations.

Instead, therefore, we have developed more sophisticated restoration and binarisation algorithms which enable us to effectively double the resolution of the imaged page. Thus we can generate full-page binary images at 120 dots per inch (dpi) from 60 dpi images captured with a typical 640x480 video conferencing camera. With half-page images we can obtain OCR error rates comparable with those from a flat-bed scanner operating at 200 dpi. Our algorithms essentially consist of two stages (see Figure 4), which we have thoroughly described and evaluated in [Taylor and Dance 1998, Taylor *et al* 1999]:

Deblurring. This stage is necessary to reduce the effects of the modulation transfer function of the camera optics and solid-state sensor. If the blur is not reduced, the edges of characters are not accurately located and defects such as the merging of thick character strokes and splitting of thin strokes severely limit OCR performance. We therefore apply a form of Tikhonov-Miller regularised deconvolution in the spatial domain for deblurring [Taylor and Dance 1998].

Binary Super-Resolution. In this phase we trade the large number of gray levels in the deblurred images for higher spatial resolution in a binary image. This is achieved by bilinear interpolation of the gray-level image followed by local adaptive thresholding. We have found that we obtain the best results using the Niblack threshold [Niblack 1986] which is taken to equal the mean gray value in the 7 x 7 square window centred at the pixel to be thresholded. The potential problem of thresholding regions of pure background or foreground colour is eliminated by placing hard upper and lower bounds on this mean gray value.

Evaluation

Throughout this research, an objective has been to provide users with more efficient ways of handling source materials. The evaluation exercise described in this section has provided us with some preliminary indications of the efficiency gains offered by CamWorks.

In the evaluation, six participants were asked to use a flatbed scanner or CamWorks to select and copy text from a paper source. The flatbed scanner was operated via a market-leading software package supporting scanning and OCR; the paper sources were scanned as black and white images at 300 dpi. All six participants were trained to use both the flatbed scanning software and CamWorks.

In both the scanner and CamWorks conditions, participants were asked to copy three selections of text, of varying lengths, from the paper sources. Participants were instructed to paste each copied selection into an electronic document and to make whatever edits were needed to ensure that just the required words had been pasted. Participants were asked not to correct any recognition errors in the pasted text.

For each of the selections copied, using both CamWorks and the scanner, the time taken to complete the task and the character recognition error rate were measured. In addition to the measures taken for using CamWorks and the flatbed scanner, we asked a seventh participant to type the three selections directly from the source.

The results of this evaluation are shown in Figure 5. They indicate that CamWorks can be used to copy

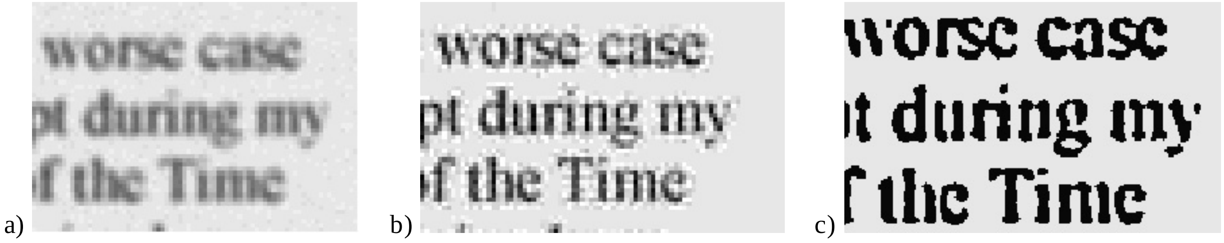


Figure 4: Camera image binarisation. a) Original degraded image. b) After deblurring c) After binary super-resolution.

50 words or less in under 30 seconds. For long sequences, however, where selection and OCR take longer, copying times using CamWorks increase to over 40 seconds. These times were considerably less than the times participants spent performing the respective copying tasks using the flatbed scanner. In each of the three conditions, it took participants over 40 seconds longer to use the flatbed scanner. Except in the 5-word condition, CamWorks was also faster than retyping.

