Video-Augmented Environments

A dissertation submitted for the degree of Doctor of Philosophy

James Quentin Stafford-Fraser

Gonville & Caius College
University of Cambridge

February 1996

To my parents, who taught me to explore,
and to my wonderful wife Rose,
who encouraged & supported me while I did so.

Preface

Except where otherwise stated, this dissertation is the result of my own work and is not the outcome of work done in collaboration.

This dissertation is not substantially the same as any that I have submitted for a degree or diploma or other qualification at any other University.

No part of this dissertation has already been or is concurrently being submitted for any such degree, diploma or other qualification.

This dissertation is copyright
© 1996 by J.Q.Stafford-Fraser

Abstract

In the future, the computer will be thought of more as an assistant than as a tool, and users will increasingly expect machines to make decisions on their behalf. As with a human assistant, a machine's ability to make informed choices will often depend on the extent of its knowledge of activities in the world around it. Equipping personal computers with a large number of sensors for monitoring their environment is, however, expensive and inconvenient, and a preferable solution would involve a small number of input devices with a broad scope of application. Video cameras are ideally suited to many real-world monitoring applications for this reason. In addition, recent reductions in the manufacturing costs of simple cameras will soon make their widespread deployment in the home and office economically viable. The use of video as an input device also allows the creation of new types of user-interface, more suitable in some circumstances than those afforded by the conventional keyboard and mouse.

This thesis examines some examples of these 'Video-Augmented Environments' and related work, and then describes two applications in detail. The first, a 'software cameraman', uses the analysis of one video stream to control the display of another. The second, '*BrightBoard*', allows a user to control a computer by making marks on a conventional whiteboard, thus 'augmenting' the board with many of the facilities common to electronic documents, including the ability to fax, save, print and email the image of the board. The techniques which were found to be useful in the construction of these applications are common to many systems which monitor real-world video, and so they were combined in a toolkit called '*Vicar*'. This provides an architecture for 'video plumbing', which allows standard video-processing components to be connected together under the control of a scripting language. It is a single application which can be programmed to create a variety of simple Video-Augmented Environments, such as those described above, without the need for any recompilation, and so should simplify the construction of such applications in the future. Finally, opportunities for further exploration on this theme are discussed.

Acknowledgements

I am very grateful to Rank Xerox Research Centre (EuroPARC) for their support of this work, both in terms of finance and encouragement. Particular thanks go to Mik Lamming, Pierre Wellner (now at AT&T) and Mike Molloy for their enthusiasm, inspiration and patience, and to Michael Taylor for helpful & challenging discussions and for proof-reading.

Peter Robinson deserves a great deal of thanks; he provided the right amounts of assistance and advice at exactly the right times – I could not have wished for a better supervisor. I would like to thank the University of Cambridge Computer Laboratory, in particular the members of the Rainbow graphics group, Gonville & Caius College, and the many colleagues and friends too numerous to be mentioned individually, who have contributed ideas, time, assistance or just caffeine. I am grateful to you all.

The names of all products referred to in this thesis are acknowledged as the trademarks of their respective owners.

Table of Contents

Video-Augmented Environments **1**
- Introduction 1
- Achieving Augmented Environments 3
- Terminology 4
- Thesis Organisation and Scope of Research 5

Example VAEs and related work **7**
- Introduction 7
- BrightBoard I 7
- In/Out Board 11
- The Virtual Window system 11
- Hand tracking and gesture recognition 12
- DigitalDesk 13
- The ALIVE system 13
- Software Cameraman & BrightBoard 14

VAE techniques **15**
- Introduction 15
- Subsampling 15
- Motion Detection & Background Separation 16
- Thresholding 23
- Resolution Enhancement and Greyscale Thresholding 25

The Software Cameraman **30**
- Introduction - The motive 30
- Doing it with software 31
- What constitutes interest? 32
- Selecting the area 32
- Selecting the eyepiece view 32
- Aesthetic Aspects of Camera Control 33
- Future Possibilities 35

BrightBoard **36**
- Introduction – The whiteboard as a user interface 36
- A Video-Augmented Drawing Surface 37
- BrightBoard In Use 39
- How does BrightBoard work? 41
- Triggering 41
- Preprocessing 42
- Feature detection 42
- Analysing 49
- Executing 50
- Evaluation 51
- Future possibilities 52

Vicar: A VAE construction kit **54**
- Introduction 54
- Vicar 55
- Scripting 55

The Architecture 57
Flow Control 58
Frame transport in Vicar 59
The sequence of source-sink interactions 60
Vicar in use: The Software Cameraman and other examples 64
In/Out Board monitor 66
A User Interface for Vicar 67
Future Possibilities 68
Related Work 71

Conclusions and Future Possibilities 73
Future Possibilities 73
Summary & Conclusion 77

The BrightBoard Program Suite 79

The features used for BrightBoard's symbol recogniser 81

The TCL source code for a simplified automatic cameraman 83

References 85

Chapter One

Video-Augmented Environments

Video cameras can now be produced, with controlling circuitry, on a single chip. Digital output reduces the need for expensive frame-capture cards, and it is reasonable to assume that simple cameras, being devoid of moving parts, will soon be a cheaper accessory for personal computers than keyboards, microphones, and mice. The possibilities for video-conferencing and other human-human interactions are obvious, but how might the ready availability of video sources enrich the way we interact with the machines themselves?

