Oblivious Document Capture and Real-Time Retrieval

Christoph H. Lampert[†], Tim Braun[*], Adrian Ulges[*], Daniel Keysers[†], Thomas M. Breuel[†]

[†]German Research Center for Artificial Intelligence (DFKI), 67608 Kaiserslautern, Germany
[*]University of Kaiserslautern, 67663 Kaiserslautern, Germany
{chl, braun, ulges, keysers, tmb}@iupr.net

Abstract

Ever since text processors became popular, users have dreamt of handling documents printed on paper as comfortably as electronic ones, with full text search typically appearing very close to the top of the wish list.

This paper presents the design of a prototype system that takes a step into this direction. The user's desktop is continuously monitored and of each detected document a high resolution snapshot is taken using a digital camera. The resulting image is processed using specially designed dewarping and OCR algorithms, making a digital and fully searchable version of the document available to the user in real-time. These steps are performed without any user interaction. This enables the system to run as a background task without disturbing the user in her work, while at the same time offering electronic access to all paper documents that have been present on the desktop during the uptime of the system.

1 Introduction

For capturing images of documents, digital cameras have many advantages over other commonly used devices like flatbed scanners. Most prominent among these advantages are their small physical dimensions and their ability to work in real-time and without physical contact at a distance. For many applications, these advantages outweigh possible disadvantages like lower image resolution and perspective distortions in the images. Another major point in favor of the use of digital cameras for document capture is their wide availability, which still growing, for example with the use of cameras built into mobile phones. Thus, if today there are hardly any products that make use of cameras for document image capture, this is largely not due to limitations of the hardware, but to missing software support.

Our aim with this paper is to bring camera-based document capture closer towards an actually useful product, by presenting a prototype system that relies on what we consider the biggest advantage of cameras for document capture: the ability to work completely without physical user interaction, and therefore without interrupting the user's everyday work-flow.

Most users will agree that it would often be very useful to have all their documents available in electronic form, if only to allow fast distribution by email or for full text search. However, it is unrealistic to hope that paper-based documents will disappear in the near or middle future, since paper offers too many advantages, e.g. excellent readability, low cost, and a basis for handwritten annotations or authentification marks like signatures and stamps.

Our setup therefore does not try to replace printed documents, but to enrich them by creating an additional digital version of each printed document that a user is studying during the day, that is, of each document that appears on his desktop. By saving, processing, and indexing all documents obtained that way, the user is provided with a digital archive of all the documents that ever crossed his desk.

To achieve this ambitious goal we let a digital camera constantly monitor the user's desktop in low resolution and trigger a high resolution capturing process whenever a document is detected. The resulting image is automatically processed, freed of typical distortions, and its textual content is extracted and stored along with the image itself. All this happens in a background process on a PC and no user interaction is required such that the user is never disturbed. If, however, the user wants to access such a document in its digital form at some time, he can do so by using a simple GUI for full text and meta search.

Even though there is a vast amount of literature on camera-based document capture and desktop surveillance, we are not aware of previous systems that aim in the same direction as we propose. However, we would like to mention the CamWorks system [10] that also targets real-time digital access to printed source, but concentrates on cut-and-paste operations using a low resolution video camera. Typical described uses for systems observing user desktops

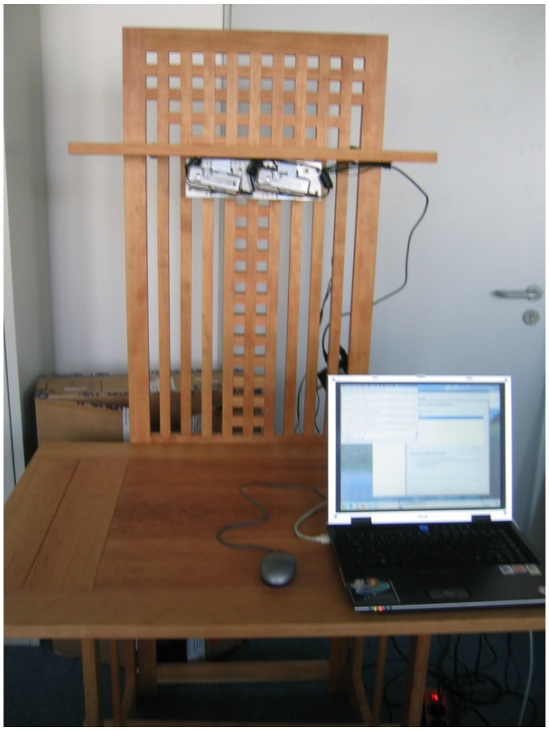


Figure 1. The complete setup: a wooden desk with two cameras attached and a laptop PC running the software components.

are not targeted at the automatic capture of document images, but at extracting user behavior from objects and gestures (e.g. [8]) or at creating 'active areas' on a desktop, e.g. providing specialized functionality like a desktop calculator [17].

