THE CONTEXT OF EVERYDAY ACTIONS: USING PERSONAL CONTEXT FOR VISUAL CONTEXTUAL AWARENESS ON WEARABLE COMPUTERS

LI-TE CHENG

The Context of Everyday Actions: Using Personal Context for Visual Contextual Awareness on Wearable Computers

by

Li-Te Cheng

A thesis submitted to the
School of Graduate Studies
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in
Electrical Engineering

Faculty of Engineering and Applied Science
Memorial University of Newfoundland
St. John's, Newfoundland, Canada

October 2002

Abstract

Advances in miniaturization and computing power have set the scene for the emergence of powerful wearable computer systems capable of active computer vision. A lightweight miniature multimedia computer can be worn by a user, receive input from a camera to sense the local environment, present virtual annotations on a heads up display, and network with others through a wireless modem. Applications include field repair, on site tele-medicine, and augmented tourism. The very notion of such a wearable, networked, augmented reality system has an inherent appeal, as it liberates computing from the desktop and integrates computing with everyday activities.

The objective of this research is to demonstrate that augmented reality interfaces can be achieved using basic 1-D or 2-D imaging methods. The notion of "Personal Context" is introduced to address information overload in augmented reality by taking a user-centric model in gathering awareness and context information. Two functional personal context prototypes were built and discussed, showing potential for piano and dance applications. Balancing the body-centric focus of personal context, the idea of mosaicing as a world model for augmented reality registration and telecollaboration is also presented, and realized in another working proof-of-concept system. Finally, a systematic study on accuracy, reliability, and speed of existing and new mosaicing methods (including the method used in the mosaicing prototype) was conducted, identifying their strengths and weaknesses as engines for augmented reality world modelling.

Acknowledgements

This thesis owes a great deal to my supervisor, Dr. John Robinson. His unending support, insight, and guidance have brought inspiration, focus, and direction. Also special thanks to John for organizing the last minute thesis run in October.

Also I would like to thank my colleagues in the Multimedia Communications Lab, particularly Charles Robertson, for their support. For almost four years, the lab has been a place of hospitality, sharing, and ideas. Additional thanks again to Charles for hosting me in October to finish things off.

It was also great to see how the Multimedia Communications Lab has changed into the Computer Engineering Research Lab. Thanks to Reza, Dr. Moloney, David Press, and the other lab members, faculty, and staff for providing space and support in October.

I would like to mention thanks to my manager at work, Irene Greif, who still let me join the Collaborative User Experience Group at IBM Research (formerly known as Lotus Research) when the Ph.D. remained unfinished, and for allowing me to return in October to finish things.

Special acknowledgments go to the Natural Sciences and Engineering Research Council, the Faculty of Engineering and Applied Science at Memorial University, and the various funders of the Multimedia Communications Lab for their generous sponsorship.

Finally, I would like to thank my parents and my brother for their constant spiritual support and encouragement. Thank you, Tina, for waiting so long, and being with me even when we are countries apart. And thanks to Him... for everything.

To Mom, Dad, Min, and Tina who had to wait so long

Contents

List of Figures

List of Abbreviations and Symbols

1-D	One Dimensional
2-D	Two Dimensional
API	Application Programming Interface
AR	Augmented Reality
BMP	BitMaP
CD	Compact Disk
COM	Component Object Model
COTS	Commercial Off The Shelf
CSCW	Computer Supported Collaborative Work
CT	Computed Tomography
CV	Computer Vision
ECG	ElectroCardioGram
FFT	Fast Fourier Transform
GPS	Global Positioning System
GIF	Graphic Interchange Format
GIS	Geographic Information System
GL	Graphics Library
GUI	Graphical User Interface
HANDEL	HAND based Enhancement for Learning piano music
HMD	Head Mounted Display
IPX	Integrated Packet eXchange

JPG, JPEG	Joint Photographics experts Group
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light Emitting Diode
Mb	Megabyte
MCL	Multimedia Communications Lab
MCLGallery	Multimedia Communications Lab Gallery library
MGL	Multimedia Graphics Library
MHz	Megahertz
MSE	Mean Squared Error
NTSC	National Television Standards Committee
OpenGL	Open Graphics Library
OS	Operating System
PC	Personal Computer
PCMCIA	Personal Computer Memory Card International Association
PDA	Personal Digital Assistant
PGM	Portable Grey Map
PPM	Portable Pixel Map
RAD	Rapid Application Development
RAM	Random Access Memory
RGB	Red Green Blue
ROM	Read Only Memory
SMIL	Synchronized MultImedia Integration Language
TCP/IP	Transmission Control Protocol/Internet Protocol
UI	User Interface
USB	Universal Serial Bus
VE	Virtual Environment
VR	Virtual Reality
Win32	Windows 32-Bit

XML	eXtensible Markup Language

Chapter 1

Introduction

1.1 Motivation

Today, handheld and worn devices like pagers, cell phones, and personal digital assistants (PDAs) are acceptable in everyday activity. Easily consulted for timely calendar, personal correspondence, and news notifications, they illustrate the drive towards technology that can be worn by the person "on the go". Besides ongoing hardware challenges like miniaturization, processing speed, and power consumption, designers of these devices must struggle with human-computer-interface issues. These issues include entering full-text data with numeric keypads and presenting relevant personal data onto tiny low-resolution displays.

One way to address the limitations of small handheld displays is to integrate the display with the user's field of vision. This can be accomplished today by off-the-shelf hardware. The visuals are provided by a miniature see-through computer display worn like a pair of glasses and a lightweight multimedia computer smaller than a wallet. Some data input and output options include a tiny coin-sized head-mounted camera sensor, microphone and earphone, wireless modem, and a one-handed keyboard. Such a portable and nomadic system can act as a platform for augmented reality, which enhances the user's real world experience with computer generated information. This information can be sensitive to what the user is experiencing at the moment, and it can be associated with relevant portions of the environment [11].