Introduction

In the office environment the computer has traditionally been thought of as a tool, in the same way that a typewriter or Rolodex is a tool. It is an item employed by a person to accomplish a particular task. It is a very flexible tool, and can fulfil the roles of typewriter, Rolodex and calculator simultaneously, but the activity is generally initiated by a 'user', a term which succinctly describes the relationship.

We might hope that the computer of the future would be more like an assistant than like a typewriter, that it would perform mundane tasks *on your behalf* when it sees they are necessary rather than always operating more directly *under your control*. Most computers spend the vast majority of their time idly waiting for humans, when they could be seeking out new business opportunities, trawling the world's libraries for information relevant to a particular research project, or simply answering the phone and throwing away junk mail. It is this vision which has fuelled the recent research into autonomous agent-based software.

The extent to which users will be happy for a computer to make decisions on their behalf will depend on their confidence that it will make the *right* decisions. The more a secretary knows about your way of life, your preferred modes of work, how cluttered your desk is and when you are in a meeting, the more useful he or she can be to you.

The same applies to a machine, but the average personal computer has no knowledge of the world outside its box and has to be spoon-fed with information through its limited range of input devices.

One approach to providing richer human-computer interaction is that of Virtual Reality (VR). Users of VR don head-mounted displays, earphones, gloves and microphones so that as many senses as possible can be used to draw them into an artificial, computer-generated world. Large amounts of money and effort are expended to create systems with higher resolution, better frame-rates, and tactile feedback so that this artificial world can resemble the real one as closely as possible. While VR certainly has valuable applications in specialist areas, the users still have to live, work, eat and sleep in the real world, and it could be argued that our lives will be richer if we can take the opposite approach: if we can make the computer an integral part of our world rather than making ourselves a part of the computer's. Our world, after all, has infinite resolution, completely smooth motion and better tactile feedback than any artificial glove is able to give.

This is the philosophy behind *Computer-Augmented Environments*; the desire to bring the computer 'out of its box' and give it more awareness of the world around, so that it *augments* and *enhances* daily life rather than attempting to replace it. Can we, for example, enable the computer to understand what we do in our offices rather than putting an impoverished 'office metaphor' on the machine's screen? Can we centre computational power around everyday objects with which users are so familiar that they don't think of them as a human-computer interface? During work on the DigitalDesk [56], for example, (see Chapter Two), we experimented with ways of enabling the computer to recognise an ordinary pencil eraser, and using that as the means of deleting parts of an electronic image which was projected onto the desk. The motivation was simple: people already know how to use erasers, and will continue to use them for hand-drawn pictures. By augmenting the eraser's capabilities we simply expand its scope of use. We neither detract from its abilities to erase pencil, nor require the user to learn a new tool.

It would be naïve to pretend that the wide range of tasks we accomplish with our current user interfaces could all be performed equally well by similarly 'projecting' them into our physical world. Nonetheless, by building on the established abilities and expectations of the user we can gain in several ways:

We can shorten the learning process. A user of the DigitalDesk might only need to be told, "You can use the eraser on projected drawings too"; a considerably shorter tuition than that required for the 'delete' process in many conventional software drawing packages.

We can make user-interfaces more 'physical'. Our bodies are designed to move in and interact with the physical world. When we limit their activity to sitting in a single chair, staring at a screen and

moving a mouse, they tend to react with aching backs, tired eyes and Repetitive Strain Injuries. To provide some variety for our bodies, we deliberately choose to perform tasks in the physical world which could have been performed on screen. We print even short documents before reading them, we carry software around the office on floppy disks rather than distributing it on the network, and we look up and dial telephone numbers ourselves when the computer could do it more efficiently. There is something very satisfying about crumpling a sheet of paper and hurling it into the waste-paper basket which has no equivalent in the world of email. A user's emotions can only be expressed by hitting the 'Delete' key a bit harder.

We can save the user from unnecessary duplication of tasks in the electronic and physical worlds. At Rank Xerox EuroPARC (where much of this work has been conducted) a convention has arisen that an open office door indicates the occupant's willingness to be disturbed, and a closed one should be seen as a deterrent. Whether consciously or otherwise, people will indicate intermediate levels of accessibility in the same way, for example by pushing the door almost completely, but not fully, closed. Experiments have been conducted [23] where machines, as well as colleagues, can take hints from such cues as these. The telephone bell is silenced after the first ring, for example, if the computer believes that a meeting is taking place. Incoming video calls can be rejected if the user is on the telephone. Assistance of this kind saves the user from tedious tasks, and is a step towards the concept of the computer as a 'secretary'.