2 Hardware

To build a prototype system that mainly demonstrates the functionality of the software, we have abstained from using any special hardware devices, like infrared light or mechanical sensors. Instead, we rely on standard components only. These are two *Canon Powershot S50* digital cameras with a resolution of 5 megapixels each and a laptop PC. The cameras are mounted to a base plane that is attached to the rear panel of a wooden working desk, and a data connection from the cameras to the laptop is established using standard USB cables. Figures 1 and 2 show this setup. All further functionality is provided by software components as described in the following section. Note that our choice of two cameras is motivated only by one of the dewarping algorithms relying on stereo vision (see Section 3.3). A setup with only one camera or with more than two cameras would be just as feasible.

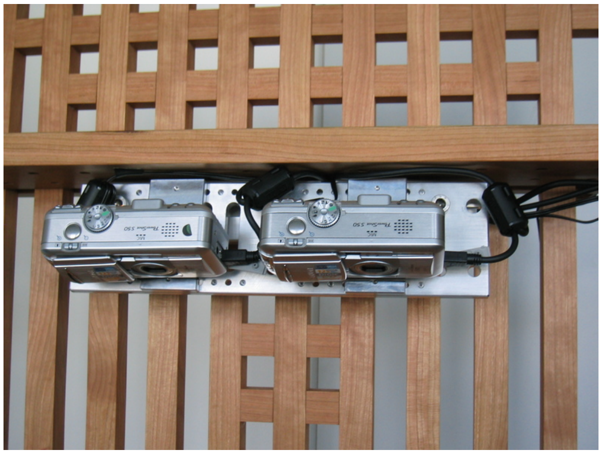


Figure 2. The camera section of the setup in more detail.

3 Software

Our software is designed as a set of independent modules. In the following sections, we will give overviews of each of the different software components.

3.1 Surveillance

Each digital camera can continuously send up to 25 low resolution viewfinder images per second via the USB interface to the PC. For our setup we analyze five input images per second to determine whether a new document has been placed within the field of view.

In the following, the employed document detection method will be introduced. A more detailed description of the approach can be found in [3].

The document detection method consists of two stages. The first stage extracts relevant features from the input image, while the second stage validates these features to discriminate between documents and other visible objects. Because nearly all printed documents are roughly rectangular, we use this a-priori information by using the outline of the object as the main feature to detect printed documents. Therefore, the extraction stage determines the contour outlines of all objects that enter the view volume of the camera. The subsequent validation stage then checks whether the outlines are approximately rectangular and rejects all non-matching contours. Valid outlines that are detected in a number of subsequent frames are accepted as a document; the corresponding image position is then passed to the document capture module which will be described in Section 3.2. In the following, the two stages of the document detection method are presented in more detail.

Feature Extraction. The feature extraction stage is subdivided into four image processing steps. The first step compensates the radial distortion present in the input images, which would otherwise deform the outlines of all visible objects.

Secondly, static background visible in the input image is removed using a standard ‘image difference’ background subtraction method, where a previously recorded background model without visible documents is subtracted from the current input image. The background model is recorded once during the setup of the system and then continuously updated to account for changing illumination. The absolute difference image between frame and background is thresholded to produce a binary image with highlighted foreground regions. To achieve some invariance to changes in illumination, the thresholding operation is performed only in the chrominance channels of the YUV color space.

The third processing step takes the generated binary image, performs morphological smoothing and then extracts the contours of all foreground regions using the hierarchical boundary detection algorithm presented by Suzuki and Abe [13]. With this algorithm, the outer boundary of every foreground object is found, while inner contours are ignored.

To decrease the runtime of the following validation step, each extracted contour is simplified in the fourth and final feature extraction step. This simplification is done with the Douglas-Peucker algorithm [7], which approximates contours by polygons of an adjustable number of edges.

Feature Validation. The simplified contours of foreground objects are examined to locate rectangular documents. To accomplish this in a noise-robust way, the feature validation stage first fits lines to each detected contour. Then, the algorithm checks whether the detected lines are pairwise orthogonal and satisfy several additional constraints. Since the line fitting effectively smoothes the object contours and tolerates small undetected or occluded parts, this process is able to robustly detect rectangular shapes even when the extracted outlines are degraded.

Lines are fit to each contour using the robust branch-and-bound algorithm RAST [4]. RAST can be employed to detect prominent linear structures in a similar fashion as the well known Hough transform. However, RAST does not suffer from quantization effects, is computationally more efficient, and can explicitly use short line segments as primitives instead of single pixels. This makes RAST especially suited for the presented system, because the contour extraction/simplification stage outputs short line segments, i.e. the elements of the polygon approximation to the contour. Since there are much fewer line segments than single points, the RAST algorithm gains an additional speedup in comparison to the Hough transform. This is of special importance in the presented system, because the document detection must run in real-time.