Such a "wearable computer" system offers greater functionality than today's PDAs, laptops, or cell phones, yet combines their strengths [227]. The portable nature of the system allows it to be taken almost anywhere, like a PDA. Laptop computing power afforded by the computer and attached devices allow the system to actively sense the user's environment, and overlay graphical and audio data. A doctor using such a system could access patient records at the bedside, and have interns around the world watching through the worn camera. A technician can carry out instructions, drawn on the display and described by remote experts. A tourist can enter a historical site, consult with a virtual tour guide, and record the visit on video. A hiker can explore a park, bring up the latest weather report, consult a map, and use GPS to navigate, set virtual landmarks, and annotate the journey with personal voice and video entries. Behind all of these diverse applications, the potential of this wearable, augmented reality computer system lies in its ability to liberate the user from the confines of the desktop, and integrate itself invisibly with the user's daily routine.

While the hardware and applications behind the augmented reality wearable computer platform can be defined clearly, the human-computer interface issues remain unsolved. Such a platform differs from traditional PDAs and laptops with its awareness of the environment, which is used to enhance the wearer's interaction in the context of current activity, and to facilitate collaboration with remote colleagues. This research attempts to address how environmental awareness can provide an interface for wearable computing devices. A fundamental underlying research problem is image registration. It drives the graphical information overlay in augmented reality, making the overlay consistent to head orientation, and relevant to what is being seen. Be it a series of instructions, a diagram, a 3-D model, or a map, the virtual overlay may be attached to an object in the real world scene, and the computer system must maintain this "lock" as the user moves around by using information from the camera or other sensors.

1.2 Research Objectives

The objective of this research is to investigate how augmented reality interfaces can be achieved using basic 1-D or 2-D image registration to form a world model and efficient visual processing methods.

The Motivation section describes the wearable augmented reality computer platform envisaged in this research. In essence, it is a miniature multimedia computer - as small as a PDA, but as powerful as a high end laptop. The system is equipped with a see-through display to provide visual based augmented reality, earphone, and microphone to provide audio output and input, a camera to provide visual input for registration, and a wireless modem to interface with other wearable or desktop computers. This platform is considered wearable due to its lightweight and unobtrusive nature, i.e. it can be worn comfortably and used during everyday tasks like walking and sitting. The platform is considered an augmented reality system due to its ability to augment the wearer's real-world experience with graphical overlays provided by the display, and by audio annotation from the earphone.

On this platform, this research proposes to build working proof-of-concept wearable augmented reality interfaces, incorporating environmental awareness, and collaboration with remote colleagues. The platform, built from a minimal set of low cost, off-the-shelf hardware, must operate at a fast frame rate and provide fast responsiveness to user interaction. The system employs image registration to ascertain the user's head orientation from the head-mounted camera's video data, and to provide real-time and relevant graphical overlays.

1.3 Contributions

This thesis presents four major contributions:

Personal Context

I introduce the notion of "personal context" in this work, where a wearable computer system's awareness and timely presentation of contextual information is achieved solely from the interaction between the user's body and the environment, from the user's point of view. The approach is software-based, relying only on camera input, and using simple and fast 1-D and 2-D vision algorithms. This introspective view of environmental awareness exploits the constant presence of the wearable computer user in changing dynamic surroundings. I built two working proof-of-concept applications to demonstrate "personal context" in computer aided learning in music and dance.

The Mosaic is the Interface

An image mosaic combines individual snapshots from a camera into a single view of a scene through interframe alignment transformations. In effect, many low resolution snapshots taken at different viewpoints in a scene integrate into one, high resolution composite, with a field of view wider than any single snapshot. Thus, image mosaicing can present an integrated visual of a remote environment, as well as a mechanism for image registration for augmented reality on wearable computers. I present a working system that uses mosaicing not only for the computer's convenience in registration and remote display, but also as a platform for remote collaboration between the human desktop user and wearable computer user. The image mosaic becomes a sketchpad interface for humans to draw annotations that are embedded into the augmented reality of the wearable computer user.

Systematic Comparison of World Modelling Methods

Although this research uses image mosaicing as the engine for registering virtual overlays onto the wearable computer's augmented world, and a big window for the remote collaborator, I employ only one of many possible mosaicing techniques. To investigate the potential of other mosaicing methods for augmented reality world modelling, I conducted a comparative study of research and commercial image mosaicing algorithms against criteria relevant to wearable computer applications.

Software Infrastructure

I created a C++ software infrastructure, MCLGallery, to enable the creation of the prototypes and evaluation systems listed in the above contributions. The software infrastructure provides a single environment for image algorithms, graphics rendering, networking, and rapid prototyping, while building on top of the best libraries available at the time. With several updates, MCLGallery was also used by fellow researchers for other projects in the same lab over the span of four years.

1.4 Constraints and Assumptions

The focus on wearable computing imposes constraints and assumptions for the final system implementation that are more stringent than a workstation based solution. The hardware requirements of this research are as follows:

Portable while Operational

The final system must function perfectly while the user is moving or standing still.

Power Efficient

The final system must operate on batteries for at least a major portion of the user's working day.

Wireless Networking

The final system must be capable of connecting with a LAN wirelessly, at least within the same building, and at a data rate that can at least support two-way video and audio transmission.

Outdoors Capability

The final system must be rugged enough to operate outdoors for a brief duration under mild weather conditions.

Active Video Sensing

The final system must use a video camera (mounted on the user's body) to actively sense its environment during operation because this research is interested in investigating image processing techniques for registration.

While most of the hardware requirements are derived from the wearable computer aspect of the problem statement, the software requirements, which constrain the underlying algorithms to be determined by this research, are determined by the augmented reality aspect. Specifically these software requirements are:

Active General Registration

From the live video provided by the camera, the system must perform general registration. That is, the system must at least ascertain the camera's orientation based on the live video information.

Graphical Annotation Overlay

Using the registration information, the system must present virtual annotations (be it text, drawings, or other multimedia) with a graphical representation, overlaid and locked on the annotated real-world object.

1.5 Criteria for Success

In addition to meeting the constraints demanded by the previous section, this research must be measured against several criteria to assess the usefulness of the final algorithm:

Best Accuracy

The registration algorithm must assess the user's orientation and lock virtual annotations on targets as accurately as possible. Objects farther away from the user suffer from a greater potential error (as small angular discrepancies can add up to large displacements), thus the effective range of the system must be determined when measuring system accuracy.