Achieving Augmented Environments

There is an inherent difficulty in the goals of these Computer-Augmented Environments (CAEs). For computers to gain much understanding of the world around, they must be equipped with sensors to monitor it. These may be directly connected to the computer, or they may relay information to it via a communication medium such as radio or infra-red. The more sensors a computer has, the more complete will be the picture it can create of its surroundings and the activities of its users. However, the more sensors we embed in the environment, with their associated cables, batteries and power supplies, the more investment is needed to bring about our goal and the more intrusive the technology becomes, so frustrating the aim of an 'invisible' or 'transparent' user interface.

The solution is to give the computer a small number of senses which have a broad scope of application and which operate remotely, i.e. without direct contact with the objects being sensed. The obvious candidate senses are vision and hearing, through the use of video cameras and microphones. Similarly, if the computer is to communicate with humans, then the analogous output systems are audio feedback and perhaps some projection facilities. Not only do cameras and microphones have a broad range of application, they can often be applied to more than one task at the same time. A single camera might monitor a whole room, and detect when the room is

occupied, when a meeting is in progress, and when an overhead projector is being used, as well as recording the meeting. Out of office hours it could also function as a security camera.

This thesis explores the use of video in the creation of Computer-Augmented Environments. Until recently the deployment of such systems in the average workplace would be limited by the cost both of cameras and of the computing power required to process video signals. However, manufacturing developments are making video cameras an economically viable alternative to more conventional sensors, and the typical office workstation is now capable of the simple image processing required for many of these 'Video-Augmented Environments'.

Terminology

It is worth taking a moment to clarify some of the terminology used in this area. The phrases *ubiquitous computing* and *augmented reality* have been used to describe approaches similar to the *computer augmented environments* discussed here, but with minor differences.

Ubiquitous Computing

Mark Weiser used the phrase 'Ubiquitous Computing' or 'Ubicomp' to denote his vision of a world where computers of various sizes are liberally scattered around the home and workplace. You jot down ideas on them as you would on a notebook, you sketch diagrams on them as you would on a whiteboard, and you wear them as you would wear a watch. These computers can talk to each other and to the stereo, the fax machine, the air-conditioning. Their ubiquity means that we think no more of them than we do of a stapler or a post-it note, and the communication abilities mean that we can take a computer to the task instead of having to bring the work to the computer [52].

While this model certainly depicts a closer integration of the real and electronic worlds, it relies on the assumption that the artefacts around us will increasingly be equipped with some computational power. This is certainly reasonable with respect to telephones and washing machines, but it may be some time before coffee cups, magazines and waste-paper baskets are equipped with batteries and infra-red communication links. The Ubicomp philosophy is not incompatible with VAEs, but the focus of VAEs is to add computational power to the devices we already use, rather than to create and exploit new varieties of computer.

Augmented Reality

The name 'Augmented Reality' has been used for systems which make use of see-through displays, especially head-mounted systems, to 'project' extra information onto the user's view of the world. Feiner *et al*, for example, used a head-mounted 'Private Eye' display to overlay on the user's view of a laser printer the information required to service

it [12]. The name emphasises the contrast with 'Virtual Reality', and could sensibly be applied to a wider field, but has become so closely associated with this particular method of interaction that it was thought best to avoid it in the present work.

Computer Augmented Environments

'Computer Augmented Environments' was the phrase chosen by the editors of the July 1993 *Communications of the ACM* as a generic heading under which to group the above and similar projects which "are united in a common philosophy: the primacy of the physical world and the construction of appropriate tools that enhance our daily activities" [55]. The present work deals with the subset of such tools which use video to monitor the physical world, and so the name *Video Augmented Environments* (VAEs) was chosen.

Thesis Organisation and Scope of Research

Organisation

The following chapter will describe some example Video-Augmented Environments and provide an overview of some related work. In Chapter Three there is a discussion of some of the problems commonly faced by VAEs and techniques for tackling them. Chapter Four describes a software-based 'automatic cameraman' which uses activity in one video stream to control the processing of another. Some of the lessons learned in the creation of the cameraman were important in the creation of BrightBoard, a system which uses video to augment the facilities of an ordinary whiteboard. BrightBoard is the focus of the thesis and is described in Chapter Five. We discovered that the construction of many such VAEs consisted chiefly in the plugging together of standard image-processing units, so in Chapter Six we discuss VICAR, an architecture for 'video plumbing' which allows simple VAEs to be created under the control of a scripting language. Finally, Chapter Seven draws some conclusions and describes possibilities for future developments.

