Diplomarbeit
von Dietmar Aust
3.9.1996

Augmenting Paper Documents with Digital Information in a Mobile Environment

Lehrstuhl VII
Fachbereich Informatik
Universität Dortmund

Betreuer der Diplomarbeit:

Prof. Dr. Heinrich Müller
Prof. Dr. Wolfgang Banzhaf

Contents

Chapter 1

Introduction

1.1 Motivation

Reasoning about knowledge and its acquisition, Dr. Vannevar Bush was the first to detail the basic concepts of a Hypermedia system [3]. Based on the technological state of the art in 1945, he envisioned a system called *Memex* where information is accessible not only through an index but also through *associations* to related information. The Memex system should enable the user to create his own personal associations between related pieces of information, what Dr. Bush considered to be a crucial step in the acquisition of knowledge. In order for the Memex system to become a reality, Dr. Bush assumed that in the future all information would be stored in some digital form.

But paper documents have not been replaced by electronic alternatives yet. Even more, the paperwork in the office is growing at a rate of 20% annually [22]. This is partly attributed to the following factors:

- It is still easier to read on paper than on a computer screen [11].
- Browsing, navigating and manipulating large documents is easier with paper than with computer-based documents [6],
- People get a better 'sense of the text' when working with and reviewing paper documents [11].

This evolution has led to the current situation, where our information is stored in a *hybrid* way, partly in paper form and partly in electronic form.

Different associations between pieces of information: Between paper documents and electronic information, four different types of associations are possible:

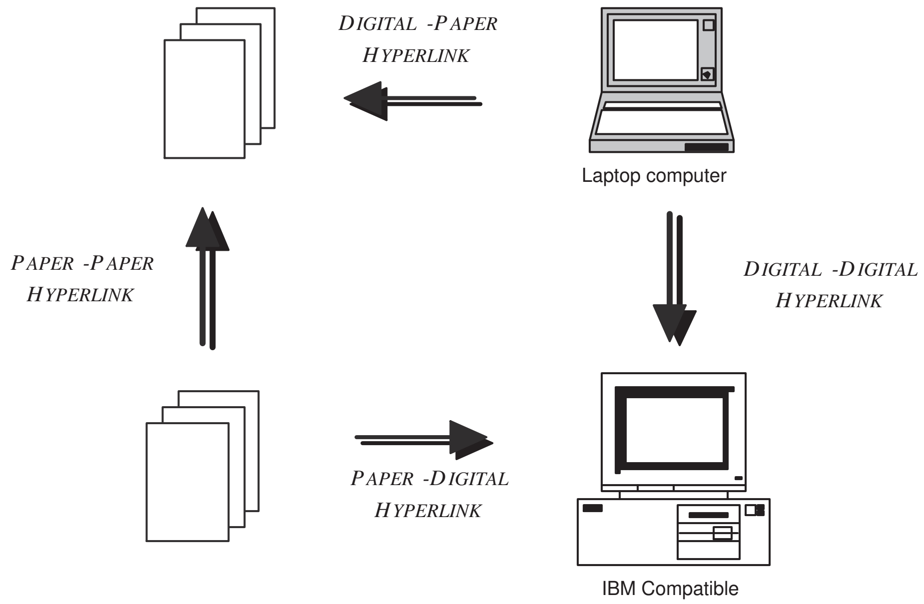

Figure 1.1: Different associations between paper documents and electronic information

1. *Paper-paper hyperlink*, e.g. from a conference paper to another paper document or book it references.

2. *Paper-digital hyperlink*, e.g. from a printed document to its online version.

3. *Digital-digital hyperlink*, e.g. multimedia hyperlink from a webpage to associated video clips, soundfiles and other webpages.

4. *Digital-paper hyperlink*, e.g. from a computer file to its printout

Hybrid working sets of documents: In our daily office work we use the information in books, other paper documents, the World Wide Web (WWW) and online databases. We use the computer to create graphics, diagrams, flowcharts and multimeda presentations. Sometimes we print our electronic documents to annotate or review them. After the review, we update the electronic version. We often retrieve the digital equivalent of its corresponding paper version and therefore switch between different media types. Thus, we have to deal with a *hybrid working set of documents*, consisting of paper documents as well as electronic information.

Unfortunately, the current technology doesn't allow us to put paper and electronic information into the same place. The two types must be arranged separately. Due to the different characteristics of the physical world and the computer such as three-dimensional space as opposed to the 'flat' filesystem, different strategies for organizing information are deployed. The user has to maintain the associations between related pieces of information in the different domains.

Recently, new technologies have been proposed for solving the above mentioned problems. They are based on the concept of *Computer Augmented Reality (AR)* [24], where computers are integrated in our environment to support and enrich the interaction with everyday objects.

1.2 Thesis focus

In this thesis, a Computer Augmented Reality (AR) system for the integration of paper and electronic documents is proposed. The system is designed to provide a more intuitive and natural way of annotating physical paper documents with related electronic information using Computer Vision techniques.

The detailed requirements for the conceptual framework and the computer vision system will be identified in 3.2 and 3.3.2.

1.3 Overview

Chapter two gives an introduction to the notion and the ideas of Computer Augmented Reality. It also presents the related work in the domain of paper based AR-systems and discusses their approaches, results and limitations.

In Chapter three a scenario-based approach is utilized to explain the problems with the current situation of maintaining relations between paper documents and related electronic information. Requirements are derived for creating a framework to provide a better integration and more efficient means of handling such hybrid sets of working documents. Based on the derived requirements, the *PaperLink* system is proposed. To establish a paper-digital hyperlink, a highlighting marker and computer vision techniques are deployed.

Chapter four details the algorithms selected and implemented for the computer vision system.

The actual design and implementation of the computer vision system is discussed in Chapter five.

Chapter six concludes and summarizes the thesis and proposes some ideas for further research in this area.

Chapter 2

Computer augmented reality and related work

2.1 Computer augmented reality

Current user interface paradigms such as the *"Windows, Icons, Mouse and Pointing device"* (WIMP) or the *desktop metaphor* are designed for human computer interaction using a desktop computer.

Most of the time we have to sit down in front of the workstation to access computer functionality and become therefore isolated from the real world. The computer remains too often the focus of attention.

Currently, computers are unaware of the real world context and thus provide little support for everyday tasks in the physical world.

Interaction techniques with electronic objects are still not as rich and easy as with their physical counterparts. In the physical world we have learned to construct spatial and temporal relationships between objects, make use of khinestetic memory, manipulate objects simultaneously using both hands, etc. This knowledge is deeply embedded into our minds. But to access the functionality of computers we have to sacrifice some of these skills and learn from scratch how to move and manipulate objects in the digital domain.

Augmented Reality is an approach for merging computers with the physical world without attempting to replace it.

It can be defined as: *Accepting the primacy of the physical world and the construction of appropriate tools to enhance our everyday tasks.* [24]

Characterizing this approach by contrasting it to Virtual Reality (VR) was often done in the literature. The main difference relative to VR is the fact that AR systems try to embrace the physical world and the interaction in it is considered the primary focus of attention. Enhancing and supporting the interaction in the physical world by using computers is the main goal of Augmented Reality.

But the concepts of Augmented Reality are really twofold. Not only can AR systems provide us with richer interaction techniques in the physical world, it also lets us utilize our physical abilities to manipulate digital objects. We can use khinestetic memory, visual cues, etc. to become more efficient in manipulating electronic information.

Implicit commands: Nowadays, our environment is unaware of us, who we are, what mood we are currently in and what our preferences are. Therefore it cannot provide any context-based support.

Instrumenting the environment with sensors, cameras and motion detectors , would make the environment 'aware' of the users and their current context. Issuing an *implicit command* means that the user does not have to specify the action he wants the machine to carry out, but instead the machine can infer from the user's current context what 'the right thing to do' is.

Integration of digital information with real world objects and events: Currently, there is electronic information available on everyday objects, like online train schedules, library databases, information about art pieces, etc. An increasing variety of information is available on the World Wide Web (WWW). Many publishers now package CD-ROMs with their books.

But the association between the physical artifacts like a train and the electronic information such as the online train schedule has to be made in one's mind. This requires a cognitive effort each time a connection must be established between the two domains. Associating a physical artifact to the electronic information is not easy and a natural and simple user interface must be provided.

The integration of digital information with real world objects can be achieved by creating 'hot-spots' on real world objects or by literally overlaying the information on top of the physical world.

2.1.1 Ongoing research

Early work: As one of the pioneers in this area, Myron Krueger had reasoned already in the late 60's [16] that the ultimate computer interface would be to the human body and its senses. This interface should perceive the user rather than receive input from him. The technology he created used the notion *come as you are*. He didn't require people to wear any specialized equipment as opposed to VR. He called this approach *computer-controlled responsive environments*. He used this technology mainly for artistic purposes.

Ubiquitous Computing: The idea of *Ubiquitous Computing (Ubicomp)* originated at Xerox PARC in 1988 [23]. One would enhance the use of computers by making many computers available throughout the physical environment making them effectively invisible to the user. The goal was to turn computers into appliances.

In the environment one would find many computers of different sizes and shapes, ranging from very small and lightweight post-it notes such as the PARC-TAB [23], over mid sized tablets up to wall-sized displays such as the Liveboard [23]. Such wirelessly connected computers would work together to help us keep track of projects and keep us in touch with our colleagues. The environment would track the user through heat, motion and other types of sensors to automatically adjust the temperature or update the computer displays according to the people in the room.

