CamWorks: A Video-based Tool for Efficient Capture from Paper Source Documents

William Newman, Chris Dance, Alex Taylor, Stuart Taylor, 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

We describe the design and evaluation of CamWorks, a system that employs a video camera as a means of supporting capture 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.

Keywords

Advanced man-machine interfacing, image processing.

Introduction

One of the benefits offered by information technology is its capacity to convert information from one medium to another ? live video to static images, page images to coded text, text to synthesized speech, and so on. This capability, achieved with the aid of technologies such as optical character recognition (OCR), enables the user to work in the medium best suited to his or her current task. Sometimes, however, there is no 'best' medium: the user needs to work in several media concurrently. Situations of this kind create a need for multimedia systems.

A particularly common example of the need to work with multiple media arises when authors, working on-line, need to access paper documents such as books, articles and reports. This combination of referring to paper source material while preparing electronic text is an exceedingly widespread form of multimedia working. Despite efforts to develop paperless authoring environments, there is little evidence that the use of paper is diminishing. Studies of authoring, some of which are summarized below, 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. We recognise that these drawbacks in on-line tools may eventually be overcome, but in the meantime authors need more efficient ways of working with paper sources. The system described here illustrates how these can be achieved, 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.

Video cameras have been proposed before as a means of capturing images of documents. For example, Wellner's DigitalDesk used an over-the-desk camera to capture images from paper documents, and a video projector to present electronic documents alongside these documents [21,22]. Ishii's TeamWorkStation superimposed computer generated drawings on video images of the desk, for use by a distributed engineering team [8]. A successor to TeamWorkStation, ClearBoard, followed Wellner's approach of projecting the image onto a drawing surface [8,9]. All of these systems take the approach of applying augmented reality techniques to transform a desk surface into a space for manipulating paper and electronic documents. We have instead augmented the electronic domain of word processing and spreadsheets with live video images of source documents. We believe we have found a more practical and efficient way of using cameras to assist authors.

Our motivation for this research is, in part, to overcome the limitations of existing technologies for dealing with source documents. The flatbed scanner is widely promoted as a means for capturing page images and for converting them, with the aid of OCR software, into editable text. This is a relatively cumbersome process, however, and is not widely used by authors. Various types of handheld devices 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. As a last resort, text can be captured from source documents by re-keying it, and this appears to be a common if error-prone solution among authors. We regard the situation as unsatisfactory, and believe digital cameras can provide a route to a solution.

This paper presents the research involved in developing an interactive tool, *CamWorks*, for capturing source material. It summarises the requirements that emerged from our studies of authors, and explains how these influenced the design of the CamWorks user interface. It describes the text segmentation techniques developed to support the selection of sequences of words in the camera image, and explains in outline how the selected images are enhanced. An evaluation of CamWorks is presented, showing that it offers significant performance advantages over flatbed scanning. The paper's conclusion discusses implications for the future of this multimedia approach to meeting authors' needs.

Requirements for Supporting Authors' Use of Sources

Document authoring is a very widespread activity, common to virtually every branch of professional work. Many studies have been undertaken with a view to understanding what is needed to make authoring more effective. Flower and Hayes have reported on numerous studies in which they considered both the cognitive elements of authoring and the writing process more generally [6]. In a longitudinal study of writers of technical documents, Severinson and Sjöholm found that when their study's participants composed documents on-line, rather than using pen-and-paper, there was a noticeable difference in how they planned, revised and amended what they wrote [18].

Perhaps the most important requirement to emerge from these studies is that authoring tools must continue to support the use of paper-based source materials. In a world that is attempting to eradicate paper, this might seem significant only in the short term. However, studies by Haas and by O'Hara and Sellen have shown the clear benefits to authors of working with paper source documents [7,15]. In the experiment by O'Hara and Sellen, for example, one group of participants performed a writing task using only paper (sources and target), while another group used only on-line documents. Those using paper were at a considerable advantage when navigating around their source documents. At the same time, there were obvious benefits to constructing the document on-line, using a word processor. Overall, these studies suggest that the preferred combination for most authors, for some time to come, is likely to be on-line creation using paper sources.