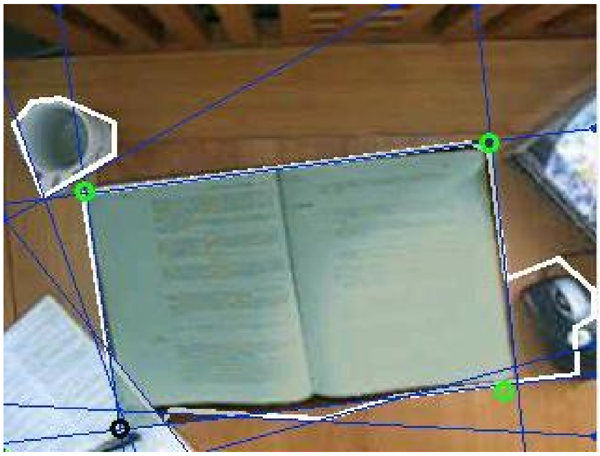


Figure 3. Internal debug image during the validation phase with highlighted contour lines. Edges and corners of a roughly rectangular document object are marked.

Before the fitted lines can be checked for orthogonal intersections, the perspective distortion caused by the particular placement of the cameras must be removed. To do so, a corrective projection is computed in an extrinsic calibration stage during system setup, in which an A4-sized sheet of paper is placed on the desktop and then used as a calibration object.

After applying the corrective projection to the fitted lines, the angles between them are identical to those between the corresponding contour elements in the real world. Thus, parallel lines correspond to parallel contour edges of an object in the camera field of view. To finally detect rectangular objects, parallel lines are collected into groups, and candidates for rectangular objects are generated from these groups by checking whether two groups are orthogonal to each other within a tolerance threshold. If this structural validation is true for any two groups, the largest rectangle that can be formed using lines contained in the two groups is determined and its four corner positions are calculated.

A rectangular object is accepted as a document only if the four corresponding corners are detected across several camera frames. We use a clustering of detected corner positions over time. The threshold for acceptance of a region as a document is then based on the number of corner positions in each cluster.

Whenever a document has been successfully detected using the described process, the actual capture process is initiated. Figure 3 shows an internal debug image of the system in which detected lines and corner points of a document are shown.

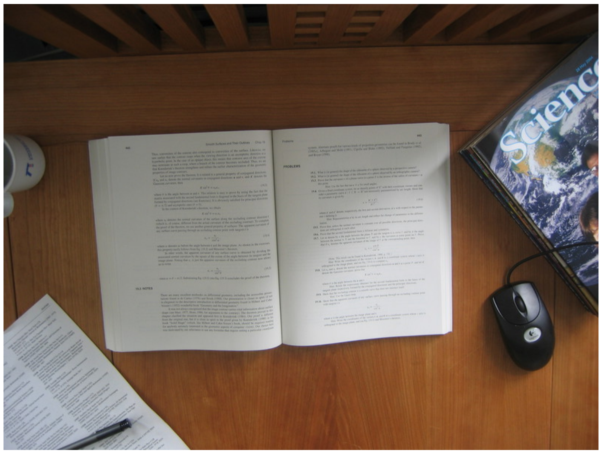


Figure 4. Captured high resolution image of the desktop, including a document.

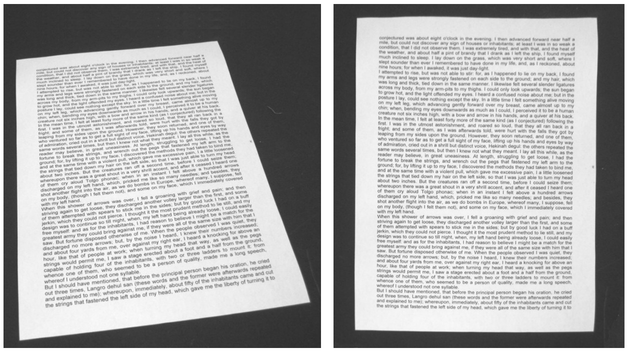
3.2 Capture

Each camera can capture a full resolution image (2614×1958 pixels) of the desktop, see Figure 4. The capture process itself is controlled by the free software library *libptp*[1], that allows to remotely control digital cameras compatible with the *Picture Transfer Protocol (PTP)* [1] via the USB interface. Because the surface of a desktop is rather large compared to the resolution of the digital camera, the resulting images have a resolution of only approximately 100 dpi for the document, which is close to the lower bound of what is useful for document capture. For a production system, more or higher resolution cameras should be considered.

3.3 Content Extraction

Preprocessing. The captured image shows the whole desktop and is first cropped to only contain the detected document itself. The coordinates of the corners of the document that were obtained during the surveillance step are reused for this. Cropping reduces the image size, usually by a factor of 4 to 8, thus allowing faster processing.