Maximum Reliability

Despite all the efforts that can be made to tune the algorithm's accuracy, there is bound to be a margin for error. Small or large, the error must be compensated for as gracefully as possible by the algorithm. Violations of the algorithm's assumption should not meet with catastrophic results.

Real-time

In order to be used in the field and maintain virtual annotations in the real world, the algorithm must be real-time. The algorithm must continuously function from a live video feed, producing orientation information and rendering the appropriate overlays as quickly as possible.

1.6 Thesis Organization

This thesis document presents the motivation, goals, background, methods, and findings from my research into wearable augmented reality interfaces. The chapter by chapter breakdown of the thesis is as follows:

Chapter 1 (this chapter) introduces the general motivation and goals of this work.

Chapter 2 covers the background behind wearable computers and augmented reality, the two major domains related to this research. The spectrum of research is highlighted and the two major areas of active research, registration and context awareness are discussed. A broad, general survey is taken to familiarize the reader with terminologies and technologies used in these research areas. However, background material and related work that are very specific to topics covered in subsequent chapters are presented in the context of the respective later chapters.

Chapter 3 establishes the common hardware, software, and algorithmic foundation behind the subsequent chapters. The hardware and software components and infrastructure forming the wearable computer testbed used in the research are described, with the reasoning behind the design decisions.

Chapter 4 introduces the notion of Personal Context as a model for human/wearable computer interaction and its distinguishing characteristics versus context awareness and augmented reality. The computer take an attentive but peripheral role, only providing information when it senses the user's immediate need. The design and workings of two proof-of-concept Personal Context systems are documented.

Chapter 5 shows how image mosaicing can power augmented reality image registration for a wearable computer, and how it enables enhanced remote user/wearable user collaboration. The design and workings of the final system built to demonstrate mosaicing as an interface is presented.

Chapter 6 follows up Chapter 5's investigation into mosaicing with a detailed evaluation of Chapter 5's mosaicing algorithm against other techniques. This chapter surveys a variety of image mosaicing algorithms as engines for augmented reality registration for wearable systems. The experiments, challenges, and results comparing these methods are discussed. In the data analysis, I examine the relationship between accuracy, reliability, and speed. Ultimately, I show what trade offs make a suitable algorithm for wearable augmented reality applications.

Chapter 7 concludes with a summary of results from the previous chapters, discusses the overall rammifications of the research, and directions for future work.

Chapter 2

Background to Wearable Augmented Reality

The contents of a twenty-first century professional's pockets reveals a collection of tools different from earlier years. The pen and paper agendas, calculators, and even laptop computers are replaced with tiny, networked computing devices. Devices like mobile phones and personal digital assistants (PDAs) come packed with features like wireless internet access, email and instant messaging, geographic positioning, full motion colour video, and camera attachments.

Instead of going to the office, today's "on-the-go" professionals carry their office. Rather than reading the latest newspaper and printed memos, readers bring them out of their pockets. Transactions through the office assistant or the desktop telephone give way to spoken voice commands at the press of a button.

But this "brave new world" of pervasive devices has its challenges. How does the rich media offered by a desktop environment get packaged and presented in a fast and mobile environment? Can context cues like location, time, and activity be harnessed to sift through piles of data to give relevant user feedback? What will we find in people's pockets in the next decade? Perhaps nothing at all. Instead, they may wear their office in a wearable augmented reality computer system.

This chapter lays out the relevant domains touched by this thesis by introducing the notions of wearable computers, augmented reality, and their intersection, wearable augmented reality. Their terminologies, general applications, current work, and relationships to today's pervasive mobile technologies are surveyed. Subsequent chapters go into further detail about closely related work and alternate approaches to my work. The following sections identify the open areas for research in these domains.

2.1 Wearable Computers

2.1.1 Definitions

Personal desktop computers, while more powerful than ever, remain on the desk, and thus chain user-computer interaction on the desktop. Today's notebook computers carry almost as much capability as their desk-bound brethren, but inherit the same old static usage model from desktops and typewriters: sit and stare, point (on a keyboard or screen) and click. Wearable computing breaks from this tradition. The wearable computing user paradigm envisages a personal computer to be truly personal: to be worn like glasses or clothing. With heads-up displays, discreet input devices, wireless networking, head-mounted cameras, and other miniature sensors and tools, a wearable computer is aware of its surroundings and acts intelligently based on the current environmental context like a second set of eyes, ears, and even brain [86].

This vision, along with increasing computing capacity in smaller, lighter, and more power-efficient components, is behind recent growth in wearable computing research. Recent developments in consumer personal handheld devices like digital assistants, wireless email pagers, and mobile phones share some of wearable computing's philosophy. For example, Palm and PocketPC PDAs offer email, voice recording, and digital photo modules. On the other hand, handhelds still demand user attention and input around a display like desktop and laptop computers. What defines a wearable computer, and distinguishes it from the current generation of handheld devices, is summarized by the following requirements [87]:

Portable while operational

Unlike desktop or laptop computers, wearable computers should be usable while the user is moving around. This also implies a degree of ruggedness in the design, i.e. the wearable computer must function perfectly in the environment through which the user is moving. And this implies a person can operate the wearable computer effectively while moving. Extreme examples of this are the WetPC, a wearable computer designed for underwater use [174], the Wearable Fire Fighting Ensemble, an advanced fire fighting suit with a wearable information support system for U.S. Navy fire fighters [251], and a wearable system for arctic search and rescue [196].

Hands-free use

Wearable computers should minimize the use of the user's hands. Devices that accomplish this include one-handed chording keyboards like the Twiddler [48], ring devices [78] and voice recognition. Consumer handhelds like a Palm Pilot or a Blackberry pager require two-handed operation for pen-input or thumb-operated keyboard. Hands-free use is an important requirement for various industrial and military applications where the user's hands may be busy in a task. Wearable interfaces also must take special considerations to minimize attention and effort. For instance, a wearable email system may want to use a query-response or easy selectable voice menu user interface [240].

Sensors

Wearable computers should use attached cameras, microphones, wireless communications, GPS, etc. to enable environmental awareness, as well as deliver and store location-relevant information.