Context-sensitive information devices: Many research efforts arose around the common theme of *context-sensitive information devices* , i.e. being able to provide additional information on your environment based on one's current context. The simplest, but so far most common approach is the use of location information about the user and so called 'hot-spots' in the environment. As soon as the user comes close to a 'hot-spot', related information would be displayed. Some recent work on context-sensitive information devices can be found in [10], [21] and [17].

Overlayed and registered information: Often, when there is additional digital information available, computers can do the registration with objects in the physical world, overlay the information using see-through goggles and therefore facilitate the task at hand. These systems can provide us with some sort of *x-ray vision* of the world. Field workers can locate parts within laser printers [9], surgeons can see x-ray or ultrasonic images overlayed on top of the patient, etc.

2.2 Related work in paper based systems

2.2.1 General areas and thesis focus

Having accepted the advantages of both media types, researchers developed systems to provide support in a hybrid working environment.

The main goals of the current research projects can be stated as:

1. Augmenting the current interaction with paper.
2. Integration of paper documents with related electronic information.
3. Providing more intuitive user interfaces utilizing the familiarity with the manipulation of paper documents.

To achieve the above mentioned goals, the currently existing systems use the following approaches, with a focus on one or more of them:

- *Making relations (hyperlinks) between paper documents and related electronic information.*
  Ariel [18] is based on the DigitalDesk technology [25] and centers around CAD engineering drawings, allowing the user to attach multimedia annotations to the paper version.
  The InteractiveDesk [1] supports these multimedia annotations to paper documents in a more general setting.
- *Projecting some features of digital documents back onto paper, such as cut, paste, copy and search.*
  The DigitalDesk [25] enables the user to select, copy and paste sketches on paper utilizing a regular stylus.
- *Using our physical skills of manipulating objects to solve the problems with manipulating digital documents.*
  The DigitalDesk supports to move around projected electronic information with a regular pen.
- *Using paper as a computational user interface.*
  Protofoil [20] is an example of providing a pure form based paper user interface to computer based filing, retrieval and OCR. Using Mosaic [19], people are enabled to issue computer commands showing paper cards to a camera.

The main focus of this thesis is the *Integration of paper documents with related electronic information by providing paper-digital hyperlinks.*

2.2.2 Discussion of the related work

2.2.2.1 Ariel

Ariel [18] is part of a three-year European ESPRIT project called EuroCODE, which provides an object-oriented, open environment for the development of distributed, multimedia cooperative work applications. Interviewing engineers responsible for the construction of the world's longest suspension bridge in Denmark, one major problem was investigated by the researchers: Making sure that the engineering drawings reflect the real state of affairs at the construction site.

They found that the engineers tend to work away from their desks and their PC's. They take a set of drawings out on site and use them to discuss problems and progress. They record comments using dictaphones or document problems with photographs and video footage. But they also make frequent changes by hand directly on the drawing. All of the personal changes to the paper CAD drawing result in discrepancies to the approved online CAD version.

Their prototype system *Ariel*, allowed the engineers to use their personalized paper version of the online CAD drawing as an interface to informal multimedia annotations. The technology is based on the DigitalDesk [25].

The system is capable of:

- Relating an individual engineering drawing to its online version.
- Capturing and respondonding to user commands.
- Recording information supplied by the user (including sketches, hand-written notes, audio and video clips) and 'attaching' it to any location on the paper drawing.
- Projecting information such as user menus, visual cues as to where augmentations exist (hot spots) onto the paper drawings.
- Providing live audio/video communication.

Limitations: Having based the system on the DigitalDesk technology, using a projection device and a camera mounted above the desk, certain constraints are imposed on the system. The user is limited to that physical desk, where the paper drawing is augmented with multimedia information. All enhancements can only be used in conjunction with that specialized desk. Taking the drawing to a different location like a meeting room for example, the drawing looses all augmentations immideately.

The use of an overhead projector to superimpose information on top of the drawing, makes the system unuseable in broad daylight.

The system does not provide support for the digital-paper hyperlink.

2.2.2.2 InteractiveDesk

Another computer augmented desk, called the *InteractiveDesk* [1], was developed at the Hitachi Research Laboratory. It uses a camera mounted above the desk to capture information and a projection device to output information on the desk.

They focus on creating and maintaining a *hybrid working set of documents*, consisting of paper documents as well as related electronic information. Attaching a printed color coded tag to a physical file folder or paper document enables the system to identify that physical paper document. The user can then interactively attach electronic information to it.

Limitations: The system is not portable. Using a fixed desk to provide the ability to augment paper documents, the user is again bound to that desk and cannot access the additional functionality anywhere else.

Printing the color coded tags and glueing them to the file folder or paper document is still quite cumbersome and takes time. It also limits the *resolution of augmentations*. It is not possible to relate information to separate keywords in a paragraph, since the color coded tags are not transparent and therefore occlude the text partially.

The system does not provide support for the digital-paper hyperlink.

Chapter 3

System requirements and proposed solution

3.1 Problem scenario

To motivate the requirements and the applicability of the proposed system called *PaperLink*, consider the following possible scenario:

John Smith prepares a conference paper for submission to the conference on Human Factors in Computing Systems (CHI'97), held in Atlanta.

In the process of doing this, three phases seem to be significant:

- *Gathering relevant information.*
  Mainly at the beginning of a project, relevant information in related conference papers and books is gathered. During the project only few new references or other relevant information is added.
- *Annotating, adding to and updating the conference paper.*
- *Retrieving information.*

Gathering relevant information: John goes to the library to search for related work in his area that he can use as a reference. He makes xerox copies of all relevant conference papers. Coming back to his office, he wants to find out more about the relevant papers and their authors. Thus he visits their homepages and the homepages of their departments. He jots down their URLs on his personal copy of their paper and creates bookmarks using his web browser for later retrieval.

He visits the CHI-homepage to learn about the submission deadline and the required layout of the conference paper.

Annotating, adding to and updating the information: In the process of writing, John sometimes prints out the document to review it. He is collaborating with Jim on that paper. John goes to Jim's office to discuss the first draft version. They review the paper together, discussing each paragraph. First they briefly review it paragraph by paragraph in a linear order, later they randomly add remarks where necessary. He uses his dictaphone to record comments in addition to the handwritten remarks.
What if John could directly attach the voice annotation to the relevant paragraph?

Returning to his office, he starts his word processor, opens the desired file in a certain subdirectory and starts entering the changes.
What if John could directly start the word processor opening the desired file automatically in one step?

Retrieving information: From time to time he checks the ACM website to get updated information on the conference, the hotels and on Atlanta itself. *What if he could start Netscape with the correct URL using his conference paper as an interface?*

While working on his prototype application, John sometimes posts questions to newsgroups related to programming in C++.
What if he could start the newsreader with the relevant newsgroup using his conference paper as an interface?

Later, John looks up the details of an algorithm in one of his own books. He recalls that he had once implemented that algorithm, but doesn't know exactly anymore, where on the filesystem the file resides. The last time used the file is already six months ago.
What if the code fragment was 'attached' to that section in the book?

3.2 Requirements

Analyzing the given scenario one can derive requirements for a system to support the integration of paper documents with related electronic information to overcome the inherent problems with relating information in different media types.

Static paper-digital hyperlinks: Paper-digital hyperlinks, embedded into each printed document would allow the instant retrieval of the original online version. Invoking the paper-digital hyperlink the document would be loaded into the application which was used to create the document.

In cases where publishers package CD-ROMs together with books or manuals, paper-digital hyperlinks could be embedded to allow the instant retrieval of the related information.

Personalized paper-digital hyperlinks: The user should be enabled to *interactively* create personal paper-digital hyperlinks between paper documents and their related electronic information.

This is in addition to the static paper-digital hyperlinks to allow the user to add digital information to the current hybrid working set of documents.

Creating and invoking links in a natural and easy way: The provided system should be easy to use. This is crucial for the acceptance by the user.

Portability The user should be able to access the augmentations wherever needed, in the office, at home and in meeting rooms. The enhanced interaction should not be limited to a specialized desk or room.

High resolution of augmentations The system should support a high resolution for the augmentations, enabling the user to create paper-digital hyperlinks from different paragraphs or even individual key words.

3.3 Proposed solution

The proposed system *PaperLink* consists of two parts. One is the conceptual framework for augmenting paper documents, the other is the computer vision system to effectively recognize a mark on paper using a camera.

The *PaperLink* system fulfills the in section 3.2 derived requirements, yet does not provide support for *static paper-digital hyperlinks*.

The following sections describe the components of the system, the required hardware and the interaction schemes to create and invoke the links. The expected use of the system is described utilizing an application scenario. Considering the functionality and the deployed hardware of the proposed system, specific requirements for the computer vision system are derived.