When capturing images with a camera, a perspective distortion occurs, which causes more distant parts of the image to appear smaller. The most severe effect of this distortion is that parallel lines in the real world do not appear parallel anymore in the image. However, since we have calibrated our cameras in advance it is easy to correct for this perspective distortion. At the same time we rotate the rectangular image object into an upright position, see Figure 5.



conjectured was about eight o'clock in the evening. I then advanced forward near half a mile, but could not discover any sign of houses or inhabitants; at least I was in so weak a condition, that I did not observe them. I was extremely tired, and with that, and the heat of the weather, and about half a pint of brandy that I drank as I left the ship, I found myself much inclined to sleep. I lay down on the grass, which was very short and soft, where I slept sounder than ever I remembered to have done in my life, and, as I reckoned, about nine hours; for when I awaked, it was just day-light.

I attempted to rise, but was not able to stir: for, as I happened to lie on my back, I found my arms and legs were strongly fastened on each side to the ground; and my hair, which was long and thick, tied down in the same manner. I likewise felt several slender ligatures across my body, from my arm-pits to my thighs. I could only look upwards; the sun began to grow hot, and the light offended my eyes. I heard a confused noise about me; but in the posture I lay, could see nothing except the sky. In a little time I felt something alive moving on my left leg, which advancing gently forward over my breast, came almost up to my chin; when, bending my eyes downwards as much as I could, I perceived it to be a human creature not six inches high, with a bow and arrow in his hands, and a quiver at his back. In the mean time, I felt at least forty more of the same kind (as I conjectured) following the first. I was in the utmost astonishment, and roared so loud, that they all ran back in a fright; and some of them, as I was afterwards told, were hurt with the falls they got by leaping from my sides upon the ground. However, they soon returned, and one of them, who ventured so far as to get a full sight of my face, lifting up his hands and eyes by way of admiration, cried out in a shrill but distinct voice, Hekinah degul: the others repeated the same words several times, but then I knew not what they meant. I lay all this while, as the reader may believe, in great uneasiness. At length, struggling to get loose, I had the fortune to break the strings, and wrench out the pegs that fastened my left arm to the ground; for, by lifting it up to my face, I discovered the methods they had taken to bind me, and at the same time with a violent pull, which gave me excessive pain, I a little loosened the strings that tied down my hair on the left side, so that I was just able to turn my head about two inches. But the creatures ran off a second time, before I could seize them; whereupon there was a great shout in a very shrill accent, and after it ceased I heard one of them cry aloud Tolgo phonac; when in an instant I felt above a hundred arrows discharged on my left hand, which, pricked me like so many needles; and besides, they shot another flight into the air, as we do bombs in Europe, whereof many, I suppose, fell on my body, (though I felt them not), and some on my face, which I immediately covered with my left hand.

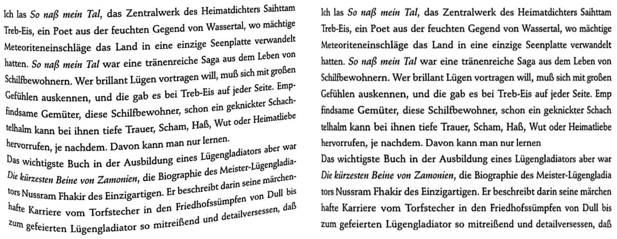
When this shower of arrows was over, I fell a groaning with grief and pain; and then striving again to get loose, they discharged another volley larger than the first, and some of them attempted with spears to stick me in the sides; but by good luck I had on a buff jerkin, which they could not pierce. I thought it the most prudent method to lie still, and my design was to continue so till night, when, my left hand being already loose, I could easily free myself: and as for the inhabitants, I had reason to believe I might be a match for the greatest army they could bring against me, if they were all of the same size with him that I saw. But fortune disposed otherwise of me. When the people observed I was quiet, they discharged no more arrows; but, by the noise I heard, I knew their numbers increased; and about four yards from me, over against my right ear, I heard a knocking for above an hour, like that of people at work; when turning my head that way, as well as the pegs strings would permit me, I saw a stage erected about a foot and a half from the ground, capable of holding four of the inhabitants, with two or three ladders to mount it: from whence one of them, who seemed to be a person of quality, made me a long speech, whereof I understood not one syllable.

But I should have mentioned, that before the principal person began his oration, he cried out three times, Langro dehul san (these words and the former were afterwards repeated and explained to me); whereupon, immediately, about fifty of the inhabitants came and cut the strings that fastened the left side of my head, which gave me the liberty of turning it to

(a) The captured image of a single sheet of paper. Before...

(b) ...and after perspective correction.

Figure 5. Removal of perspective distortion for a planar document.