"Attention-getting"

Wearable computers should alert the user of important information even when the system is not being used (like an alarm clock or a telephone).

Always on

Wearable computers should be operating, sensing, and responding to their environment continuously, unlike PDAs which are only activated for specific tasks.

Many of these requirements match Mark Weiser's vision of ubiquitous computers [257]. Ubiquitous computers are similar to wearable computers, as they are environmentally sensitive, but differ in that ubiquitous computers are meant to be embedded throughout the environment rather than solely on the user. But wearable computers can work together with ubiquitous computers, such as to provide internal building location and multimedia annotations on the real world using the MIT Locust Swarm system [226]. There is overlap in these requirements with mobile phones and high-end PDAs. Mobile phones and lower-end PDAs still lack the computational power for wearable computing's environmental and contextual awareness.

2.1.2 Applications

Wearable computers have begun to appear in reports from the press [119] [104] [54] [232], and commercial vendors are starting up. Strategic alliances for development and commercialization are being formed, such as between Sony and Xybernaut Corporation [172]. While not in widespread or very active use at present, wearable computer systems have been recognized by industry, the commercial sector, and the military [245] [29].

Example of wearable computers and related technologies can be seen in Figure 2.1. Potential applications of wearable computers include:

Inspection and Validation

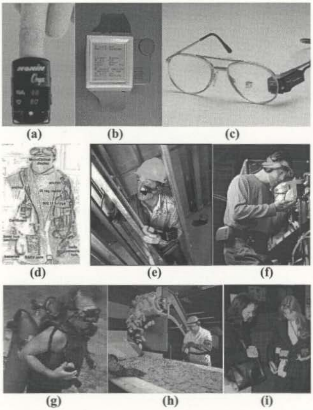


Figure 2.1: Examples of Wearable Computers and Related Technologies

(a) Nonin's miniature medical diagnostic device [169], (b) IBM Research's Linux computer in a watch with wireless connectivity [200], (c) MicroOptical's eyeglass display [49], (d) MIT's MITHRIL wearable vest design [133], (e) Xybernaut's wearable system used in home inspection [55], (f) Xybernaut's wearable system used for repair [55], (g) The WetPC underwater wearable computer [173], (h) GeorgiaTech's chicken inspection system [85], (i) ViA's wearable for mobile airline agents [104]

Wearable computers can be used in the field by inspectors to input voice data, consult multimedia reference material, gather new data, and make decisions on the spot. For example, NASA is investigating the use of wearables to assist the lengthy and arduous inspection of the space shuttle tiles and engines after landing [18] and Georgia Tech is prototyping and evaluating a voice input system for food inspection workers on the processing plant floor [162] [163]. BOC Gases, which sells nitrogen, oxygen, and carbon dioxide for manufacturing, has their inspectors use wearables running troubleshooting and validation software to inspect and validate customers' sensor systems used for temperature control on site [4]. Bath Iron Works, an advanced naval ship designer and builder firm, equip their inspector liaisons with wearables to record trouble spots during ship inspections and get remote advice via wireless connections to databases and experts at the main office [250]. The close integration of the wearable with the user's body is cited as a key advantage for home inspection using wearable computers [252].

Maintenance and Repair

Wearable computers can be used on the field by technicians to perform computer assisted maintenance and repair tasks. The wearable computer can act as a hands-free multimedia reference manual, use wireless networking to access databases and maintenance logs, and transmit video from the field to an expert for consultation. For example, Boeing, Honeywell, Carnegie Mellon University, and Virtual Vision Inc. have developed prototype systems to assist in aircraft maintenance [159], and Carnegie Mellon University and McDonnell Douglas Aerospace developed wearable computers for military vehicle maintenance [220] [57]. NASA is investigating the use of a hands free wearable computer to assist astronauts performing hand-intensive work in space [18]. Kraut evaluated the use of on-site video consultation to complement a technician's work [128]. Xybernaut Corporation has customer case studies featuring their wearable systems used by technicians for manufacturing plant, field service, vehicle fleet, and power utilities maintenance tasks [5]. NorthWest Airlines has deployed ViA wearable computers for their technical maintenance personnel to increase hands-free tasks and eliminate paper from repair and inspection work [249].

Medical Diagnosis and Telemedicine

Medical oriented wearable computers can assist medical users in diagnosis by providing virtual heads-up style read-outs (e.g. ECGs), patient records, and wireless access to databases and references. Telemedicine can be applied as well, by mounting video and audio inputs on the user and transmitting information to remote experts for teleconsultation (wireless communication may be desirable for simplicity of installation). The patient could also be wearing a simple wearable computer, recording and transmitting vital signs to the visiting physician's wearable computer, or to the hospital's network. Medical applications of wearable computers, particularly for on-site emergencies, have been discussed in the University of Oregon's MediWear project [20]. ViA reported trials with medical wearable computers [104]. Although not actual wearable computers, family doctors are starting to use miniature diagnostic and treatment tools, and such devices could possibly be interfaced with wearable computers [81]. Some of these miniature medical diagnostic technologies are already being investigated in conjunction with wearable computing to create computers responsive to human emotional and medical states [190].

Inventory Collection

The wearable computer can act as a hands-free assistant in cataloguing and performing inventory checks. Inventory can be made "smart" by planting ubiquitous computing devices on them, which would be interfaced by a wearable computer. FedEx identified the potential benefit of wearable computers to enable couriers to monitor and track parcels effectively [245].

On-Site Emergency and Field Services

Wearable computers can provide emergency workers immediate up to date information (e.g. overall situation status, weather reports, location specific data) via wireless communications, while they perform their tasks. Coupled with sensors such as a camera, a wearable computer at an emergency site can transmit valuable data for relief effort planning, teleconsultation, news, and insurance purposes [104].

Northwest Airlines' Portable Agent Workstation uses a wearable system to assist mobile airline agents in providing fast ticketing and information services to customers on the move, and to expedite the waiting process in long lines [248].