3.3.1 The conceptual framework

To interactively embed digital information into paper documents one could utilize the following hardware components (fig. 3.1):

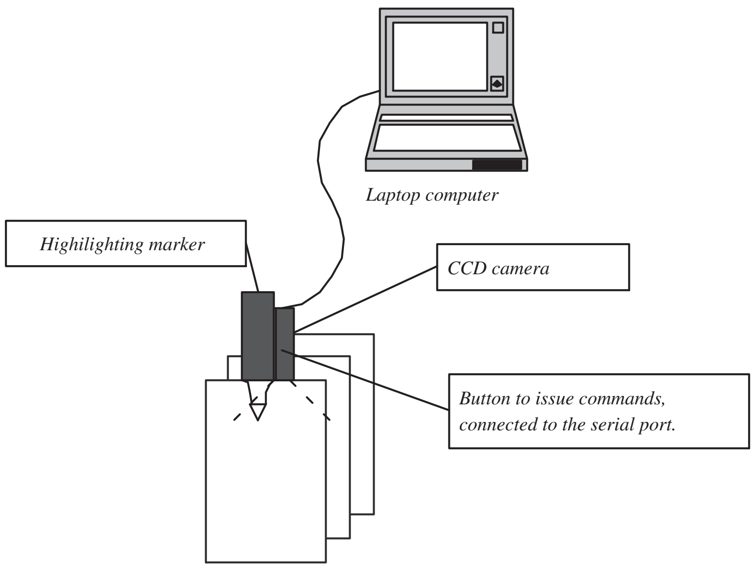

Figure 3.1: The hardware components: The highlighting marker with a camera attached to its side, connected to a portable computer

- a highlighting marker,

- a small and lightweight camera with a microlens attached to the side of the highlighting marker,
- a button on the side of the CCD camera, connected to the serial port of the computer and
- a portable computer.

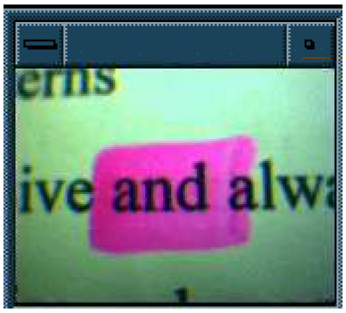

Figure 3.2: Colored mark on paper

Creating a paper-digital hyperlink: To create a paper-digital hyperlink, digital information has to be embedded into the paper document to serve as an 'anchor' to attach electronic information to. This is done by making a *colored mark* (fig. 3.2) on a paper document by highlighting an arbitrary piece of text. The text can be either printed or handwritten.A laptop computer can provide the electronic information to augment the colored mark with. All information available in electronic form, such as documents, spreadsheets, searchable databases or information in the WWW can be viewed as *digital augmentations* (fig. 3.3).
Associating digital information to a colored mark converts the colored mark into an *augmented mark*. There is only a semantic difference between the two. The connection between the augmented mark and the digital information itself is called a *paper-digital hyperlink*.

Invoking a paper-digital hyperlink: Once a paper-digital hyperlink from a paper document to digital augmentations in the computer has been created, the link can be easily *invoked* by pointing with the camera to the augmented mark and pressing the button on the camera. The colored mark is then identified by the system and the corresponding digital augmentation is retrieved and displayed.

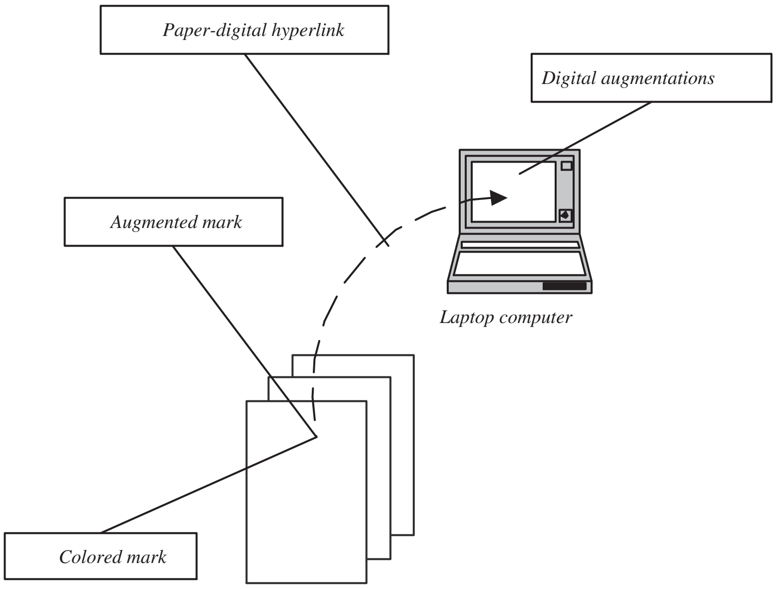

Figure 3.3: Illustration of the semantics

Rationale for using the highlighting marker: The highlighting marker is ubiquitous and already part of the infrastructure for work with paper documents. It is used to mark important keywords and concepts. Creating the paper-digital hyperlink by actually leaving a visible mark on the paper serves as visual feedback for locating augmented marks.

3.3.2 Requirements for the Computer Vision system

Considering the desired functionality of the proposed system, specific boundary conditions for the Computer Vision system to classify the different colored marks can be derived.

Since especially in the field of Computer Vision most of the systems built and the algorithms deployed are tailored so much to solve a very specific, clearly restrained problem, first of all we have to define the problem at hand in detail.

Functional requirements of the system:

- Adding a new colored mark to the system's knowledgebase by computing and storing its characteristics.
- Identifying (classifying) a previously stored colored mark.

The following boundary conditions are expected to be fulfilled:

- The patterns are all two-dimensional.
- The patterns are non-occlusive and always completely visible within the given image.
- There is only a limited amount of rotation.
- The time for adding a new pattern or classifying an existing pattern should be below 2sec.
- Only one image will be available per colored mark.

3.4 Application

To demonstrate the expected use of the system, consider the following application scenario:

3.4.1 Managing a hybrid working set of documents

This is a very general application scenario applicable for many different settings. For illustration purposes, consider how the scenario from section 3.1 would be differerent using *PaperLink*.

Gathering relevant information: John goes to the library to search for related work in his area that he can use as a reference. He makes xerox copies of the relevant conference papers. Coming back to his office, he wants to find out more about the relevant papers and their authors. Thus he visits the authors' homepages and the homepages of their departments. To facilitate the later retrieval, John attaches the authors' homepages to his personal copy of their conference paper. He uses the highlighting marker to highlight a small part of each authors name, points with the camera to the colored mark and presses the button on the camera. When the button is pressed, the PaperLink system tries to identify the current colored mark. Since this colored mark is not known to the system yet, the colored mark is then added to the current knowledge base. When prompted, John uses the keyboard to enter the URL and specify the type of augmentation as being an URL.

John creates a *project sheet* for this project. A project sheet holds augmented marks for the overall project, e.g. newsgroups and databases. On this project sheet he also creates a colored mark and associates the electronic version of his conference paper to it. Another colored mark is associated to a newsgroup on programming in C++.

Annotating, adding to and updating the information: In the process of writing, he prints out the document to review it. Since John is collaborating with Jim on this paper, he walks over to Jim's office to discuss the first draft version. He takes the PaperLink system with him, consisting of his laptop, the highlighting marker with the camera and a microphone, both connected to the laptop. They review the paper together, discussing each paragraph. First they briefly review paragraph by paragraph in a linear order, later they randomly add remarks where necessary. When John wants to make a verbal comment on a certain paragraph, he creates a colored mark on the first word of the paragraph, then presses the button on the camera. When prompted, John chooses to create a voice annotation. As soon as he presses the space bar, the system starts recording. To stop the recording, John presses the space bar again.

Back in his office, he wants to make the changes to the file as discussed in the meeting with Jim. He therefore takes out the *project sheet*, points with the camera to the augmented mark to which the word document is associated and presses the button on the camera. The system automatically retrieves the requested file and loads it into the word processor it was created with.

Retrieving information: From time to time he checks the ACM website to get information on the submission deadline, the layout requirements for the paper, infos on hotels and on Atlanta itself.

When actually coding, he looks up the details of an algorithm in a book on his shelf. He recalls, that he had once implemented that algorithm and created an augmented mark for that code fragment. He therefore points with the highlighting marker to the augmented mark, then presses the button on the camera. The PaperLink system automatically starts the C++ development environment and open the requested file.

Chapter 4

Components of the vision system

A generic computer vision system consists of the components Image Acquisition, Segmentation, Feature Extraction and Classification.

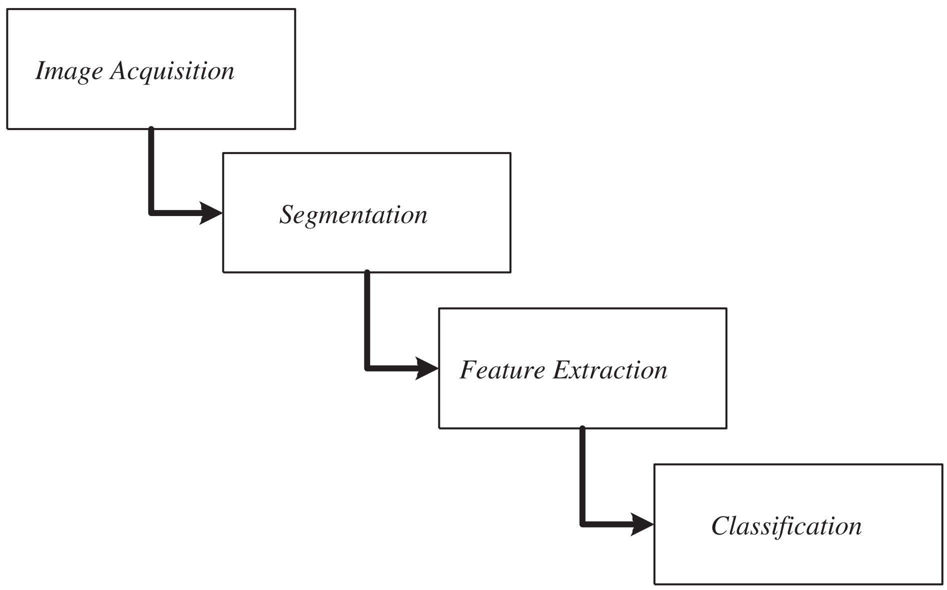

Figure 4.1: Components of a generic vision system

In the *Image Acquisition*, video images from a camera or a different video source are digitized by the capture hardware.