Ich las *So naß mein Tal*, das Zentralwerk des Heimatdichters Saihttam Treb-Eis, ein Poet aus der feuchten Gegend von Wassertal, wo mächtige Meteoriteneinschläge das Land in eine einzige Seenplatte verwandelt hatten. *So naß mein Tal* war eine tränenreiche Saga aus dem Leben von Schilfbewohnern. Wer brillant Lügen vortragen will, muß sich mit großen Gefühlen auskennen, und die gab es bei Treb-Eis auf jeder Seite. Empfindsame Gemüter, diese Schilfbewohner, schon ein geknickter Schachtelhalm kann bei ihnen tiefe Trauer, Scham, Haß, Wut oder Heimatliebe hervorrufen, je nachdem. Davon kann man nur lernen.

Das wichtigste Buch in der Ausbildung eines Lügengladiators aber war *Die kürzesten Beine von Zamonien*, die Biographie des Meister-Lügengladiators Nussram Fhakir des Einzigartigen. Er beschreibt darin seine märchenhafte Karriere vom Torfstecher in den Friedhofssümpfen von Dull bis zum gefeierten Lügengladiator so mitreißend und detailversessen, daß

(a) A captured image of a curled book surface. Before...

(b) ...and after the dewarping process.

Figure 6. Removal of geometric distortion using the straightening of text lines.

OCR. In the next step, we want to extract the textual information from the document by performing OCR. This is not a trivial task in this setting, because—in contrast to images obtained from flatbed scanners—document images captured by cameras often contain distortions even after perspective correction. The main reason for these remaining distortions is that the documents themselves are not necessarily planar but the pages may contain an inherent curl. This causes e.g. text lines on a book page not to be straight in the captured image, see Figure 6(a). Several methods for the removal of such effects have been proposed, and we will discuss some of them in more detail in the following section. However, to our knowledge none of them is capable of working in real-time, which would be necessary in our setting, because the user might want to access a document immediately after it has been captured.

[1] http://sourceforge.net/projects/libptp

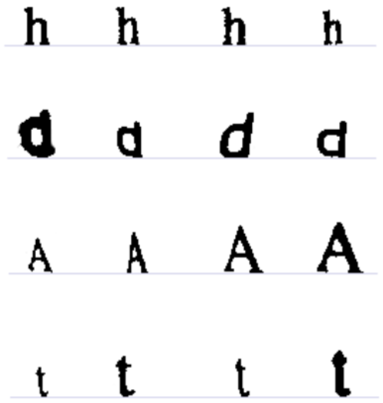


Figure 7. Typical distortions of characters as caused by nonlinear warping of the document image and noise.

We therefore have chosen the following new approach: we use a fast OCR step that is specially designed to work well for text showing a certain amount of geometric distortion. Typically distorted characters are shown in Figure 7. Single characters are not only affected by the nonlinear warping of the document image, but the binarized characters are also affected by low resolution, varying illumination, and noise.

To perform OCR robust to distortion, we train an artificial neural network with images of letters showing the same geometric distortion that occurs when taking pictures of non-planar pages. To reliably produce a large amount of training images they were created artificially using the FreeType typesetting library[2]. Noise and jitter were imitated using Baird's defect model [2]. In addition, the letters were scaled to several widths to simulate the horizontal and vertical shortening occurring when the page surface is not observed straight from above. Similar approaches are used for the recognition of handwritten digits, where the variability of the data is of a different type and for example simulated using smoothed random two-dimensional distortions [12].

Apart from the neural network, also the other steps of the OCR system are designed to work with non planar documents. For example, the tracking of text lines can only be done locally linear, since those are not straight along the whole image anymore. The RAST line finding algorithm [4] was adapted for these special requirements. Also, even on the scale of grouping letters into words, the curl of the base line can be significant, causing e.g. projection profiles or smearing approaches to fail. Instead, a robust clustering approach was used here. A detailed description of the OCR algorithm is given in [14].

Image Flattening. Ultimately, the goal is of course to obtain document images without any distortion by flattening the image. This makes the image more pleasing to the human eye, see Figure 6(b), and also much easier to process with an OCR system.

Note that the flattening step is optional in our approach and the real-time system is based on directly processing the captured and preprocessed image without flattening. However, image flattening can be used to improve the results as a background task, while already preliminary OCR results are available for real-time search.