Computer Aided Instruction

In conjunction with the maintenance and repair, as well as the medical applications, wearable computers can provide on site computer aided instruction through presenting multimedia and exploiting situation-awareness through sensors. Part of the work at Boeing and the military is oriented to this application [159] [57]. There is also potential for teaching students in a laboratory setting (e.g. have sensors in an experiment communicate with the student's wearable computer to provide the student immediate readings; have a hands-free; multimedia tutorial given via the wearable computer; transmit video taken by the student's wearable to the instructor for evaluation).

Surveying, Mapping and Navigation

By making them aware of their environment through the use of sensors, such as GPS sensors, wearable computers can be valuable tools in surveying, mapping, and navigation. For the surveyor and geographer, a wearable computer can log user specified positional and geographical data, and possibly present the user with a virtual map of what has been mapped already [233]. Farmers can couple GPS and personal notes with crop, climate, and soil conditions on the field [246]. Surveyors benefit from integrated GPS and surveying software running on a rugged, long-lasting, wearable computer system [247]. For the traveller, location-context sensitive information and directions can be provided by a wearable computer [226] [84] [1]. Coupled with a video camera, a navigation savvy wearable computer can act as a guide for blind users.

Personal Communication and Knowledge

Wearable computers have been heavily used as mobile terminals to browse the web and check email [227][240] [144], acting as a personal communications appliance (like PDAs). They can act also as knowledge appliances, where the user can enter notes and appointments, and use sensors and software to provide a supplement to human memory (e.g. use a video camera to do face recognition) [148] [222]. Wearable computers can overcome personal disabilities [147] and facilitate communication between the user and the environment, such as through a wearable sign language-to-speech translator [227]. Novel designs for personal use include an all-audio based wearable computer for personal work [193] [206]. The multimedia data-gathering and data-recall capabilities of a wearable can combine a photographer's and reporter's role for journalism [53].

2.1.3 Research Issues and Benefits

As seen in the previous section, wearable computers cover a broad spectrum of useful applications, and much research is underway. An area in constant development is in hardware design - specifically in power consumption, architectures for more compact and lighter designs, and portable displays [87]. For instance, MicroOptical demonstrated a 320 pixel by 240 pixel resolution display that unobstrusively fits in a pair of eyeglasses [49]. Beaming a high resolution image directly onto the retina, the virtual retinal display, proposed in [121], is now available in a wearable head-piece format for military and business partners [51].

Some research work introduces design methodologies, principles, and architectures for wearable computers [15] [75] [162] [163], and assesses the effectiveness of using wearable computers [163] [171]. Also, research work that involve novel sensor technologies, such as galvanic skin response, blood pressure, respiration, to create wearables responsive to human emotional and physical states are being investigated [190].

But the key theme in the wearables vision is redefining human-computer interaction beyond desktop paradigms. A barrier to commonplace adoption of wearable technologies, in addition to the above hardware issues and manufacturing costs, is the user experience. A challenge lies in software that leverages the hardware and sensors to enhance the wearer's awareness and experience far beyond conventional handheld devices. The cited examples do provide some added benefit, but in narrow niche applications. The following sections on augmented reality and context awareness address this human/wearable computer interface problem.

2.2 Wearable Augmented Reality

2.2.1 Definitions

> "Augmented reality (AR) is a variation of virtual environments (VE), or virtual reality... VE technologies completely immerse a user inside a synthetic environment. While immersed, the user cannot see the world around him. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world."
>
> *Ronald Azuma, A Survey of Augmented Reality [11]*

Observing that spectacles augmented the human sense of sight, the scientist Robert Hooke imagined that other devices could be developed to augment the other human senses [202] [94]. Augmented reality is the domain that realizes this, enhancing what the human senses in the real world with virtual information, be it graphical, audio, tactile, etc.

With the recent developments in real-time high resolution computer graphics for virtual reality, computer games, and the media (notably in television and film), there has been a large focus on visually based augmented reality systems, although other modalities have begun to be studied (e.g. 3-D audio). Also, the visual sense is one of the most important senses, as most of our daily activities depend on it. So it is only natural to have an emphasis on the visual sense in this background chapter, and in this research.

The potential of augmented reality systems lies in providing a powerful means of information visualization and sensory fusion. By merging the real world with the virtual, augmented reality vividly presents how the abstract and the invisible are related to reality. Typically, the augmented reality visual display would be head mounted and so the information is readily combined with the user's field of view. Therefore, the user does not need to consult a monitor or a hand-held device for the augmented information.

Augmented reality offers an unrestrictive and "attention-getting" interface for wearable computing systems. An unrestrictive interface allows one to do other things while doing computing or experiencing the environment. Its ability to overlay the virtual on the real is a powerful way to present an enhanced view of reality. Wearable computer systems typically use a see-through heads-up display that is also used in some augmented reality work. A wearable, augmented reality computer system empowers the user to step outside the laboratory and explore the real world in a hands-on manner.

As specified by Azuma [11], an augmented reality system has the following three characteristics:

Combines real and virtual

This characteristic distinguishes an augmented reality system from a virtual reality system. Whereas virtual reality immerses the user in a new, synthetic computer generated environment, augmented reality leaves the user in reality, and incorporates computer generated information into this reality.

Is interactive in real time

The user should be able to interact with reality and the virtual information in real time. Also, the virtual information should be able to update itself in real time. Otherwise, the lack of real time interaction with the virtual data will create a noticeable discrepancy between the virtual information and the real world.

Is registered in three dimensions

Registration involves the "locking" of virtual information to the associated real world object, regardless of the user's current position and orientation (e.g. maintaining a virtual wireframe mesh around a moving, rotating real object). To an augmented reality user, virtual objects that are registered in three dimensions appear to have the same optical characteristics as their real counterparts. This can be accomplished by several means, including rendering virtual information in the same depth plane as targeted objects in the scene [151], and by correlating head-tracking sensor data against environmental sensors or a predetermined model of real objects to generate a 3-D virtual object rendered with respect to viewing angle and perspective depth [11]. Three dimensional registration is specified to distinguish augmented reality systems from the two dimensional registration that can occur within movie productions mixing real life with traditional animation, video editing and "blue-screening" techniques. This characteristic distinguishes augmented reality from some wearable computer systems. A wearable computer might employ display hardware similar to an augmented reality system (e.g. a see-through head mounted display), but does not necessarily require registration. For instance, wearable computers are often used for reading email or retrieving text data, which do not require any registration, while the user is on the move. However, three dimensional registration does not necessarily require full three-dimensional analysis of the real world scene and foreground/background object analysis. Without a considerable amount of hardware infrastructure, it is difficult to do full three-dimensional scene analysis for registration, and most methods described in the following sections assume few foreground objects of interest, little or no object manipulation and occlusion effects, and in some cases, a flat background merged with most of the foreground objects.