Scope

The research described concentrates on practical applications of video to real-world problems using readily available hardware. In doing so it touches on a wide variety of topics within and on the borders of Computer Science, including image processing, pattern recognition and OCR, logic programming, human-computer interaction and document image understanding. A conscious decision was made at an early stage to explore a wide space of applications and not to be sidetracked into the minutiae, fascinating though many of these are. To evaluate some of these applications on today's standard hardware, speed considerations have been paramount. Often a sophisticated algorithm has been rejected because a simpler one has been found to suffice, or because the performance requirements of interactive

applications have forced us to sacrifice the elegant in favour of the practical.

Goals

In short, this work aims:

- To explain why video is important and useful as a generic input device.
- To examine some examples of Video-Augmented Environments and note techniques which were found to be useful in their construction.
- Using these findings, to create a toolkit to simplify the building of similar applications in the future.

Chapter Two

Example VAEs and related work

Introduction

In this chapter we discuss some examples of recent research which illustrate the application of relatively simple image processing to real-world video.

BrightBoard I

BrightBoard explores the use of a whiteboard as a computer interface. Chapter 4 will describe the motivation for such a system and the final implementation, but here we will introduce the concepts by describing an early prototype, *BrightBoard I*.

Background

BrightBoard I had *its* origins in a simple whiteboard-recording program. A camera was suspended from the ceiling or mounted on a high bookshelf to give it a clear view of the whiteboard. The program would save an image of the board whenever it detected a 'significant change' (a concept not precisely defined at that stage). The current view was displayed on the screen, and a scrollbar allowed a user to scan earlier images and so to review recent activity on the board, recover previously erased information, and so forth.

Experimenting with the system, users would frequently request extra features such as the ability to print images, email them, or pass them to another program for processing. Rather than expanding the system into a single monolithic application, the decision was made to simplify it into a whiteboard-based 'control panel' which could be extended through the use of shell scripts and external programs, and which would chiefly be operated from the board itself. The program was rewritten from scratch and named BrightBoard.

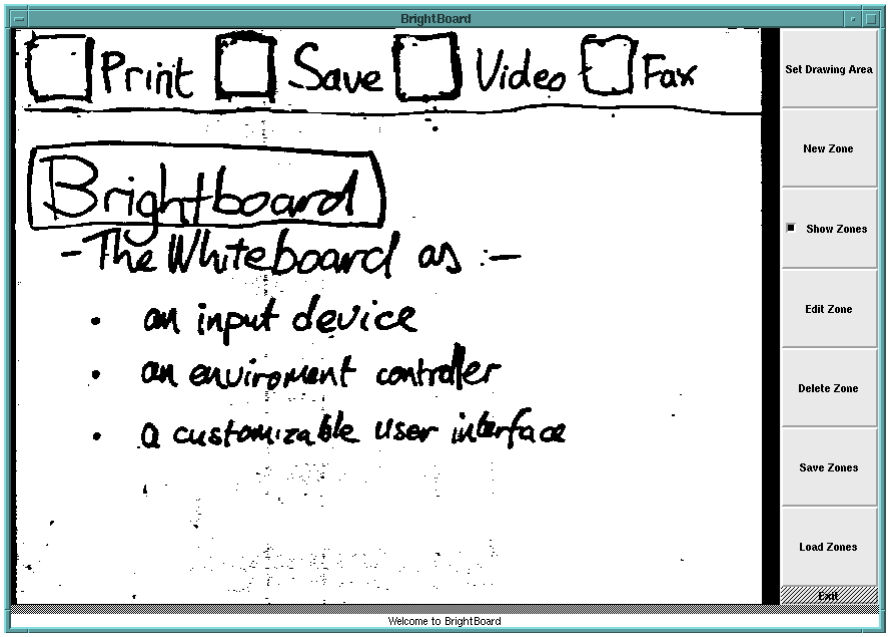


Figure 1. The user is presented with an image of the board

BrightBoard I - The user's view

The user, on starting the program, is presented with a thresholded image of the board which is updated every few seconds. He uses the mouse to select an area of the image which will be a 'sensitive zone', typically corresponding to a checkbox or other region marked on the board. Commands can then be specified which will be executed when changes occur within the zone (Figure 1 to Figure 3) The system detects when the zone becomes significantly darker (corresponding to a mark being made on the board) or lighter (when a mark is erased).