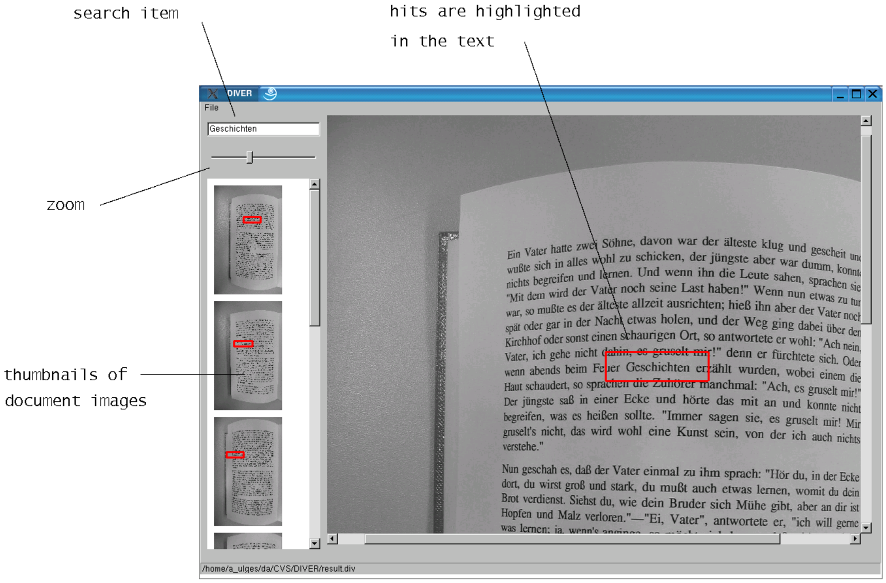
Many approaches have been proposed for image flattening, often estimating the 3D shape of the document surface using additional hardware like structured light sources [5], laser scanners [11] or a second camera [19]. In the latter case, printed text and graphics are exploited as texture for a stereo algorithm from which the 3D shape is reconstructed. Other approaches work without an explicit page model, but require additional assumptions with respect to the page content, like the existence of parallel horizontal text lines [6, 9, 18].

In our system, we have chosen to integrate two such methods as optional modules. The *straightlines* algorithm from [16] has the advantage that it essentially can reuse much of the information that has been obtained during an initial OCR step. Local estimates of the text line slope and the base line distance are used to determine the tilt angle of the surface and its distance to the camera on a mesh of points on the book surface. From this information, a scale factor for each letter is derived and the letters are warped one-by-one onto a rectangular grid of output image cells. The method works well if the documents contains regular text lines with fixed line spacing, see Figure 6(b). However, in its current state the algorithm is limited to single column documents containing only text.

The second method is computationally more expensive, but has the advantage of being able to process pages of arbitrary content, including e.g. handwritten text or illustrations. It generates a 3D page model using two camera images and stereo vision. The resulting 2D surface in 3D space is then flattened by inscribing a regular mesh of equilateral triangles which is conformally mapped onto a planar mesh and textured with patches from the camera images. For details on the stereo dewarping algorithm, see [15].

Since both methods have their own advantages, it is useful to let both try to dewarp the captured image. However, this needs large computing resources, and so we start them as background threads, working at a low CPU priority. That way, only when the system would otherwise be idle, the de-

[2] http://www.freetype.org


search item
hits are highlighted in the text
zoom
thumbnails of document images
DIVER
File

Geschichten

Ein Vater hatte zwei Söhne, davon war der älteste klug und gescheit und
wußte sich in alles wohl zu schicken, der jüngste aber war dumm, konnte
nichts begreifen und lernen. Und wenn ihn die Leute sahen, sprachen sie:
"Mit dem wird der Vater noch seine Last haben!" Wenn nun etwas zu tun
war, so mußte es der älteste allzeit ausrichten; hieß ihn aber der Vater noch
spät oder gar in der Nacht etwas holen, und der Weg ging dabei über den
Kirchhof oder sonst einen schaurigen Ort, so antwortete er wohl: "Ach nein,
Vater, ich gehe nicht dahin, es gruselt mir!" denn er fürchtete sich. Oder
wenn abends beim Feuer Geschichten erzählt wurden, wobei einem die
Haut schaudert, so sprachen die Zuhörer manchmal: "Ach, es gruselt mir!"
Der jüngste saß in einer Ecke und hörte das mit an und konnte nicht
begreifen, was es heißen sollte. "Immer sagen sie, es gruselt mir! Mir
gruselt's nicht, das wird wohl eine Kunst sein, von der ich auch nichts
verstehe."

Nun geschah es, daß der Vater einmal zu ihm sprach: "Hör du, in der Ecke
dort, du wirst groß und stark, du mußt auch etwas lernen, womit du dein
Brot verdienst. Siehst du, wie dein Bruder sich Mühe gibt, aber an dir ist
Hopfen und Malz verloren."—"Ei, Vater", antwortete er, "ich will gerne
was lernen; ja, wenn's anginge,

/home/a_ulges/da/CVS/DIVER/result.div

Figure 8. A screen-shot of the simple search user interface. Search strings can be entered, and matches in the document images are highlighted using colored boxes.

warping routines are called, and the system does not lose its ability to capture images and answer queries in real-time.