2.2.2 Augmented Reality Research and Applications

There are a number of recent applications and developments in augmented reality systems [11], as illustrated in Figure 2.2 (explanations of each of the examples in this figure follow). These applications overlap with wearable computer applications, but focus on visualization issues as opposed to mobility benefits (e.g. in medical applications, the feature of wearable computers lies in their portability, whereas augmented reality provides a new means of seeing the environment). In other instances of apparent overlap, the augmented reality implementation leans more towards a non-wearable computer solution (e.g. the entertainment applications of augmented reality lend themselves towards a specially equipped room as opposed to a portable computer).

Medical Visualization

Augmented reality systems have been used as an aid for visualization and training for surgery. For instance, by incorporating sensor and video data, they can overlay graphical information to assist minimally invasive surgical techniques (where normally it is difficult to view the very small operating region). An augmented visual overlay displaying readouts of medical sensor instruments can provide the surgeon useful information without the need to look up from the patient. By overlaying and locking graphical instructions (text and diagrammatic information) on practice cadavers and body parts, augmented reality can provide a computer-assisted hands-on environment for training doctors in new procedures. Examples of current works in these areas in-

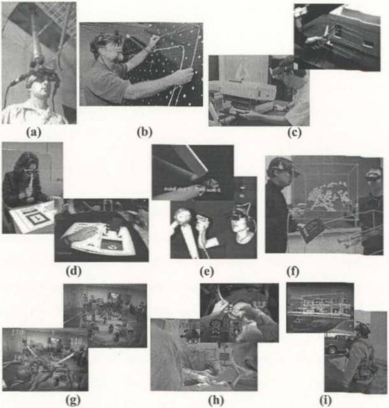


Figure 2.2: Examples of Augmented Reality

(a) Ivan Sutherland's original augmented reality display [231], (b) Boeing's AR aircraft wiring assembly AR [28], (c) The KARMA augmented printer application [72], (d) The MagicBook AR-enhanced book [24] [25], (e) University of Columbia's AR construction application [73], (f) The Studierstube collaborative AR system [77], (g) The AquaGauntlet AR game [103], (h) AR-enhanced surgery at the Wallace-Kettering Neuroscience Institute [90], (i) The U.S. Navy's Battlefield AR System [165] clude UNC Chapel Hill's work on an augmented reality ultrasound system, which effectively gives the user "X-ray" vision on an ultrasound patient [11], research into augmented surgical procedures and training at the MIT AI lab [83] [19] [82], a virtual ECG readout for augmented reality at University of Washington [115], and an augmented reality system to rehabilitate Parkinson's patients suffering from motor problems [256]. Several systems take imagery from video, CT scans, and microscope pictures to construct enhanced models for diagnosis and preparation [8], and recently, for actual procedures [90].

Manufacturing and Repair Reference and Instruction

Augmented reality systems can enhance manufacturing and repair work. By overlaying virtual information on the worker's field of view, an augmented reality system can act as a hands free tool or multimedia reference manual. Through registration, the information provided to the user remains fixed on the relevant part regardless of the worker's orientation, providing a location-aware application. An example of work in this application area includes Feiner's KARMA system at Columbia University (Figure 2.2(c)), which is an augmented reality system to enable users to maintain a laser printer [72]. Another augmented reality system at Columbia University is a system to assist construction of space frames [73]. Boeing also has has combined augmented reality with wearable computers to assist workers to build wiring harnesses for aircraft electrical systems [159]. A consortium of local and international companies in Germany, including AUDI, Volkswagen, Siemens, Ford, and Daimler-Chrysler are proceeding on a four year project to develop AR technologies to enhance assembly, production, and manufacturing [46].

Annotation and Visualization

Augmented reality has the potential of annotating and visualizing the real world with overlaid, virtual information. Such capabilities provide timely information on the current situation, as realized by the traditional heads up displays used for military pilots and infantry. Annotation and visualization could be as simple as a virtual "post-it" note attached to a real world object, or as complex as a wire mesh graphic with detailed labels and hyperlinks. Examples include Feiner's "Windows on the World" system, where X Windows can be incorporated into the real world or attached to people [69]. Columbia University's Augmented Architecture system allows civil engineers and architects to visualize hidden properties of actual buildings for inspection and assessment [74] [255]. Fitzmaurice's work demonstrates an augmented annotation system for a real map [76]. Linking the real world with the web with virtual hyperlinks and multimedia information are discussed in [148] [223]. Starner et al. present another example with annotating features in the environment under human supervision or by face recognition (mimicking an "augmented memory") [148].

Robot Path Planning

In non-real time robotic control (e.g. teleoperation of robots over long distances such as on other planets), augmented reality can provide a means for a user to plan out a robot's trajectory. The user can operate a virtual version of the robotic system, the motions of which are overlaid on the actual image of the robot in the real world. By viewing the virtual robot with respect to the real world image, the user can judge a proper path for the robot. An example of such a system is the University of Toronto's ARGOS and ARTEMIS systems [197] [67].

Art and Entertainment

Augmented reality has the potential to enhance the entertainment industry. Traditionally done offline, virtual sets, props, and actors could be incorporated in real-time via an appropriate augmented reality system. Although the computing power to produce this in real-time at studio quality is far off, a current augmented system may be useful in early prototyping and conceptual work. For the gaming market, an augmented version of "laser tag" can be achieved (similar to the virtual training facilities for U.S. infantry). For the toy market, augmented systems could provide to children a virtual playpen or virtual sets and virtual characters, for their real world toys (akin to the blending of reality and fantasy from director Robert Zemeckis' 1988 movie "Who Framed Roger Rabbit?"). The ALIVE project at MIT demonstrates the incorporation of responsive, computer animated characters in the real world [145]. Mann has spent years exploring how altering reality can be, among other things, a new form of photographic art, collaborative visual artwork, paintball games using light paintings, and a form of film documentary commenting on privacy violations [151].