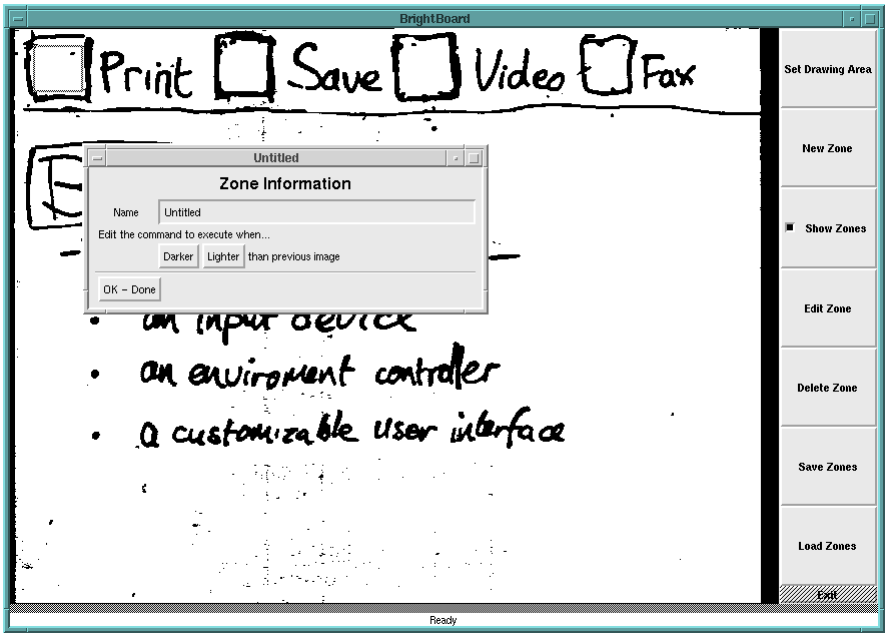


Figure 2. The user selects a sensitive zone and is prompted for information about it

Each may trigger a separate action, so checking a box might switch on the video recorder at the start of a meeting, and erasing the mark would stop the tape during confidential discussions or at the meeting's end. The action can consist of any Unix shell command, and the user may request that all or part of the image be passed to the command via a temporary file.

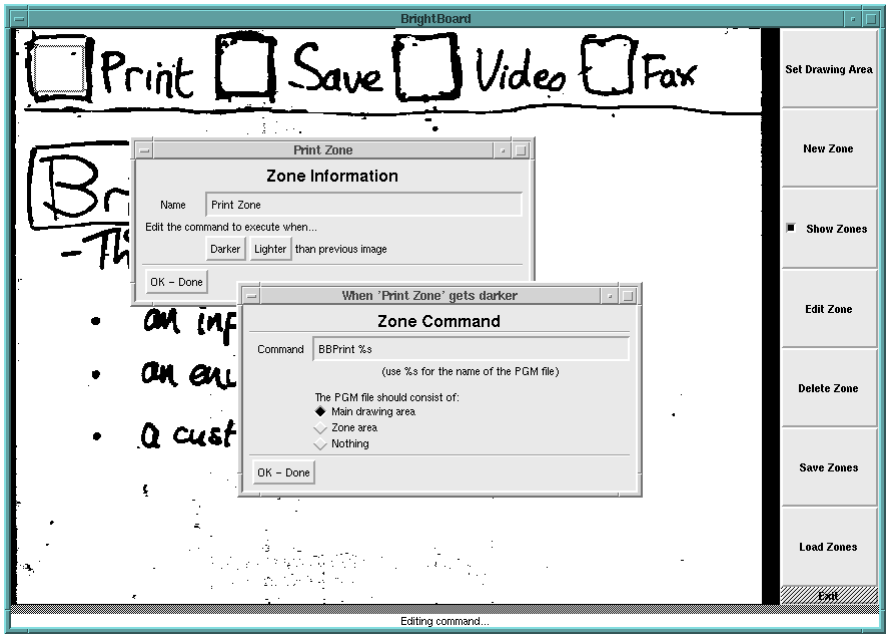


Figure 3. The user enters details of commands to be executed

Some sample commands

A simple whiteboard recorder might be created by selecting the whole area of the board as a sensitive zone, and specifying the command for both `lighter' and `darker' as:

```
cp %s ~/board/
```

(The %s will be replaced automatically by the name of a temporary file containing the image). A more sophisticated version could be in a script called BBSave, triggered by a `Save' zone:

```
#! /bin/sh
echo STAMP=‘date +%y%m%d%H%M%S‘
cp $1 ~/board/$STAMP
cat saved.au >/dev/audio
```

This saves the image using a filename derived from the current date and time, then uses audio (recorded speech) to announce that it has done so. The command executed by BrightBoard would then be:

```
BBSave %s
```

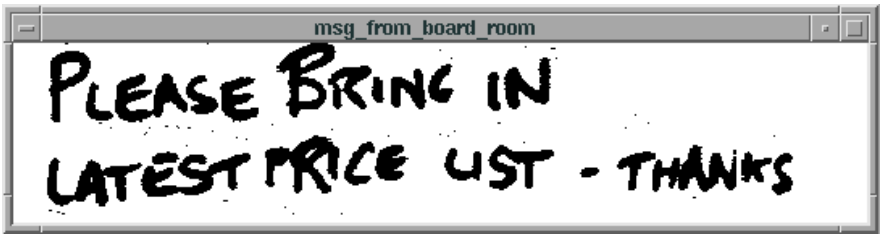
A `Print' button might be implemented as:

```
pgmtops %s | lpr
```

These ‘Save’ and ‘Print’ examples would generally act on the whole of the drawing area. In contrast, the following 'message zone' example

uses just the area within the zone. A larger area is selected as the sensitive zone, and anything written there pops up on a secretary's screen, using the image-viewing command 'xv':

```
xv -display barbara:0 %s
```



BrightBoard I's zone configuration data can be saved to disk and restored at a later date. The format of the file is such that it can easily be edited or generated by users or by other software; perhaps by a program which automatically detected the checkboxes.