3.4 Archiving

Directly after the image capture and the initial OCR process, a digital image and the extracted text of the document are available. The text and the image are saved together along with a time stamp to the hard disk. To keep the prototype simple, the data is simply stored as individual files instead of database entries. For the text a simple ASCII format containing the actual words and their position within the document image is used. The images themselves are stored in JPEG, PNG, or DjVU format. Afterwards, the system is ready to capture the next document.

Whenever a background dewarping process for an image is finished, the flattened document image is saved to the same folder in which the original is located. Additionally, a second OCR process could be started now to improve the results of the initial OCR pass.

3.5 Retrieval

As we have emphasized before, the crucial part in oblivious document capture is that no user interaction is required at runtime. The system works continuously in the background, even if the PC happens to be used for other tasks at the same time. However, the user of course also needs a possibility to access the archive of captured documents. We have designed a very simple prototype user GUI for this. Assuming that the biggest advantages of electronic versus paper documents is their capability for fast and flexible search, we concentrate on this aspect here.

In the initial version, a simple but effective full text search is implemented with the additional possibility to specify an interval for date and time of capture to which the search is limited. After a successful query, all documents containing the search expression are presented in thumbnailed image form, additionally highlighting the search string using a colored box. Figure 8 shows a screen-shot of the simple search interface. We chose to present the image version and not the text version of the document to the user, because it might contain more information than the OCR had been able to extract, e.g. images or handwritten remarks.

Note that in a production system, special more emphasis would be placed on tailoring the text search to the special needs encountered here. Since we know that the text resulted from an OCR process, we would use a word similarity measure that takes into account possible errors and enables us to weight various mismatches. For example, the confusion of the lower-case letter ‘l’ with an upper-case letter ‘I’ would receive less penalty than other mismatches in the string matching.

Once a document has been selected for a larger view, it is possible to navigate to those which were captured directly before or afterwards, using ‘forward’ and ‘backward’ buttons that users are familiar with from their web browser. This makes is possible to access all pages of a multi-page document in the same order they were read the first time the user had them on his desk.

4 Summary

We have described a prototype system that can perform document capture and archiving without any user interaction. The user’s desktop is monitored and whenever a new document (here, a roughly rectangular and reasonably sized object) is detected, a high resolution image is captured. The textual content of the document is extracted using a special OCR component that can deal with the distortion that camera captured document images typically show. In addition, background threads are started to create an image version free of these distortions. Both the image and the textual content are archived, ready to be accessed by the user at any later time using a customized GUI application that allows full text search.

The resulting system is able to support every day office work-flow without interrupting it by taking a step into the direction of full text search for documents printed on paper.

Apart from its functionality, the main advantage of the proposed system is its modular setup. Each step described in Section 3 is designed as a separate module that can easily be replaced by an improved one. Also, more dewarping modules than the two implemented so far can be integrated, e.g. specializing on different classes of document types.

5 Current Status and Future Work

At the moment, the system is still in the development stage. All individual modules were implemented and tested independently. The connecting data interfaces will be aligned soon, such that the system can then work completely without user interaction.

The system was designed to be a proof-of-concept. Therefore, some improvements will be necessary before it can be considered to be of practical use. The first issue is that the resolution of the digital cameras chosen seems not high enough. At least 200 dpi at document level are desirable, even better 300 dpi, and the current system cannot yield more than 100 dpi when monitoring a full desktop surface. Another practical problem is that the delay between triggering a capture and actually obtaining the data currently is in the range of a few seconds and therefore somewhat too large. This is mainly due to slow camera action and data transfer, partly also caused by the *Canon Powershot* cameras still using the old USB1.1 standard. Switching to newer and higher resolution cameras is therefore a high priority on our list of improvements.

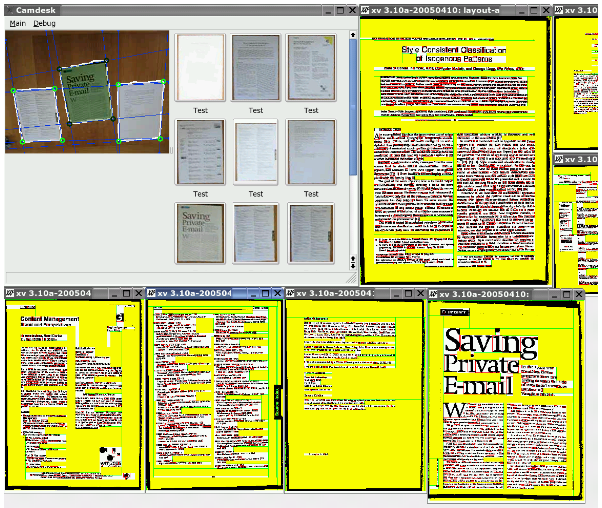


Figure 9. Screen-shot of a debugging output showing various captured documents that have undergone a first layout analysis step, i.e. the detection of white space.