2.2.3 Current Wearable Augmented Reality Research

Researchers have begun developing and studying augmented reality on wearable computing platforms. A few examples are noted below.

Schmalstieg et al [77] present a shared, collaborative augmented reality space, with each user viewing 3-D models on a head-mounted display. While their initial work is tethered to a lab server, recent results demonstrate their system operating for multiple users, multiple devices, and for in-situ and telecollaboration contexts [211]. Computer supported collaborative work (CSCW) using augmented reality is distinguished by the use of virtual graphical models, augmentation of the real world, cooperation between in-situ and remote users, and independent viewpoint control and viewing perspective for each user.

Billinghurst et al [23] examine the interface issues for a wearable augmented reality system, and uses a 3-D user-centered cylindrical space to present spatially organized information without overwhelming the user. His recent work([114] [25]) marries Ishii's idea of tangible computing devices [110] with augmented reality. Specially marked books and desktop spaces trigger interactive augmented reality and virtual reality environments. A wearable augmented reality computer user can view graphical characters and buildings attached to the books and spaces, move them around with a real physical tool, and become fully immersed in a virtual reality version of the same scene.

Feiner et al [112] discuss the need to present only the most relevant overlays to give a concise and manageable augmented reality interface. This prevents information overload and helps present a system aware of its surroundings as well as sensitive to the user's needs. Feiner et al [70] summarize wearable augmented reality systems Feiner's group built to provide graphical annotations on buildings for a user wandering around a campus and as structural visualizations for architects and construction workers.

MacIntyre and Kooper [144] demonstrate the beginnings of a 3-D wearable computer web browser. Web information is presented on real world surroundings and is used to alert the user. This is similar to Spohrer and Stein's idea of "information in place." [225].

Mann observes that augmentation leads to information overload, since the computer now has the ability to clutter the real world with too many distracting overlays. Thus, he proposes the novel notion of diminishing reality to cope with information overload [151]. His "mediated reality" wearable computer system allows him to hide undesirable and distracting advertising from his view [152]. Using only a head-mounted camera and vision algorithms, he can replace the wallpaper on real-world walls with computer terminal windows [150]. Besides diminishing reality, Mann's "mediated reality" allows him to control how reality itself is experienced. For example, reality can be slowed down to allow a user to see hidden details too fast for the human eye; optical manipulation can be applied to give the user different fields of view. Thus, his work encompasses augmentation, diminishment, and manipulation of reality. The reader is referred to Mann's book for further details in this advancement beyond augmented reality [151].

The work above, as well as many others, use AR as a medium to paint information around a mobile wearable computer user through a see-through head mounted display. The most compelling aspect of AR's superposition of virtuality on reality is the "registering in 3 dimensions" characteristic. This has a powerful effect as a wearable computer user moves around: scenery remain labelled and virtual objects reveal their three-dimensional nature. However, the registration of virtual overlays in augmented reality is a difficult problem.

2.3 Registration

2.3.1 The Problem

Unlike the domain of interface research, the registration problem is well defined and focused, but lacking any easy answers. In essence, the problem is to have a computer establish and maintain a lock on moving targets, as well as determine its own orientation in the environment using a limited set of sensors (usually only one camera). Typically the registration is graphically represented as text or line graphics attached to the current image scene taken by a video camera. The information remains attached regardless of the camera motion, and the presentation remains consistent with the geometry of the scene (e.g. a wireframe graphic rotates with its rotating target). Although this problem is similar to target tracking, the registration problem must also consider the distance and orientation of the viewer with respect to the environment. For example, a wireframe may be attached to the entire room (including the walls), and must follow the camera's panning and tilting. This requires estimation of the sensor's orientation with respect to the environment, which is known as the "general" registration problem. The focus of this research is solving the general registration problem using an implementation based on a wearable computer platform.

A number of papers, as cited in [11], suggest that it may be impossible to achieve real-time registration with hardware tracking alone. Hardware tracking includes magnetic, ultrasonic, and mechanical tracking schemes, each of which have their limitations and strengths (magnetic tracking is non-line of sight but suffers from metallic interference, ultrasonic tracking is reliable but requires line of sight, and mechanical tracking offers precision and robustness but has limited range). A vast range of vision based registration schemes have been investigated [30], but require significant processing time on a mobile computing platform.

2.3.2 Approaches

An example of an effective hardware-based solution for registration is Bajura and Neumann's closed loop feedback system, which provides very accurate registration [14]. Their system makes use of a magnetic head mounted tracker to provide general orientation data, and images from a head mounted video camera as a reference for computing differences between the registration model and reality. The differences are used to correct the virtual overlay presented by the augmented reality display. The display is an immersive head mounted display, which lets the system synchronize the rendering of corrected images with the presentation of live video to the user. Because the system is immersive, there is only a small amount of overall lag in presenting the video. This lag, however, would be readily noticeable in a see-through display, since the system cannot synchronize corrected information with what is seen in reality. Another limitation is that the system relies on LEDs to allow the video camera to quickly obtain scene information.

Another hardware-based registration method is presented by Azuma and Bishop. Their method relies on inertial sensors to provide prediction information to assist registration, and depends on an array of ceiling-mounted LEDs to determine head orientation accurately [9]. [10] demonstrates a bulky registration system based entirely on gyros, GPS, an orientation sensor, and specialized computing hardware to provide fast and accurate outdoor registration. [211] uses magnetic head mounted trackers and an instrumented pad for multi-user augmented collaboration with virtual 3-D models.

A network of devices transmitting position codes via infra-red can provide an approximate indoor positioning reference system to help AR, such as [226] and [34].

An ultrasonic-based system deployed over an office is discussed in [91].

Instead of using hardware such as trackers and LEDs to assist registration, hardware could be used to assist computational speed of registration algorithms. An example of this is Uenohara and Kanade's system, which use specialized digital signal processing and video capture boards to achieve registration at 30 frames per second [241]. Their method uses template matching to find feature points for registration. Although the results are fast and accurate, the template images of the object to register and sample feature points on the target object must be known beforehand.