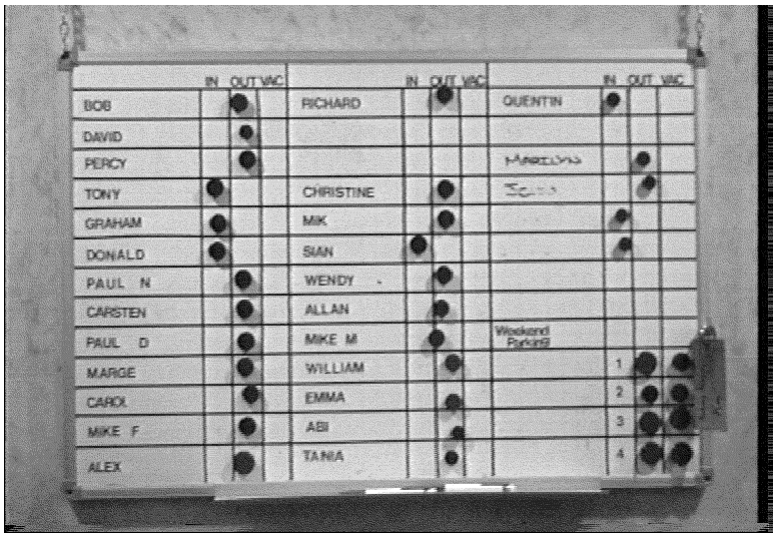
Discussion

The biggest challenge for *BrightBoard I* was distinguishing a mark made on the board from the hand or head of a user. It used a very crude algorithm which detected the persistence of any change: if the zone becomes darker and remains so for several seconds then it is probably not a writing hand or the head of a passer-by. This is obviously not foolproof, and is achieved at the expense of response time; nonetheless, it made for quite a usable prototype with a ceiling-mounted camera which rarely suffered from obstructions in its view. We will examine a more sophisticated 'person detector' in Chapter 5.

The system also relied on the fact that the sensitive zones were in a fixed place in the camera's image. Should the camera, board or checkboxes be moved, the system had to be reconfigured.

Despite these limitations, a surprising amount can be done with a system which simply counts the number of changed pixels, and *BrightBoard I* was used for other prototype systems such as the In/Out board described below.

In/Out Board



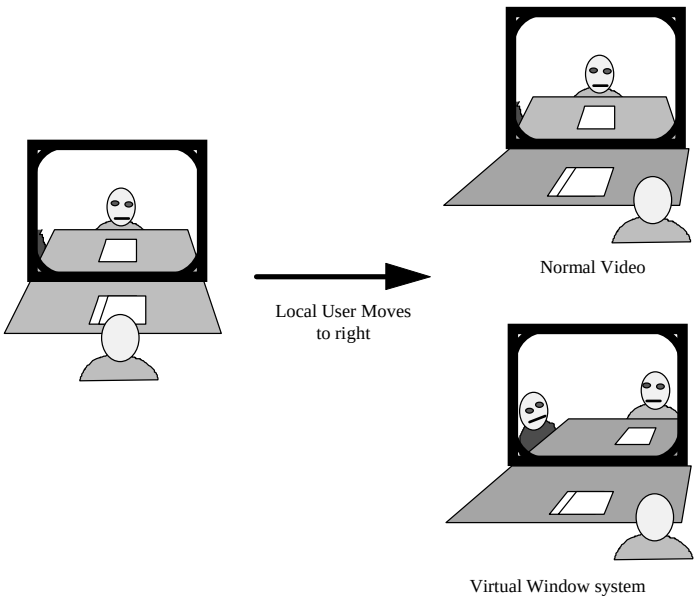
At EuroPARC, there is a board in the entrance lobby on which members of the lab place magnetic markers indicating whether they are in, out, or away on vacation. It is a very natural interface with which users feel comfortable and it requires almost no effort as one passes by on the way in and out of the building. People probably would not voluntarily use a system which involved a screen and keyboard, so this is a prime example of the advantages of 'augmenting' an existing physical user interface.

Instrumenting this system could be very useful - when somebody arrives the computer could notify others who were waiting for them, or switch on their office lights and computer screens. Writing a video-based system to do this is a simple process – *BrightBoard I* has been used to demonstrate the idea, with no changes to the software being required. In this example system, users were sent email each day detailing their arrival and departure times and the number of hours worked. Adding computational features to such a mechanical, tactile system any other way, perhaps by replacing the magnetic markers with push-switches, would be considerably more complex and costly.