On the software side, the current surveillance module should be made aware of the document content instead of only the shape of its outline. That way, a document that has already been captured would not have to be captured again, and a new document that is found at exactly the same position as the previous one (e.g. different pages of a book) will be detected more reliably.

We also plan to address another use of the document database that results from the captured images, i.e. content-based access. We will integrate both image-based and text-based retrieval of similar documents in addition to the keyword search already possible.

The field of document image dewarping is a very active field of research, and we plan to improve our algorithms in that field as well. Our special aim is to make the approach using only one camera better suitable for complex documents that contain more than just text. This will in particular require a module for separating text from images and for layout analysis, both of which are currently under development. Figure 9 shows a screen-shot of the results of a first layout analysis step, the detection of white space in the document images.

Acknowledgments

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project IPeT (01 IW D03).

References

[1] American National Standards Institute. Electronic still picture imaging - Picture Transfer Protocol (PTP) for Digital Still Photography Devices. ANSI/PIMA 15740:2000.

[2] H. S. Baird. Document Image Defect Models. *Document Image Analyis*, pages 315–325, 1995.

[3] T. Braun. Camdesk - Towards Easy and Portable Document Capture. Technical Report, Robotic Systems Group / IUPR, Technical University Kaiserslautern, Germany, 2005.

[4] T. M. Breuel. Robust Least Square Baseline Finding using a Branch and Bound Algorithm. In *Proc. of the SPIE – The Int. Society for Optical Engineering*, pages 20–27, 2002.

[5] M. S. Brown and W. B. Seales. Document Restoration Using 3D Shape: A General Deskewing Algorithm for Arbitrarily Warped Documents. In *Int. Conf. on Computer Vision (ICCV01)*, volume 2, pages 367–374, Vancouver, Canada, July 2001.

[6] H. Cao, X. Ding, and C. Liu. Rectifying the Bound Document Image Captured by the Camera: A Model Based Approach. In *7th Int. Conf. on Document Analysis and Recognition (ICDAR2003)*, pages 71–75, Edinburgh, UK, August 2003.

[7] D. H. Douglas and T. P. Peucker. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Caricature. *The Canadian Cartographer*, 10(2), 1973.

[8] T. Kawashima, T. Nakagawa, T. Miyazaki, and Y. Aoki. Desktop Scene Analysis for Document Management System. In *10th Int. Workshop on Database & Expert Systems Applications (DEXA '99)*, pages 544–548, Washington, DC, 1999.

[9] J. Liang, D. DeMenthon, and D. Doermann. Flattening Curved Documents in Images. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR2005)*, volume II, pages 338–345, June 2005.

[10] W. Newman, C. Dance, A. Taylor, S. Taylor, M. Taylor, and T. Aldhous. CamWorks: A Video-based Tool for Efficient Capture from Paper Source Documents. In *IEEE Int. Conf. on Multimedia Computing and Systems*, volume 2, pages 647–653, June 1999.

[11] M. Pilu. Undoing Page Curl Distortion Using Applicable Surfaces. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR2001)*, pages 67–72, Kauai, HI, December 2001.

[12] P. Simard. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. In *7th Int. Conf. Document Analysis and Recognition*, pages 958–962, Edinburgh, UK, August 2003.

[13] S. Suzuki and K. Abe. Topological Structural Analysis of Digital Binary Images by Border Following. *Computer Vision, Graphics, and Image Processing (CVGIP)*, 30(1):32–46, 1985.

[14] A. Ulges. Indexing and Recognition of Documents Captured with a Handheld Camera. Master's thesis, Technical University Kaiserslautern, Germany, 2005.

[15] A. Ulges, C. H. Lampert, and T. M. Breuel. Document Capture using Stereo Vision. In *Proc. of the ACM Symposium on Document Engineering*, pages 198–200, Milwaukee, WI, 2004.

[16] A. Ulges, C. H. Lampert, and T. M. Breuel. Document Image Dewarping using Robust Estimation of Curled Text Lines. In *8th Int. Conf. on Computer Vision and Pattern Recognition (ICDAR2005)*, Seoul, South Korea, August 2005. In press.

[17] P. Wellner. Interacting with Paper on the DigitalDesk. *Commun. ACM*, 36(7):87–96, 1993.

[18] C. Wu and G. Agam. Document Image Dewarping for Text/Graphics Recognition. In *SPR 2002, Int. Workshop on Statistical and Structural Pattern Recognition*, volume 2396 of *Lecture Notes in Computer Science*, pages 348–357, Windsor, Ontario, Canada, August 2002.

[19] A. Yamashita, A. Kawarago, T. Kaneko, and K. T. Miura. Shape Reconstruction and Image Restoration for Non-Flat Surfaces of Documents with a Stereo Vision System. In *17th Int. Conf. on Pattern Recognition (ICPR2004)*, volume 1, pages 482–485, Cambridge, UK, 2004.