The *Segmentation* process then converts this raw image data into sets of pixels corresponding to objects in the image. Pixels with similiar properties are grouped together, based on either color value similarity or spatial proximity of the pixels themselves.

The *Feature Extraction* then computes features on those pixel sets to characterize each object in a unique way. Features can characterize the texture or the shape of an object. Features can correspond to local or global attributes of the objects. The result of the Feature Extraction is usually a vector of numerical values characterizing an object.

During *Classification* each object is assigned a label corresponding to a prototype object stored in a predefined set of objects.

The rest of the chapter details how each of those components are implemented by the proposed system.

4.1 Image acquisition

The color image data obtained from the capture hardware are 32-bit RGB values.

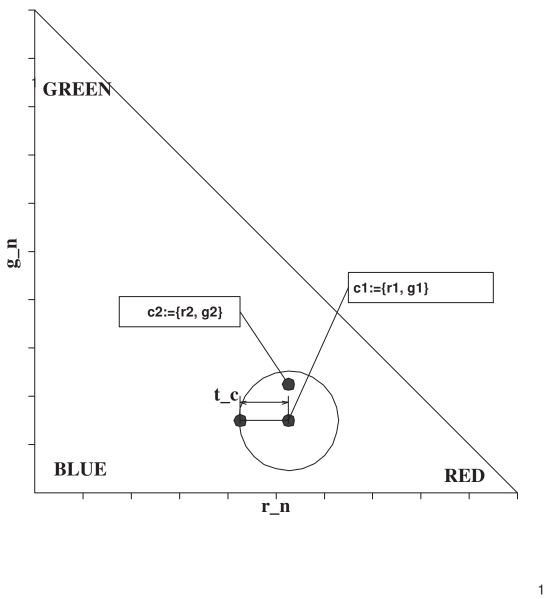

Figure 4.2: Normalized color coordinates

To obtain an illumination independent representation, the chromaticity coordinates [13] are computed:

$$r_n = \frac{r}{r+g+b}$$
$$g_n = \frac{g}{r+g+b}$$

$$b_n = \frac{b}{r+g+b}$$

Since r_n, g_n and b_n sum to unity, any two of them would sufficiently define any color in this color space. The normalized red (r_n) and green (g_n) components were chosen. Figure 4.2 shows a representation of the 2-dimensional normalized color space.

4.2 Segmentation

Due to the nature of the problem the outcome of the segmentation is more than one representation of the object of interest. Since each *colored mark* should be treated as a unique pattern, it can be interpreted as consisting of two separate parts which both characterize the colored mark in a unique way. One representation is the *highlighted text* and the other is the *colored region*. Only text that is completely enclosed within the colored region is considered to be highlighted.

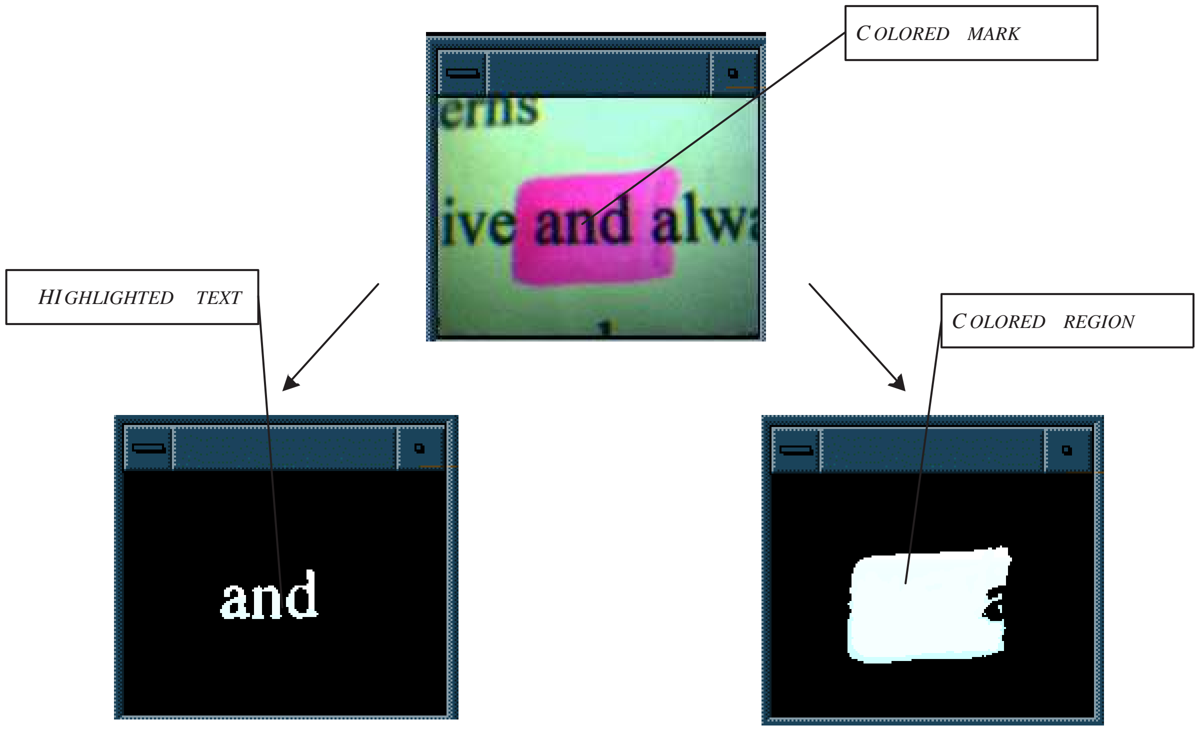

Figure 4.3: The colored mark consists of the highlighted text and the colored region

Therefore, the image segmentation in the proposed system will produce representations of

- the largest colored region
  - as a Run-Length-Encoding (RLE) of its binary image
  - as a list of 4-connected boundary pixels representing its boundary
- the highlighted text
  - as the raw binary image (used for the interface to other programs)
  - as the RLE of the highlighted text
  - as a list of subimages corresponding to each extracted character

The steps necessary to produce the described outcome are illustrated in figure 4.4 and detailed in the following sections.

4.2.1 Applying a color filter

The color image is described by a function $f_c(x, y)$ with

$$f_c(x, y) = \{r_n, g_n\}, \tag{4.1}$$

where r_n, g_n denote the normalized red and green component for the pixel at (x, y).

The color of the highlighting marker c_h is provided by the user. The similarity threshold t_c is a constant.

Applying a color filter to $f_c(x, y)$ produces a binary image $f_b(x, y)$ in which all pixels with value 1 correspond to pixels in f_c *similar* to the color of the highlighting marker c_h.

The similarity between two color values $c_1 = \{r_{n1}, g_{n1}\}$ and $c_2 = \{r_{n1}, g_{n1}\}$ is based on their euclidean distance d_c in the color space. Is the distance below a certain threshold t_c (fig. 4.2), the color values c_1 and c_2 are considered to be similar.