While these studies are informative about the factors that affect authoring, they leave some important questions unanswered. How do authors select material from source documents for use in the documents they write? What kinds of transformations do they apply to the material? We have undertaken studies of our own in order to answer questions like these and thus gain a better understanding of authors' needs for document capture tools.

Our first study involved 25 postgraduate humanities students, each of whom kept a diary of a day spent in a local academic library [16]. It highlighted the need for efficient ways of capturing text while reading. A follow-on study has focused on use of sources during writing - sources that include notes taken while reading. Participants in this study included some of the 25 graduate students, together with people in various professional jobs: lawyers, consultants, educationalists and journalists. We videotaped the participants at work, and made detailed analyses of how they interleaved access to sources with writing, and how the source text contributed to the target text.

A preliminary analysis of this latest study identifies several 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.
- *Authors need efficient ways of capturing longer text segments*. We noticed, for example, that some participants would quite commonly spend two minutes or more typing a paragraph verbatim.
- *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. At the final stages, however, authors usually take care to ensure the accuracy of verbatim quotes.

Some of these requirements have emerged relatively recently. For the most part, they confirm the validity of the multimedia approach we have adopted. Our most recent study is helping us understand how to evaluate our prototype system, and is suggesting some further research directions. We return to these two topics at the end of the paper.

Designing the CamWorks User Interface

A video camera mounted over the author's desk has the potential to meet our primary requirements for the capture of text segments, both small and large, during reading and writing, without interrupting the flow of work. It can deliver an image of a paper document to the user, who can then quickly select the material to be copied. There are, however, a number of further aspects to authors' requirements. These needed to be addressed in the design of our CamWorks prototype.

We needed to decide whether to provide a static or a "live" camera image of the document. We experimented with both, 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 of the flow of reading or writing. Our basic design strategy for CamWorks has therefore been to provide a continuously updated video image of the document under the camera.

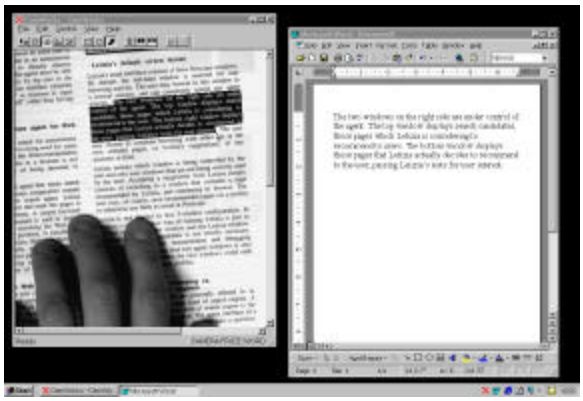

Figure 2. The CamWorks user interface. Note the selected text in the CamWorks window, which has been copied into Microsoft Word™.

The need to support the capture of small text segments has led us to take a novel approach to selection of regions of the video image. 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, and can copy the text into the Windows™ clipboard. This requires rapid real-time segmentation of the image, using a technique we describe below. The result is that the user can employ very similar methods of text selection within the source and target documents.

An implicit requirement, emerging from our studies, is the need to support the capture of editable text. Images of text segments therefore need to be converted to sequences of characters using OCR software, and this conversion must not be so slow as to add significantly to the total capture time. Accuracy is also important, and we have set a target of keeping recognition errors below 1 percent. Our current solution strategy is to build in a commercial OCR package (ScanSoft's TextBridge™) so that conversion is automatically applied to selected text.

Authors occasionally need to select larger regions of the page, for example when copying a diagram. For these operations, selection of a rectangular region is the most effective method. We have therefore supplemented CamWorks' text-selection methods with a set of region selection functions. Again, these are invoked in a very similar way to the selection of a region in a graphics editor.