The Virtual Window system

A simple video-analysis system has been used by Gaver *et al* to enhance the capabilities of a video-conferencing link [16].

A user's head movements in a local office are monitored by a cheap dedicated camera and control the movement of the video-conferencing camera in a remote office. The metaphor used is that of a window onto the remote office, where by moving his head to the right, for example, a user can see 'around the corner' to the left.



This provides a greater effective field of view and resolution, some semblance of 3-D, and allows for smoother interaction between users. The authors describe a simple example:

> "…it is common to hold something up to show a remote colleague, only to misjudge and hold it partially off-camera. Correcting the error usually requires explicit negotiation ("a little to the left…no, *my* left!"). The Virtual Window system allows the remote viewer to compensate for his or her partner's mistake simply by moving, without requiring any explicit discussion about the mechanics of the situation."

Chapters 3 & 4 will discuss in more depth the topic of tracking human movement.

Hand tracking and gesture recognition

A variety of systems can be controlled efficiently by monitoring the position and gestures of a user's hands. The *Charade* system, for example, used hand gestures to navigate through a hypertext system on a projected display [2]. Such systems have traditionally required the user to wear special gloves and position-sensing devices, but Jakub Segen of AT&T Bell Labs describes a system using video alone which allows, for example, the user to control a mouse-style pointer on the screen, or to 'fly' through a Virtual Reality-like environment, using hand movements and gestures watched by a camera [45].

DigitalDesk

Figure 4. The DigitalDesk

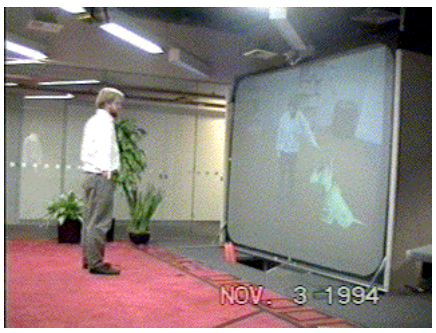
The DigitalDesk [53,56] is a normal desk on which users can lay out their papers and leave their coffee cups, but it also has some characteristics of a workstation. Mounted above the desk are one or more cameras and a projector, which enable a computer to read paper documents placed on the desk, to monitor a user's activity there, and to project images, documents and annotations down onto the desk. One of the interesting problems tackled by the DigitalDesk is that of calibrating the multiple input and output devices so that a mapping may be made between their individual coordinate systems.

A major goal of the project was the blurring of the boundary between paper and electronic documents. A number of prototype applications have been built which allow, for example:

- copying and pasting of images and text from paper documents into electronic ones (the 'PaperPaint' application)
- mathematical operations on numeric data contained in paper documents ('DigitalDesk Calculator')
- language translation: pointing at words in a French document will cause an English translation to be projected alongside ('Marcel').

The DigitalDesk is one of the most fully developed examples of Augmented Reality, and the reader is referred to the References for a more detailed description.

The ALIVE system



ALIVE [34,35] stands for Artificial Life Interactive Video Environment and is the name of a system, created at the MIT Media Lab, which develops some of the ideas behind Myron Krueger's *VideoPlace* [29]. A camera is fixed to the top of a large projection screen, and monitors a 16ft x 16ft area in which a user is free to move about. The image of the user is separated from the background and incorporated in another scene (typically a view of a room) which is then projected onto the screen in front of the

user. Also included in the image are computer-generated animated creatures which move around the image under the control of autonomous agents. Their activity is affected by their interactions with one another and by the movement and gestures of the user. The result is a 'magic mirror' in which the user sees himself in a virtual world where he can, for example, bend down to pet an artificial hamster which then obligingly rolls over onto its back to be tickled. Other creatures can be sent away by pointing to the far side of the room, where they sulk until the user moves back towards them again.

This system is great fun to use, but the technology required to make it work, in terms of both hardware and software, is substantial and rather specialised, and as such it is really beyond the scope of applications considered here. In addition, it is not so much a 'Video Augmented Environment' as an example of 'Environment-Augmented Video'! However, ALIVE is of particular interest here because of the richness of interaction afforded by the use of a single video camera as an input device, and because it involves several simple techniques which can be useful in other systems. These include the following:

- The user's position in the 3D space is known, despite only having a single video source. The camera looks down on the user at an angle, and because the system knows the relative positions of camera and floor, and the fact that the user's feet must be *on* the floor, it can deduce the users proximity simply by finding the lowest point in the outline of the user. This greatly increases the realism, because the user's image then occludes, or is occluded by, parts of the virtual world depending on his/her position.
- Simple gestures are detected by tracing the user's outline in low resolution and finding the points of maximum convex curvature, which typically correspond to hands, head and feet.
- Because the agents are modelled as animals, users are much more tolerant when they fail to understand or act on a command immediately – a common problem in VAEs!