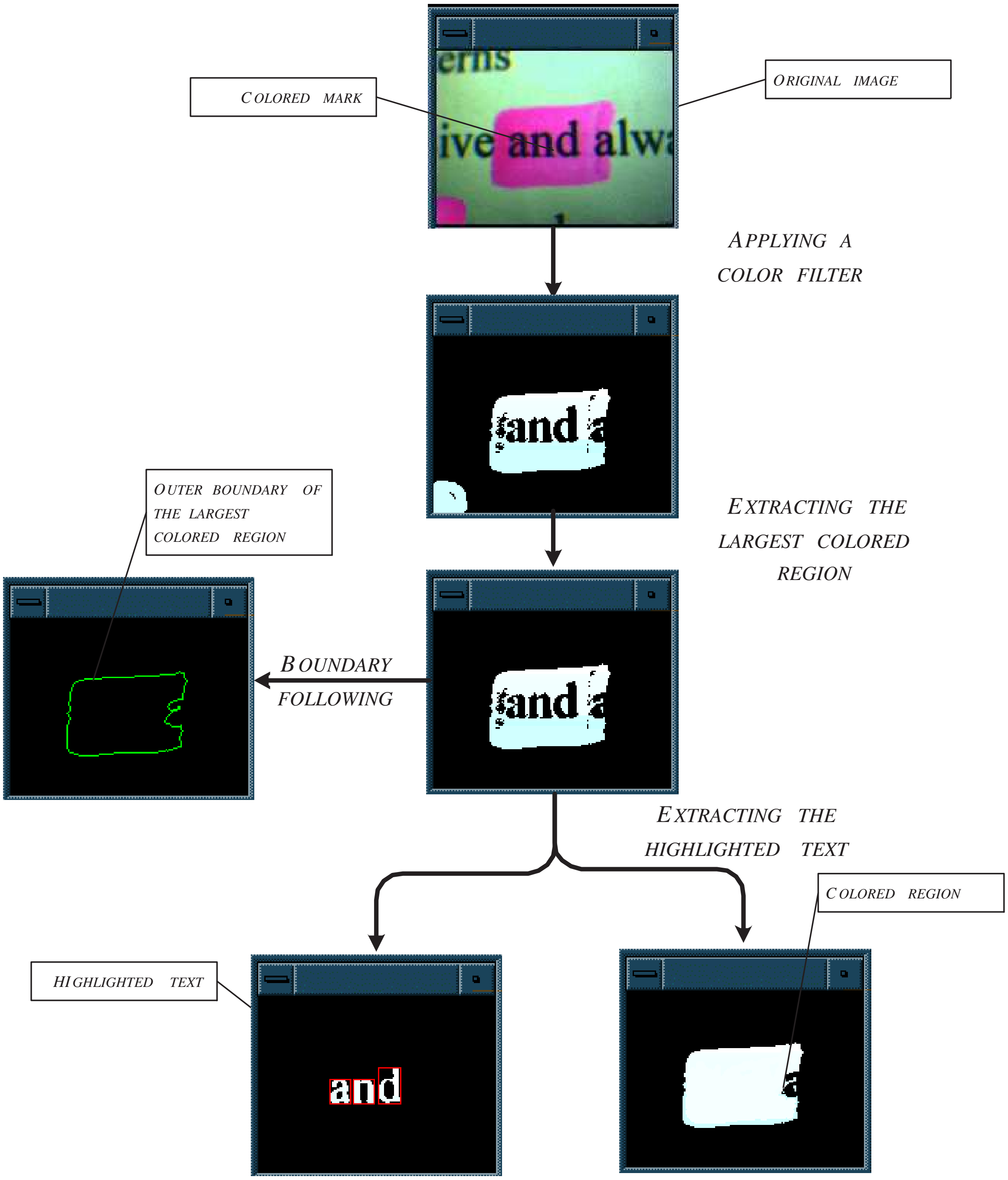

Figure 4.4: This figure shows the transformations of the original image as produced by the image segmentation

To cope with varying lighting conditions, the following thresholding algorithms were implemented:

Global thresholding with one adaptive threshold One global threshold is determined based on the color values in the current image. A two-dimensional histogram of the color values is computed based on a subsampling of the current image. Only colors *similar* to the selected color are considered for the computation of the histogram. The color corresponding to the maximum in the histogram becomes the adjusted selected color c_h.

Adaptive thresholding The whole image is subdivided into $m \times n$ tiles $tile_{ij}$. For each tile a local threshold t_{ij} is computed. During the actual thresholding, on each tile $tile_{ij}$ its corresponding local thresholds t_{ij} is applied.

Neither approach produced satisfactory results. The segmented images were very noisy. This was attributed to the fact that some black pixels of the highlighted text were falsely identified as being colored. The analysis of the color values of the highlighted text showed, that some of its black pixels took on the color of the highlighting marker. This appears to be an inherent problem when using a highlighting marker to color text on paper.

Further analysis of the grayscale image image indicated a solution. Applying a threshold to the grayscale image separates the text in the image from the background and the colored region. How to update the thresholded image to correctly obtain the highlighted text is shown in 4.2.3.

4.2.2 Extracting the largest colored region

To find the subimage containing the largest colored region, a *component labeling* is performed on the binary image. In this step all connected regions and their corresponding enclosing rectangles are computed. All subsequent operations are performed on the resulting subimage. This reduces the overall runtime. Other smaller colored regions within the current image are ignored.

The *component labeling* was implemented as described in [2] (there referred to as 'blob coloring'). Given a binary image $f_b(x, y)$ with

$$f_b(x,y) = \begin{cases} 1, & \text{pixel } (x,y) \text{ is colored} \\ 0, & \text{else} \end{cases}$$

of 4-connected components, the objective is to assign a label to each separate component. The function $f_l(x, y)$ assigning a label to each pixel is initially equal

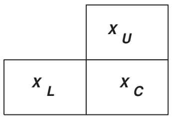

Figure 4.5: L-shaped template

to $f_b(x, y)$. The image is scanned from left to right and top to bottom with a special L-shaped template shown in fig. 4.5.

Alg. Component labeling:

Initially, set label l to 1. Scan the image from left to right and top to bottom.

```
if ( f_b(x_c) = 0 )
   then continue
else
   begin
     if ( f_b(x_u)=1 and f_b(x_l)=0 )
        then f_l(x_c) := f_l(x_u)
     if ( f_b(x_l)=1 and f_b(x_u)=0 )
        then f_l(x_c) := f_l(x_l)
     if ( f_b(x_l)=1 and f_b(x_u)=1 )
        then begin
               f_l(x_c) := f_l(x_u)
               f_l(x_l) is equivalent to f_l(x_u)
             end
comment: the two labels are equivalent
     if ( f_b(x_l)=0 and f_b(x_u)=0 )
        then f_l(x_c) := l; l := l + 1
   comment: new label
   end
```

4.2.3 Extracting the highlighted text

To compute the pixels representing the highlighted text, three major issues have to be considered:

1. Given the binary image $f_b(x, y)$, a black background pixel, a white background pixel and a black highlighted pixel are indistinguishable, since $f_b(x, y) = 0$

for all of them. Therefore simply inverting the binary image would not produce the correct result.

2. Due to the inherent problem of using a highlighting marker (discussed in 4.2.1), some of the pixels belonging to the highlighted text are falsely identified as being colored pixels.

3. Experiments showed, that the outer boundary can be reliably obtained. Just the inner boundary of the highlighted region is not correctly known at this point due to the aforementioned problem. Thus, scanline algorithms to compute the interior a polygon cannot be applied.

Therfore, the highlighted text is computed as follows.

The background of the image has to be marked to enable a distinction between the white and black background pixels as opposed to the black highlighted pixels.

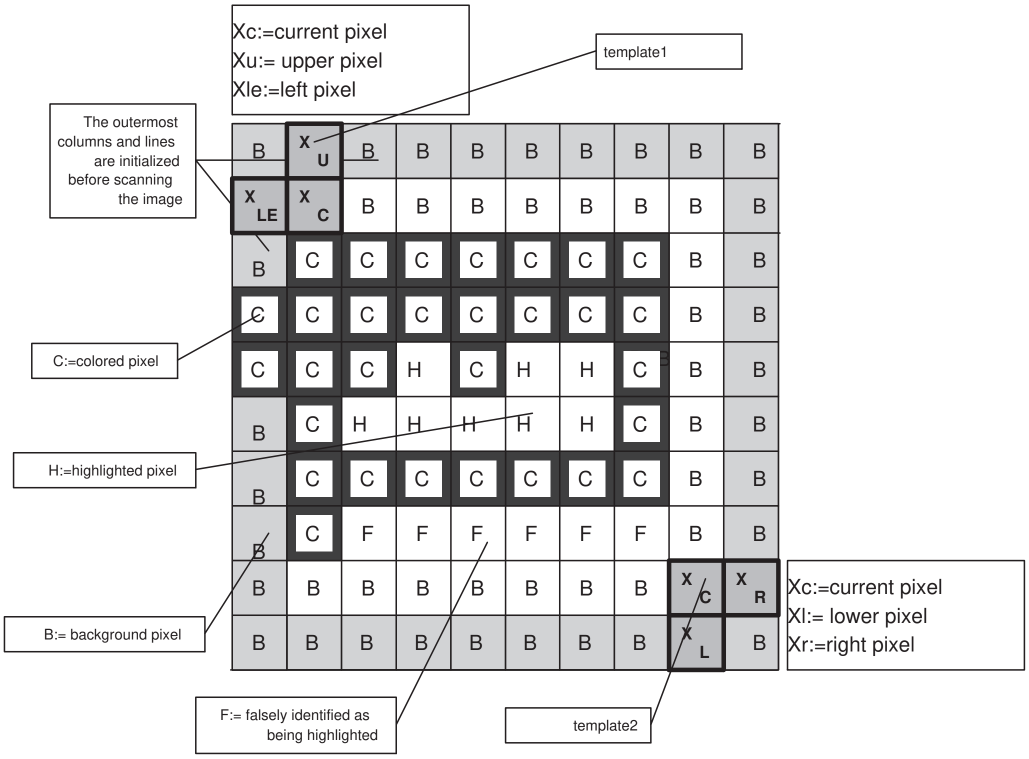

Figure 4.6: Extracting the highlighted text

First, the outermost columns and rows are initialized with either the symbol B for background pixels and the symbol C for colored pixels (fig. 4.2.3). The image is then scanned from left to right and top to bottom with template1. During this step, some background pixels, indicated by the symbol F, are falsely identified as highlighted pixels. Therefore, the image is scanned again, from right to left and bottom to top using template2.

At this point all pixels identified as highlighted are correct. Nevertheless, some of the highlighted pixels are identified as being colored. To correct that, all colored pixels whose grayscale value is below a certain threshold are added to the highlighted text. This exploits the fact that applying a threshold to the grayscale image separates all the text in the image from the background and the colored region, assuming that the text pixels are the darkest ones in the image.

While scanning the image, both the colored region and the highlighted text are stored using the RLE format.

4.2.4 Boundary following on the largest colored region

To compute the boundary of the colored region, the following algorithm [15] was used:

Algorithm: Boundary Following

1. Find starting pixel $s := (x_s, y_s)$ with $f_b(x, y) = 1$ for the region using a systematic scan from left to right and top to bottom of the image.

2. Let the current pixel in boundary tracking be denoted by c. Set $c := s$, and let the 4-neighbor to the west of c be $b := (x_b, y_b)$ with $f_b(x_b, y_b) = 0$.