Supporting the user interface

In this section we describe the image processing components needed to support the word-to-word selection interface. These components implement skew detection, column segmentation, word segmentation and binarisation (image conversion from colour to black and white). The overall process followed when the user begins selecting a region of text is shown in Figure 3.

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.

To reduce the influence of lighting variations, we perform the skew calculation on binarised images. We have achieved satisfactory results by using a particularly efficient *high-gradient* binarisation. The method involves setting all pixels whose gray-level gradient is greater than some threshold to black and all others to white. Gradients are estimated using the Sobel filter [10].

Most previous techniques for skew detection require extraction of the bounding boxes of connected components [2,11]. Our skew estimation method, which is

[1] Measured on a 200MHz Pentium.

based on that described in Bloomberg [3], only requires the counting of black pixels and hence is very efficient. 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* of this window, that is, we sum the number of black pixels within each row. As shown in Figure 4, the projection 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 (e.g. 0.25 degree) and pick the angle that maximises the profile variance.

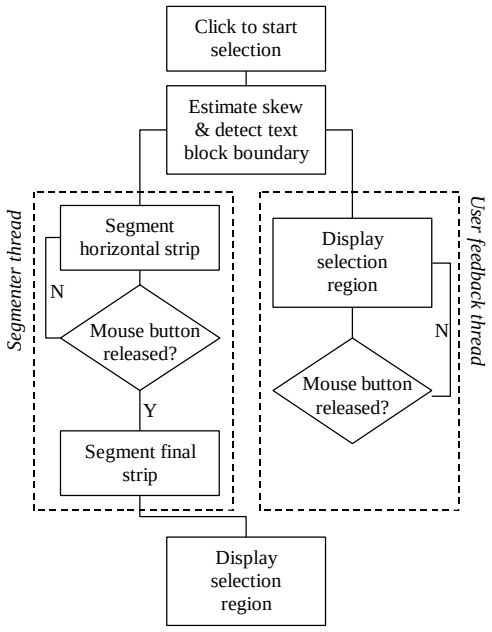

Figure 3. Segmentation and feedback in CamWorks during text selection.

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.

We start the skew detection process using a 100 x 100 pixel window, which generally suffices for 10-12 point fonts. However, if this window contains very little text, or if the text is of a relatively large point size, skew detection will be inaccurate. The skew estimate is deemed inaccurate when the ratio of the peak profile variance to the average profile variance is less than a threshold. When this test fails, skew estimation is repeated with a window with doubled linear dimensions.

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. We have been able to accomplish this by using the page segmentation routines of our OCR package, using a reduced resolution image of the whole page to gain speed. Other approaches that we might have used had this solution been unavailable to us include top-down grouping of text using multi-scale wavelet based texture analysis [5], identification of streams of white space surrounding a block of text [1,17], or bottom-up grouping by repeated merging of neighbouring connected components [13].

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. We have found that word segmentation of the whole column prior to updating the display of the selected region can lead to an unacceptable delay for the user. Instead we have developed a novel approach using a *dynamically expanding* segmentation region.

This approach proceeds by segmenting one horizontal strip of the column at a time. The strips are chosen to be about four text lines high. At each stage, a decision about which strip to segment next is made on the basis of the user's cursor movements. If the user has finished selecting, any remaining strips between the initial and final pointer position are segmented. Otherwise, the next strip is chosen to extend the segmented region in the direction of the current pointer position relative to the initial click position. As shown in Figure 3, this process operates in a separate thread of program execution. Thus user feedback about the selected region is available as soon as a few lines of text have been segmented.