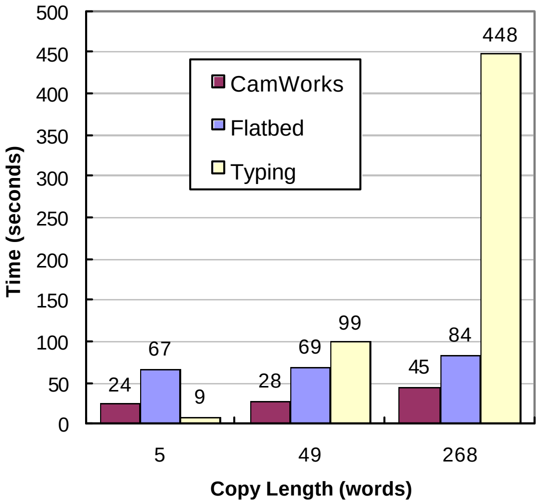


Figure 5: Average times to complete copying tasks using CamWorks, a flatbed scanner and for a touch typist.

The character recognition error rates for CamWorks, at 6.5%, were significantly higher than for the flatbed scanner (0.6%). Given that the version of CamWorks used for the experiment scanned at the equivalent of only 200 dpi, whereas the flatbed scanner operated at 300 dpi, this is not altogether surprising. Work in hand is expected to achieve error rates much closer to the flatbed rate.

Conclusions

Our studies have identified a widespread need amongst authors for rapid means of capturing source material from documents. We have developed a tool, CamWorks, to meet this need that allows rapid capture of text from paper.

The performance of CamWorks is encouraging. In our evaluation, we found it reduced the time taken to capture a short text sequence from 60 seconds or more, when using a flatbed scanner, down to 30 seconds or less. It also provided a faster alternative to re-typing source text, except in the 5-word condition. OCR error rates are higher using CamWorks, but recent research suggests that these can be brought down to near the 1 percent rate achievable with a 300 dpi flatbed scanner.

References

Antonacopoulos A. and Ritchings R.T. Flexible Page Segmentation Using the Background. In *Proceedings of the 12th International Conference on Pattern Recognition*, **2**, 1994, pp. 339-344.

Bloomberg D. S., Kopec G. E. and Dasari, L. Measuring document image skew and orientation. In *SPIE Conference on Document Recognition II,* **2422**, 1995, pp. 302-315.

Haas, C., Writing Technology - Studies on the materiality of literacy. Mahwah, New Jersey, Lawrence Erlbaum, 1996.

Ishii H. and Kobayashi M., ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact. Proceedings of CHI '92 Human Factors in Computing Systems (May 3-7, Monterey, CA) ACM/SIGCHI, N.Y., pp. 525-532, 1992.

Ishii H., Kobayashi M. and Arita K., Iterative Design of Seamless Collaboration Media. Communications of the ACM Vol. 37, pp. 83-97, 1994.

Jain, A.K. Fundamentals of Digital Image Processing. Prentice Hall International, Englewood Cliffs, 1989.

Niblack W, An Introduction to Digital Image Processing. Prentice Hall, Englewood Cliffs, N.J., 1986.

O'Gorman L., The Document Spectrum for Page Layout Analysis. in IEEE Transactions On PAMI, Vol 15, No. 11, Nov 1993.

O'Gorman L. and R. Kasturi, Document Image Analysis. IEEE Computer Society Press, Los Alamitos, 1995.

O'Hara K. and Sellen A. J., A Comparison of Reading Paper and On-Line Documents. Proceedings of CHI '97 Human Factors in Computing Systems (March 22-27, Atlanta GA) ACM/SIGCHI, N.Y., pp. 335-342, 1997.

O'Hara K., Smith F., Newman W. M. and Sellen A. J., Student Readers' Use of Library Documents: Implications for Library Technologies. Proceedings of CHI 98 Human Factors in Computing Systems (April 18-23, Los Angeles CA) ACM/SIGCHI, N.Y., pp. 233-240, 1998.

Pavlidis T. and Zhou J. Page Segmentation and Classification. *CVGIP: Graphical Models and Image Processing*, Vol. 54, 6, 1992, pp. 484-496.

Taylor M. J. and Dance C. R. Enhancement of Document Images from Cameras. In *SPIE Conference on Document Recognition V,* 3305, 1998, pp. 230-241.

Taylor M. J., Zappala A., Newman, W. M. and Dance C. R. Documents Through Cameras. To appear in *Image and Vision Computing*, 1999.

Wellner P., The DigitalDesk Calculator: Tangible Manipulation on a Desk Top Display. Proc. ACM Symposium on User Interface Software and Technology, UIST '91 (Hilton Head SC, November 11-13), 1991.

Wellner P., Interacting with Paper on the DigitalDesk. Comm. ACM Vol. 36, 7, pp. 86-96, 1993.