3. Let the 8-neighbors of c starting with b in clockwise order be $n_1, n_2, n_3, n_4, \ldots, n_8$. Find n_i for the first i that is in S.

4. Set $c := n_i$ and $b := n_{i-1}$

5. Repeat steps (3) and (4) until $c = s$

4.3 Feature extraction

After the Segmentation an object is described by a set of pixels. The Feature Extraction is the process of computing numerical features on such pixel sets to characterize them in a unique way.

Considering the requirements for the system, the features have to be invariant to geometric changes such as

- Translation
- Scale and
- Rotation.

4.3.1 Simple global features

Eccentricity The eccentricity of a shape can be interpreted as the ratio of the maximum chord A to maximum chord B perpendicular to A (fig. 4.7), which is equivalent to the ratio of the sides a and b of the smallest enclosing rectangle for that shape.

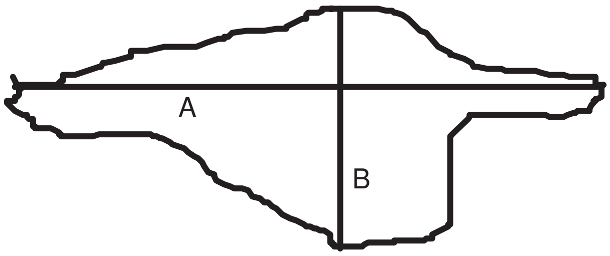

Figure 4.7: Eccentricity

An approximate formula based on the moments for the shape, as described in 4.3.3, can be given as [2]:

$$e = \frac{(M_{20} - M_{02})^2 + 4M_{11}}{area} \tag{4.2}$$

Compactness Another global feature characterizing the shape of an object is the ratio $(perimeter^2)/area$, which is dimensionless and minimized by a disk.

$$compactness = \frac{(perimeter)^2}{area} \tag{4.3}$$

Number of highlighted characters

Percentage The percentage denotes how large the extracted text is in relation to the size of the colored mark.

$$percentage = \frac{\text{area of the highlighted text}}{\text{area of the colored region}} * 100 \tag{4.4}$$

Distance between centroids The distance between the centroid of the colored region and the centroid of the extracted text characterizes the relative location of the extracted text within the colored region.

Dividing the squared distance by the area of the colored region, this measurement satisfies all invariance criteria.

4.3.2 Boundary signature

Signatures are projections of functions into a lower-dimensional space to reduce the complexity and provide a more compact representation. By definition, projections are non-information preserving transformations.

The following signature [4], characterizing the boundary of the colored region was implemented.

The boundary B of an object in the discrete case is denoted by a sequence of n connected points p_i. $B = \{p_i = (x_i, y_i) | 1 \leq i \leq n\}$

A *boundary function* Δ in the discrete case is proposed, where Δ_i denotes the distance from a point p_i on the boundary to the centroid $(\bar{x}, \bar{y})$ of the object.

Subsampling the boundary results in a reduced set of $n_s = 50$ equally distributed boundary points, which was found to be a reliable estimate for boundaries between 150 and 600 points.

Therefore, $\{\Delta_i : i = 1, \ldots, n_s\}$ denotes the sequence of distances to the centroid $(\bar{x}, \bar{y})$.

The three most stable features were implemented.

In the discrete case, $(\Delta_i - \Delta_{i+1})$ is used for the derivative of the boundary function.

Computing the variance of the derivative results in:

$$vfd = \frac{1}{A} \sum_{i=1}^{n_s} (\Delta_{i+i} - \Delta_i)^2 \tag{4.5}$$

The normalized variance of Δ is

$$var = \frac{1}{A} \sum_{i=1}^{n_s} (\Delta_i - \mu)^2, \tag{4.6}$$

where $\mu = \frac{1}{n_s} \sum_{i=1}^{n_s} \Delta_i$ denotes the average distance from the centroid.

Finally, the discrete version of the symmetry measure is computed as:

$$sym = \frac{1}{\sqrt{A}} \sum_{i=1}^{\frac{n_s}{2}} |\Delta_i - \Delta_{i+\frac{n_s}{2}}| \tag{4.7}$$

taking into account the relative location of opposing points on the boundary.

4.3.3 Moments

Starting from the area representation of an object, statistical measurements of the image points can be calculated which are called *digital moments.*

If the area description is represented by

$$f(x,y) = \begin{cases} 1, & \text{pixel (x,y) belongs to the object} \\ 0, & \text{pixel (x,y) belongs to the background} \end{cases}$$

then the digital moments m_{pq} of $f(x,y)$ [14] can be defined in the discrete case as:

$$m_{pq} = \sum_x \sum_y \; x^p \; y^q \; f(x,y). \tag{4.8}$$

where $(p+q)$ denotes the order of the moments.

The area of the object is defined by m_{00} and the centroid $(\bar{x}, \bar{y})$ can be computed as:

$$\overline{x} = \frac{m_{10}}{m_{00}} \quad \text{und} \quad \overline{y} = \frac{m_{01}}{m_{00}} \tag{4.9}$$

To obtain a translation invariant representation, the centroid is shifted to the origin. This defines the *central moments* as:

$$\mu_{pq} = \sum_x \sum_y (x - \overline{x})^p \; (y - \overline{y})^q f(x,y) \tag{4.10}$$

Normalizing the central moments with respect to the area of the object makes them scale invariant [5]:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}{}^{(\frac{p+q}{2}+1)}} \tag{4.11}$$

Finally, to obtain a representation which is invariant to rotation as well, the following non-linear combinations of the digital moments have been proposed. Combining the moments as suggested, removes the dependency on the angle of rotation.

$$
\begin{aligned}
\phi_1 &= \eta_{02} + \eta_{20} && (4.12)\\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\ {\eta_{11}}^2 && (4.13)\\
\phi_3 &= (\eta_{30} - 3\ \eta_{12})^2 + (3\ \eta_{21} - \eta_{03})^2 && (4.14)\\
\phi_4 &= (\eta_{30} - \eta_{12})^2 + (\eta_{21} - \eta_{03})^2 && (4.15)\\
\phi_5 &= (\eta_{30} - 3\ \eta_{12})\ (\eta_{30} + \eta_{12})\ [(\eta_{30} + \eta_{12})^2 - 3\ (\eta_{21} - \eta_{03})^2] && (4.16)\\
&\quad +(3\ \eta_{21} - \eta_{03})\ (\eta_{21} + \eta_{03})\ [3\ (\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] &&\\
\phi_6 &= (\eta_{20} - \eta_{02})\ [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] && (4.17)\\
&\quad +(4\ \eta_{11}\ \eta_{03} + \eta_{12})\ (\eta_{21} + \eta_{03}) &&\\
\phi_7 &= (3\ \eta_{21} - \eta_{03})\ (\eta_{30} + \eta_{12})\ [(\eta_{30} + \eta_{12})^2 - 3\ (\eta_{21} - \eta_{03})^2] && (4.18)\\
&\quad -(\eta_{30} + 3\ \eta_{12})\ (\eta_{21} + \eta_{03})\ [3\ (\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] &&
\end{aligned}
$$

In this system, only the first three of the seven moment invariants were used.

The angle of rotation between the major axis of the object and the x-axis of the coordinate system can be computed as follows:

$$
tan(2\ \varphi) = \frac{2\ \mu_{11}}{\mu_{20} - \mu_{02}} \tag{4.19}
$$

The remaining ambiguity as to the rotation of the main axis by 180 degrees is removed by adding the two constraints

$$
\mu_{20} > \mu_{02} \qquad \text{and} \qquad \mu_{30} > 0
$$

4.4 Classification

To chose a suitable classification technique for solving the problem one has to consider the requirements for the vision system as described in section 3.3.2. Given only one training sample per class, neural networks and statistical classification techniques have been discarded. Neither might the required training time for these types of classifiers satisfy the specified constraints.

The nearest neighbor rule, requiring no training time, was chosen for classification.

The implemented classifier operates in two phases:

1. Training phase: One training sample per class is stored, consisting of the known classID and the feature vector. This constitutes the *knowledgebase*, which is required during the classification phase.

2. Classification phase: Each object as described by its feature vector is assigned a classID based on the knowledgebase using the nearest neighbor rule.

The *nearest neighbor classification algorithm* is based on the idea, given a set of training samples, an unclassified individual should belong to the same class as the closest of its neighbors.

The Euclidean distance between two n-dimensional feature vectors v_i and v_j is computed as:

$$d_{ij} = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \tag{4.20}$$

where x_{ik} denotes the value of the k^{th} feature for the i^{th} training sample.

Each dimension of the feature space can have a different value range, which might cause misclassifications. The overall distance d_{ij} between two feature vectors can be biased by a distance $(x_{ik} - x_{jk})$ in a dimension k with a large value range, outweighing possibly more significant distances in the other dimensions.

One way to normalize the distance in each dimension is to divide each distance by the value range for that dimension. The value ranges can be estimated by statistically analyzing the training data.

A *Multiclassification* was implemented, allowing the classification rule to be applied to different feature sets in 'parallel'. A simple voting scheme was applied to combine the different classification results.

Chapter 5

Design of the vision system

5.1 Data flow

Data flow diagrams are used to describe the data flow of the implemented system. They are composed using the elements *data source*, *data object module* and *function module*.