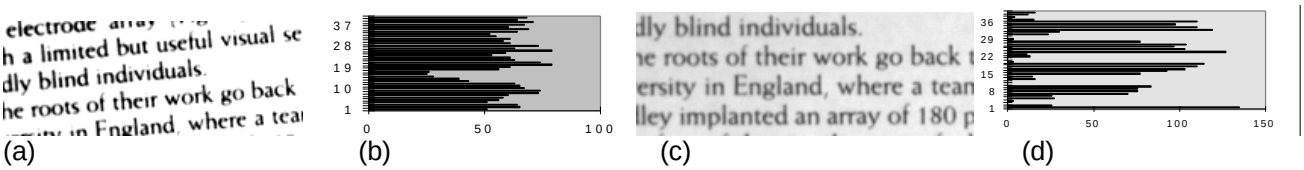

Figure 4. Projection profiles: (a) skewed text; (b) horizontal projection profile of (a); (c) non-skewed text; (d) horizontal projection profile of (c).

The segmentation of an individual horizontal strip proceeds in a bottom-up fashion: from connected components to characters, words and finally lines. This inner segmentation operation is essentially the Docstrum method of O'Gorman and Kasturi [14], except that we are using pre-calculated skew to make the Docstrum run faster. In detail, the steps of our method are as follow:

1. The strip is deskewed and binarised. Connected components are detected. A single horizontal strip may cut a text line, thus those connected components that lie within one text line height of the top or bottom of the strip may be only fragments of letters. Such components are excluded from subsequent calculations on segmentation of the first strip and are merged with connected components from the top and bottom of subsequent strips on future segmentations.
2. 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.
3. 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.
4. 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.
5. 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 5), which we have thoroughly described and evaluated in [19,20]:

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 [19].

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 [12] 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.

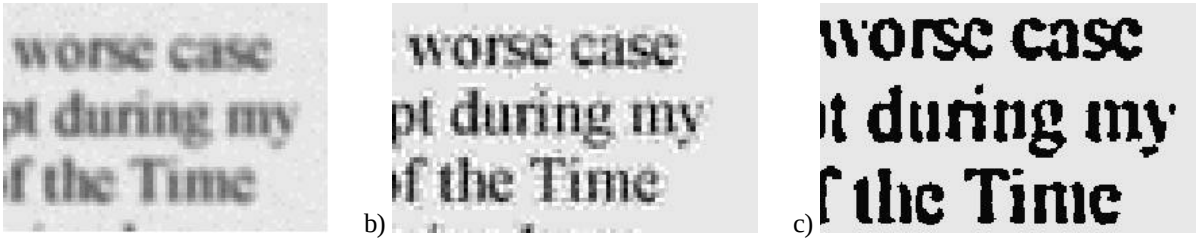

Figure 5: Camera image binarisation. a) Original degraded image. b) After deblurring c) After binary super-resolution.

Evaluation

Throughout this research, an objective has been to provide authors with more efficient ways of handling source materials than are currently available. 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 set tasks similar to the text-capture tasks we had observed during our studies. In separate conditions, each participant was 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. The participants were trained to use the software and were given some time to familiarise themselves with the scanning process prior to any measures being taken. All six participants were already familiar with CamWorks and did not require any training to use it. They were asked, however, to use only the word-to-word selection technique when copying the text in the CamWorks conditions.

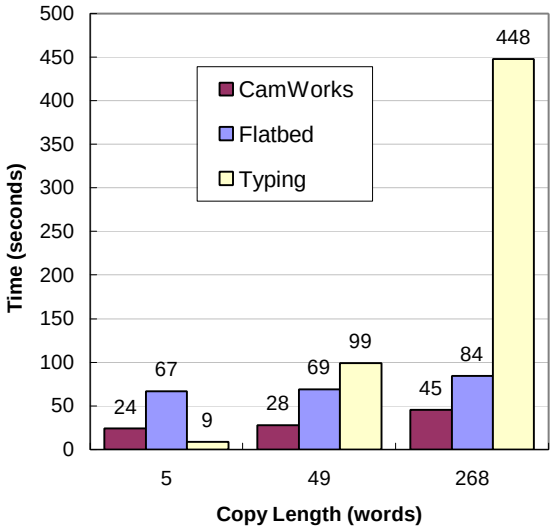

Figure 6. Average times to complete copying tasks using CamWorks, a flatbed scanner and for a touch typist.