Software Cameraman & BrightBoard

In Chapters 4 & 5 we will discuss in more detail two further VAE examples, but first let us examine some techniques which are useful when constructing such tools.

Chapter Three

VAE techniques

Introduction

For most applications video has a very low signal-to-noise ratio when compared to other sensors; that is, the information in which we are interested forms only a small part of the information captured by the camera. The task of the application is often to reduce several megabytes per second of video data to a simple analysis like "there is/isn't a person in the room". In this chapter we discuss a few problems which are common to VAEs, and suggest some techniques for dealing with them. The issues are particularly relevant to BrightBoard, but they all have a much wider potential scope of application.

Subsampling

The first 'technique' is so simple that it scarcely deserves the name, but it is sufficiently important to be worth mentioning first. Real-world video images contain a substantial amount of data, and if they are to be examined in any but the simplest of ways, a reduction in the resolution of the image often allows the single largest speed gain. Stephen Smith [48] points out that a second of digital video typically contains more information than the complete works of Shakespeare.

An application will often have no need of a high-resolution image at all. In other cases, certain parts of the analysis can profitably be done at low resolution. Later chapters will describe how BrightBoard, for example, uses a 40x30 image to detect whether users are obstructing the view of the whiteboard before capturing a 740x570 image for detailed analysis. The number of pixels to be analysed in the 'idle' monitoring state is reduced from 420,000 to 1200, allowing a great improvement in response time, and/or a reduction in workstation load.

Motion Detection & Background Separation

VAE applications, in general, are interested in movement and change within a video image. Sometimes we wish to analyse the movement itself, as in the software cameraman described later, sometimes we just wish to know that movement has occurred because it is a trigger for something else, and sometimes we wish to avoid situations where change is occurring e.g. if we wish to capture whiteboard images without any humans in the way.

Often it is human movement which is of interest – people in an office, hands on a keyboard, crowds on a station platform – but other things we might wish to monitor include pages emerging from a printer, documents moving on a desktop, or cars entering and leaving a parking lot. The common factor in all these situations is that movement occurs against a relatively stable background. 'Background separation' techniques aim to detect components of the image which are not part of a stable 'background' and so may be of interest.

Movement, colour and intensity

The majority of visual occurrences which attract our attention are due to changes in luminance; variations in the brightness of particular areas of the visual field [18]. These may be:

- direct changes in intensity due to a light source variation, for example from a flashing warning light, or the flickering of a flame.
- intensity changes caused by movement.

The algorithms described here assume that the images captured are greyscale and not colour. The human visual system is more sensitive to changes of intensity than to changes of colour and the movement of objects is seen primarily by the difference in intensity between them and their background, rather than by chromatic variations [33,36,37]. In addition, monochrome cameras currently offer better image clarity for a given price than their colour counterparts, and so are perhaps more likely to be used in VAEs. We will therefore concentrate on monochrome, but it is worth noting that much important colour-based work has also been done. Ueda et al, for example, have impressive demonstrations of video segmentation based on chromatic analysis of real-world video [51].

Not all motion can be detected by changes in intensity. Horn & Schunck [25] point out that a rotating sphere with no surface markings causes no temporal changes in image intensity, despite the presence of real-world motion. Related to this is the 'aperture effect': if an unmarked object is viewed through a smaller aperture (e.g. a pencil seen close-up but through a key-hole) then only the component of motion perpendicular to an edge can be detected. Such situations rarely occur in the real-world applications considered here, however, and it should be noted that the human visual system could not detect motion under such conditions either.

Differencing

The simplest way to detect movement is to compare each captured video frame with its predecessor. At each pixel position, the magnitude of the change in pixel values is calculated, and any movement will generally cause significant values to be recorded. See Figure 7 (C). There will always be minor pixel differences between two live images even when the scenes appear identical to the naked eye. These are the result of such things as:

- small variations in daylight illumination
- the slight flicker of electric lighting
- vibration induced e.g. by the fans of nearby equipment
- the electronic limitations of the camera and frame grabber
- electrical noise induced in the video circuitry.

Many of the above can be eliminated by ignoring changes below a certain threshold, and unconnected single pixel changes.

Differencing is simple, fast, and often supported in hardware. The chief limitations are as follows:

- If the frame rate is high relative to the speed of any movement, there may be insufficient change between frames to allow much analysis.
- If the frame rate is low, then the position of fast-moving objects will be sufficiently altered between frames for differences to occur in two distinct areas: where the original location has reverted to the background, and where the background has been obscured by the object's new position. This is illustrated in Figure 5. There is no easy way to distinguish between the two areas, and this is a problem if we wish to detect the position of the moving object.

It would be useful, therefore, to have some idea of what constitutes the background, so that we can detect changes *relative to that,* instead of merely detecting changes.