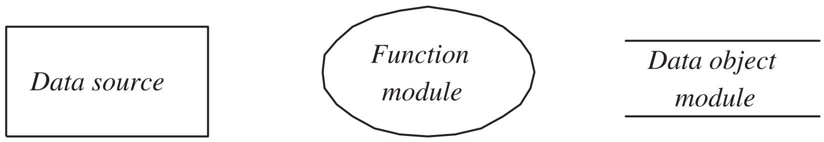

Figure 5.1: Elements of a data flow diagram

Physical sources like devices or files are represented as data sources. Data object modules manage data structures by storing and retrieving data elements. Data object modules don't actually modify the contained data, only function modules do. In general, data object modules provide separate functions to read from and write to the data sources. This is indicated by the dotted lines. Regular lines indicate the data flow between a data object module and a function module. The names next to the arrows corresponds to an entry in the data dictionary as described in 5.1.4. Function modules request data from data object modules, manipulate the data and then write the data back into a data object module.

The implemented vision system can operate in three different operation modes, namely

1. Adding a new pattern to the current knowledge base
2. Removing an existing pattern from the current knowledge base

3. Recognizing existing patterns (Classification)

5.1.1 Adding a new pattern to the current knowledge base

Given an image of a new colored mark, features for that mark are computed and stored in the current *knowledgebase*. The current knowledgebase contains one feature vector for each colored mark.

Figure 5.2 shows the data flow for adding a new pattern to the current knowledgebase. The function module *Image acquisition* obtains a color image either from the capture hardware or from a sequence of files. This color image is stored in the data object module *Composed image*. The function module *Image segmentation* processes the color image and stores in different data object modules

- the run length encoding (RLE) of the colored region,
- the boundary of the colored region,
- the highlighted characters and
- the highlighted text.

The data object module *highlighted text*, which stores the highlighted text in raw binary format, constitutes an interface for other programs to use the highlighted text. This can be used to obtain the highlighted text and feed it into an OCR program.

The user provides a label for the new colored mark, called a *pattern class ID*. The computed features, together with the pattern classID, constitute a *training sample*. This training sample is stored in the data object module *training samples*.

Based on all training samples in the current knowledgebase, new statistics for the current knowledgebase are computed and stored in the data object module *feature statistics*.

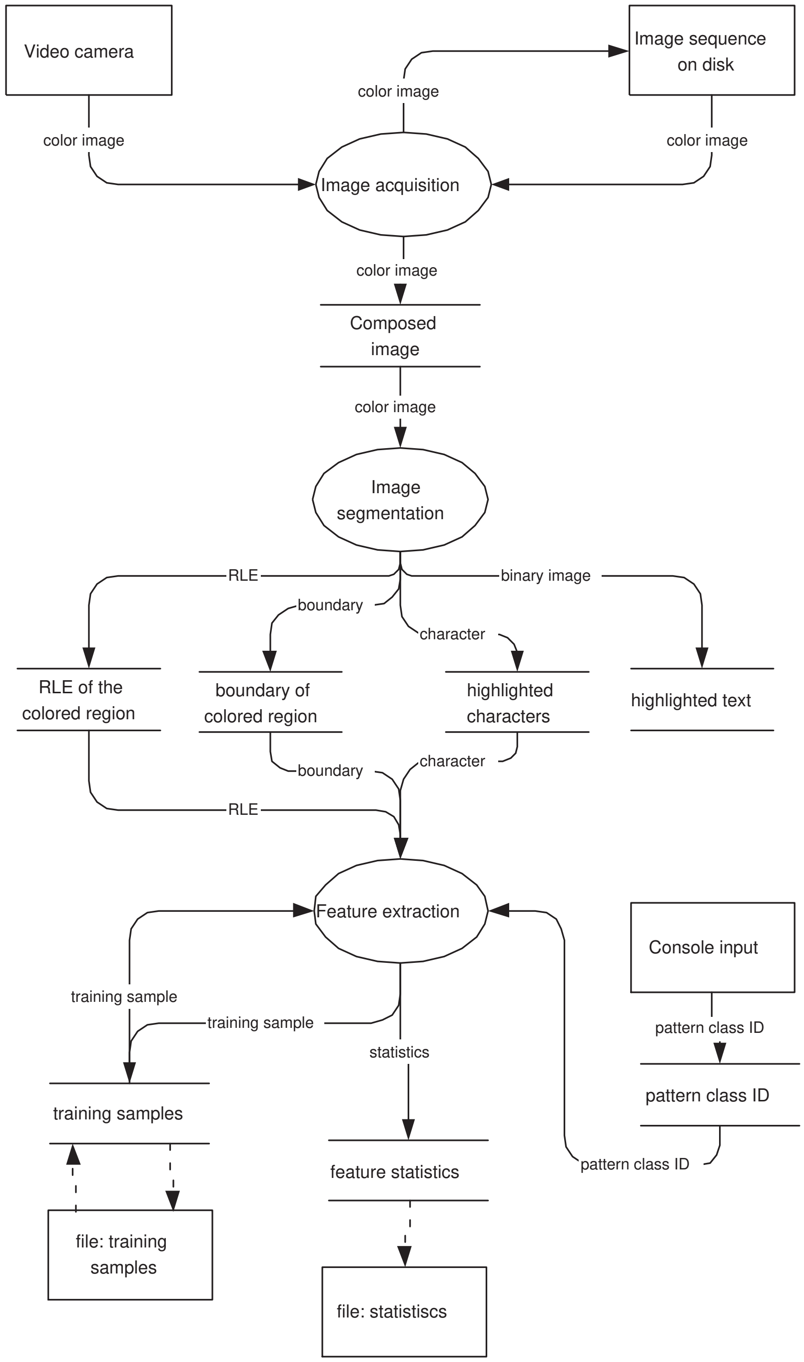

Figure 5.2: Data flow for adding a new pattern to the knowledgebase

5.1.2 Removing an existing pattern from the current knowledge base

The colored mark, specified by the user with its pattern class ID, is removed from the current knowledgebase (fig. 5.3).

The function module removes the specified training sample from the data object module *training samples* and computes new feature statistics for the reduced knowledgebase.

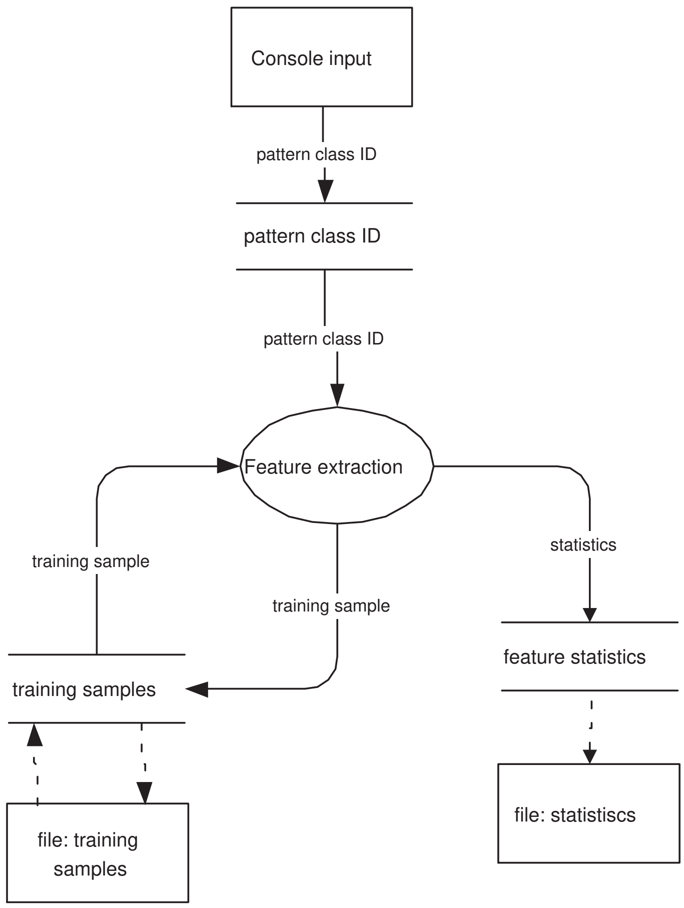

Figure 5.3: Data flow for deleting an existing pattern

5.1.3 Recognizing existing patterns (Classification)

In this operation mode, unknown patterns are compared to all the known patterns and a pattern class ID for the recognized pattern is generated (fig. 5.4).

The steps of image acquisition, image segmentation and feature extraction are exactly the same as during *adding a new pattern to the knowledgebase* as described in section 5.1.1. During classification, only the selected features are compared to the training samples stored in the data object module *training samples*. The nearest neighbor classification is applied, using all training samples and the feature statistics.