In both the scanner and CamWorks conditions, participants were asked to copy three selections of text, of varying lengths, from the paper sources. The selections were chosen to be roughly 5, 50 and 250 words long. Source documents were printed in a 13-point font, and the selections to be copied were marked out in red. Participants were instructed to paste each copied selection into an electronic document, and then to make whatever edits were needed to ensure that just the marked-out words had been selected and pasted. In the flatbed condition, this usually involved deleting some text before and after the required selection. 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. Time was measured from when the document was picked up from a face-up reading position beside the computer's keyboard, to when the participant had copied the appropriate text and returned the source to beside the keyboard. 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 6. They indicate that CamWorks can be used to copy 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.

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.

In summary, this evaluation confirms that CamWorks can offer substantial benefits to the author as a tool for capturing source material. It has the potential to reduce capture times to a point where there is little disruption to the writing process, something that flatbed and other scanning technologies seem unlikely to achieve. It can provide a faster alternative to retyping long passages of source text. OCR error rates are high, however, and would probably deter authors from using the current CamWorks technology as a means of capturing from small-print sources.

Conclusions

Our studies have identified a widespread need amongst authors for rapid means of capturing source material from documents, and we have tried to meet this need. We have observed authors' well-justified preference for using sources in paper form, and have therefore developed a tool, CamWorks, that allows rapid capture of verbatim material from paper. We have adopted a multimedia approach in which we provide a live video view

of the source document alongside the electronic document in preparation.

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 high 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.

Improvement of OCR error rates is just one area where further research is needed. CamWorks needs to be evaluated against other devices, such as hand-held and sheet-feed scanners. Some of these appear capable of faster capture than flatbed scanners, but they all involve the user in extra steps that are likely to drive up capture times. Other applications of camera-based scanning also deserve investigation, e.g., scanning solid objects or fragile documents, or the support of collaboration as already suggested by Ishii [9].

Beyond these immediate areas of further research lie questions about the future of authoring work, which is already showing the influence of increased access to on-line collections of source documents. On-line sources permit the inclusion of media such as animations, live video and audio that cannot be reproduced on paper. Today's standard practice of writing on-line from paper sources, which has motivated the development of CamWorks, is likely gradually to give way to new practices which will in turn demand efficient multimedia support tools.

Acknowledgements

We are grateful to other members of Xerox Research Centre Europe who have helped in this work, especially Fiona Smith, Kenton O'Hara, Mauritius Seeger and Bob Anderson. Several members of the Centre volunteered as participants in the evaluation, including Ercan Kuruoglu, Michelle Smith and Lotte Dean. Others in Xerox have offered useful advice, including Keith Emanuel and Denise McLaughlin.

References

[1] 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.

[2] Baird H., The skew angle of printed documents. Proc. SPIE Symp. Hybrid Imaging Systems, Rochester, NY, pp. 21-24, 1987.

[3] 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.

[4] Chen S., and Haralick R. M., An automatic algorithm for text skew estimation in document images using recursive morphological transforms. ICIP-94, Austin, TX, pp. 139-143, Nov. 1994.

[5] Doermann D., Page Decomposition and Related Research. Proc. Symp. Document Image Understanding Technology, pp. 39-55, Bowie, Md., 1995.

[6] Flower L. and Hayes J. R., Plans That Guide the Composing Process. In Frederiksen C. H. and Dominic J. F., eds., Writing: The Nature, Development and Teaching of Written Communication Vol. 2. Hillsdale, New Jersey, Lawrence Erlbaum, 1981.

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

[8] 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.

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

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

[11] Jain K. and Yu B., Document Representation and its Application to Page Decomposition. IEEE Trans. PAMI, Vol. 30, No. 3, pp. 294-308, 1998.

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

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

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

[15] 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.

[16] 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.

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

[18] Severinson E. K. and Sjöholm C., Writing with a computer: a longitudinal study of writers of technical documents. International Journal of Man-Machine Studies, Vol 35, 5, pp. 723-749, 1991.

[19] 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.

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

[21] 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.

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