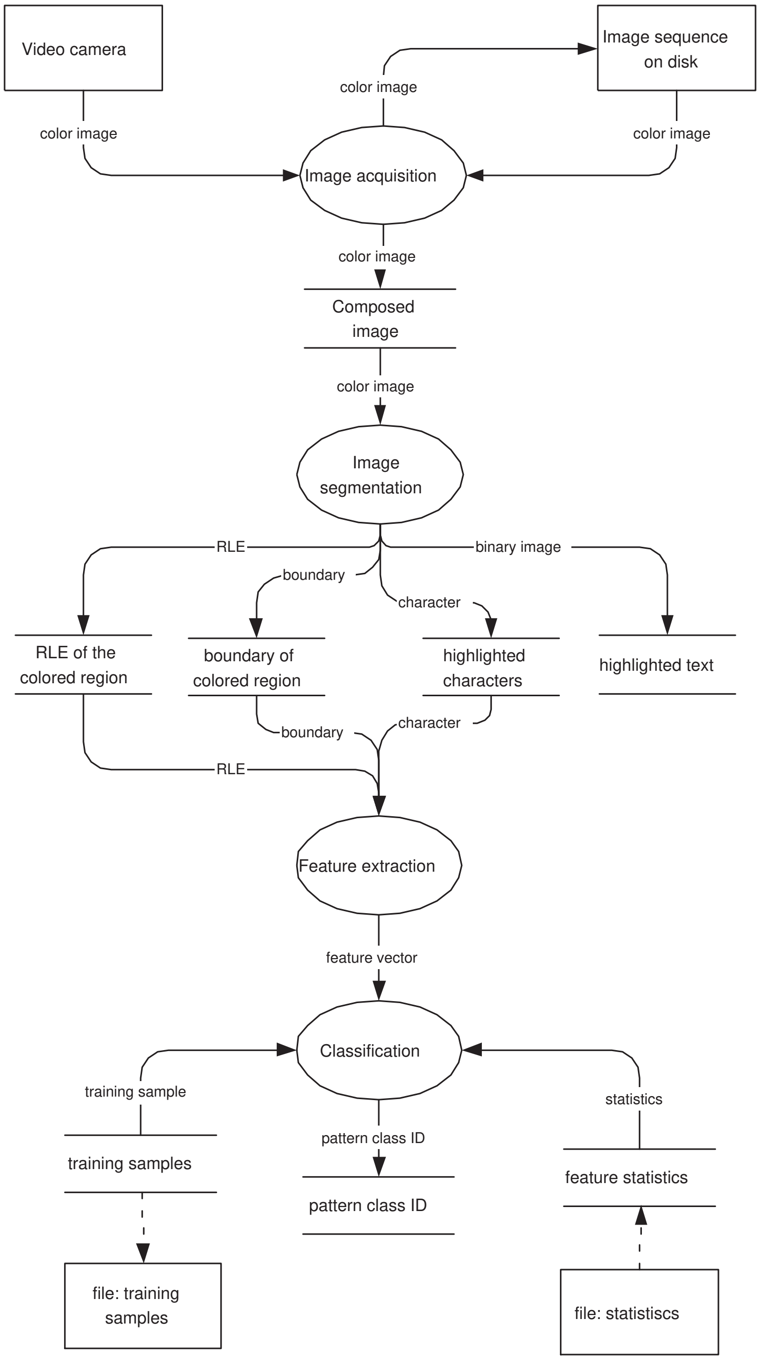

Figure 5.4: Data flow for recognizing existing patterns (Classification)

5.1.4 Data dictionary

1. **color image** = { *image width*, *image height*, *pixel value** }
   - **image width** = INTEGER
   - **image height** = INTEGER
   - **pixel value** = UNSIGNED LONG
2. **pattern class ID** = INTEGER
3. **RLE** = $\{start\ index,\ end\ index\}^*$
   - **start index** = INTEGER
   - **end index** = INTEGER
4. **boundary** = { *number of boundary points*, *point** }
   - **number of boundary points** = INTEGER
   - **point** = { x, y }
   - **x** = INTEGER
   - **y** = INTEGER
5. **binary image** = { *image width*, *image height*, *binary pixel value** }
   - **binary pixel value** = 0 | 1
6. **character** = { *image width*, *image height*, *binary pixel value** }
   - **bounding box** = { *upper left*, *lower right* }
   - **upper left** = *point*
   - **lower right** = *point*
7. **feature vector** = *feature**
   - **feature** = DOUBLE
8. **training sample** = { *pattern class ID*, *feature vector* }
9. **stats** = { *stats per feature over all classes*, *stats per feature per class* }
   - **stats entry** = { *minimum*, *average*, *maximum*, *variance*, *standard dev* }
     - **minimum** = DOUBLE
     - **average** = DOUBLE

  - **maximum** = DOUBLE
  - **variance** = DOUBLE
  - **standard dev** = DOUBLE
- **stats per feature over all classes** = $stats\ entry^{dimension\ of\ feature\ space}$
- **stats per feature per class** = $stats\ entry^{dimension\ of\ feature\ space*number\ of\ classes}$

Chapter 6

Summary and conclusions

In this thesis, a Computer Augmented Reality system for the integration of paper and electronic documents was presented.

Considering the limitations of the existing systems, a scenario based approach was utilized to understand and derive requirements for a system to support *paper-digital hyperlinks* from paper documents to related electronic information. Based on the derived requirements, a prototype system called *PaperLink* was designed and implemented.

In the implementation of the computer vision system, the outcome of the image segmentation was initially strongly influenced by noise. The analysis of the color values showed that the noise was due to an inherent problem of using a highlighting marker to color text on paper. The originally black pixels of the text partly changed their color due to the physical interaction with the highlighting marker. This problem was solved using a combination of the color image and the grayscale image of the colored mark.

The literature review indicated that systems have been proposed to provide a paper-digital hyperlink, but none of the systems supports a digital-paper hyperlink. This is the last missing hyperlink of the four possible ones between paper documents and electronic information.

The PaperLink system does not support static paper-digital hyperlinks. A combination of static paper-digital and personal paper-digital hyperlinks seems promising and needs further investigation.

Bibliography

[1] Arai, T.; Machii, K.; Kuzunuki, S.: "Objects on InteractiveDesk", in *the ACM Symposium on User Interface Software and Technology (UIST'95)*, pp. 37-38, Nov. 1995.

[2] Ballard, D.; Brown, C.: "Computer Vision", Prentice-Hall, 1982.

[3] Bush, Vannevar: "As We May Think", in the *Atlantic Monthly*, July 1945.

[4] Bryant, A.; Bryant, J.: "Recognizing shapes in planar binary images", *Pattern Recognition*, Vol. 22, No.2, pp. 155-164, 1989.

[5] Dai, M; Baylou, P.; Najim, M.: "An efficient algorithm for the computation of shape moments from run-length codes or chain codes", *Pattern Recognition*, 25(10), pp. 1119-1128, Feb. 1992.

[6] Dillon, A.: "Reading from paper versus screens: A critical review of the empirical literature", *Ergonimics*, vol. 35(10), pp. 1297-1326, 1992.

[7] Duda, R.; Hart, P.: "Pattern Classification and Scene Analysis", John Wiley and Sons, 1973.

[8] Cooperstock, J.: "Evolution of a Reactive Environment", *Computer Human Interaction (CHI)*, pp. 170-177, May 1995.

[9] Feiner, S.; MacIntyre, B.; Seligmann, D.: "Knowledge-Based Augmented Reality", In *Communications of the ACM*, Vol. 36, No.7, pp. 53-62, July 1993.

[10] Fitzmaurice, G.W.: "Situated Information Spaces and Spatially Aware Palmtops", In *Communications of the ACM*, Vol. 36, No.7, pp. 39-49, July 1993.

[11] Hansen, W.; Hass, C.:"Reading and writing with computers: A framework explaining differences in performance", *Communications of the ACM*, 31(9), pp. 1080-1089, 1988.

[12] Hu, M.: "Visual pattern recognition by moment invariants", *IRE Transactions on Information Theory*, IT-8: pp. 179-187, 1962.

[13] Foley, J.; van Dam, A.; Feiner, S.; Hughes, J.: "Computer Graphics, Principles and Practice", *Addison Wesley*, 1990.

[14] Gonzalez, R.; Woods, R.: "Digital Image Processing", *Addison-Wesley*, 1992.

[15] Jain, R; Kasturi, R; Schunck, B: "Machine Vision", *McGraw-Hill*, 1995.

[16] Krueger, M.W.: "Artificial Reality II", *Addison-Wesley*, 1991.

[17] Long, S.; Aust, D.; Abowd, G.; Atkeson, C.: "Cyberguide: Prototyping Context-Aware Mobile Applications", *Computer Human Interaction (CHI'96)*, April 1996.

[18] Mackay, W.E. et al.: "Ariel: Augmenting Paper Engineering Drawings",(video), *Computer Human Interaction (CHI)*, May 1995.

[19] Mackay, W.E.; Velay, G.; Carter, K.; Ma, C.; Pagani, Daniele: "Augmenting Reality: Adding Computational Dimensions to Paper", in *Communications of the ACM*, Vol. 36, No.7, pp. 96-97, July 1993.

[20] Rao, R.; Card, S.; Johnson, W.; Klotz, L.; Trigg, R.: "Protofoil: Storing and Finding the Information Worker's Paper Documents in an Electronic File Cabinet", *Proceedings of CHI'94*, pp. 180-185, 1994.

[21] Rekimoto, J.; Nagao, K.: "The World Through the Computer: Computer Augmented Interaction with Real World Environments", *UIST*, pp. 29-36, Nov. 1995.

[22] Seybold, A.: "The DOCIT", *Andrew Seybold's Outlook on Professional Computing*, 11(2), 1992.

[23] Weiser, M.: "The computer for the twenty-first century", *Scientific American*, pp. 94-104, Sept. 1992.

[24] Wellner, P.; Mackay, W.; Gold, R. (Editors): "Computer Augmented Environments: Back to the Real World", In *Communications of the ACM*, Vol. 36, No.7, pp. 24-26, July 1993.

[25] Newman, W.; Wellner, P.: "A desk supporting computer-based interaction with paper documents", in *Proceedings of CHI'92, and Technology*, pp. 587-592, Nov. 1992.