In contrast, the registration methods of Szeliski and Coughlan, and Mellor are examples of purely software based registration techniques, requiring no a priori information, and producing notable accuracy [235] [155]. The Szeliski and Coughlan method parallels later work done by Shum and Szeliski, and Mann and Picard, in image mosaicing [219] [153]. The method is a hierarchical scheme, applying a form of optical flow analysis on smaller, lower-resolution video frames, to provide initial registration estimates, and then iteratively improve these estimates by analyzing higher resolution versions of the video frames. The results appear to be robust and accurate, but the computation requirements are complex, and can incur a noticeable processing delay. Neumann and You [167] can derive 3-D estimates slowly from long video outdoor sequences with minimal intra-frame differences via optical flow analysis. Simon et al [221] exploit planar surfaces in the scene to concentrate automatic extraction of control points for somewhat quicker video frame motion estimation, although this method requires initial manual identification of a planar surface.

As opposed to the pixel-by-pixel optical flow approach taken by Szeliski and Coughlan, Mellor uses circular fiducials to achieve registration. Mellor's algorithm, similar to the BBC Virtual Studio work in [238], actively searches the current scene for distinctive circular markers, and computes correspondences between the markers in the current scene with the previous to find general registration information. Although the algorithm for this is very simple and reasonably fast to compute, it requires placement of distinctive fiducials into the scene, and does not address problems such as occluded fiducials, and fiducials outside the current field of view. [166] has experimented with different shaped and coloured circular fiducials to register virtual annotations. Billinghurst et al use multiple rectangular fiducials to achieve robust 3-D object registration [114].

Considerable current work is attempting to implement hybrid schemes, merging the speed of hardware with the accuracy of software [11] [243] [139] [166]. Outdoor positioning infrastructure, such as GPS, can be combined with computer vision to provide accurate terrain registration [17]. Thomas et al [237] use fiducials, sensors, and GPS to obtain position and orientation, and a commercial game 3-D graphics engine to provide AR graphics.

In any case, the search for a real-time registration scheme that robustly operates under general conditions (e.g. outdoors as well as indoors) continues.

2.3.3 Metrics

Besides tackling the registration problem, AR researchers also consider methods to evaluate registration. Evaluation schemes are closely related to calibration, where measurements from different sources (often from a magnetic tracker and a vision system) are evaluated, compared, and used to refine the registration model. AR registration work has largely focused on head motion and body position, and so metrics concentrate on head pose angle and distance.

For instance, Hoff and Vincent analyze head pose in detail from a variety of head tracking sensors [93]. Hoff and Azuma show a minimal calibration algorithm to improve accuracy of a magnetic compass for outdoor augmented reality registration [92]. Tuceryan and Navab [239] show how to calibrate a magnetic head tracker quickly with only one reference point. MacIntyre and Coelho use registration error models to help warn the user of the amount of error, and trigger different registration algorithms depending on proximity to target objects [143]. Wagner et al [253] compare a complex visual registration scheme to find 3-D correspondences versus gyro-sensors.

A shortcoming in existing evaluation methodologies is the use of internal parameters to perform measurements. This restricts the available set of registration techniques for evaluation. In pure hardware registration schemes, the analogy is trying to take advantage of internal readings from software drivers, or take direct measurements from the registration devices themselves, if the experimenter knows where to look and how to extract the data. Similarly, in software methods, evaluators would face the challenges of understanding opened source code to derive the needed measurements, or relying on whatever user interface the software can provide for experimental data, the latter being especially the case for closed commercial products.

In circumstances where an experimenter wants a wide breadth of techniques to evaluate, and still wants a standardized set of tests, a "black box" approach is preferred to deriving measurements from registration methods. Chapter 6 goes into detail about a black box evaluation methodology for software based mosaicing registration algorithms.

2.4 Context Awareness

2.4.1 Definition

Whereas augmented reality is mostly focused on the presentation of an enhanced view of the real world, context awareness concentrates on gathering data from the real world to fuel the presentation. This data is context, i.e. according to Dey and Abowd [66]:

> "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered

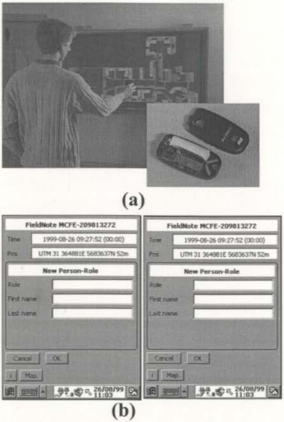
> relevant to the interaction between a user and an application including the user and applications themselves"

Context-awareness systems take a user's surrounding context and evaluate whether if it can help provide relevant information and services to help the user's current task. Critical types of context include location, identity, activity, and time [66] [181]. These types can be measured in the environment by physical and logical (i.e. deductive software) sensors. The measurements can be composed into a vector describing the current situation around a user and trigger scripted responses sensitive to immediate activity [212]. The selected context types and system design are derived from careful study of the desired task to enhance [214]. This allows a wearable computer to give timely and useful assistance to the user through an augmented reality interface.

2.4.2 Previous Research and Applications

Research and application in context awareness can be divided by the different types of context, namely location, identity, activity, and time. A few examples are illustrated in Figure 2.3.

Location and identity are often mixed together. Indoor positioning systems like the Active Badge [254], BAT [91], Cricket [194], Locust [226], RADAR [13] and IRREAL [34] use some hardware infrastructure to provide a location reference system. The cited systems use infrared (Active Badge, IRREAL, Locust), triangulated ultrasonic signals (BAT), and triangulated measurements of radio signal strength (RADAR) technologies for positioning. Then each individual carries a device to access the infrastructure, like a badge [254], personal BAT tag [91], an ultrasonic and RF receiver [194], a wearable computer with special receiver [226], a wireless network card [13] or a PDA [34]. Some systems broadcast personal identity and location like with Active Badges, 802.11 wireless cards (via the network card's unique hardware address), and the BAT awareness, and others like Cricket, Locust, and IRREAL use



FieldNote MCFE-209813272

Time 1999-08-26 09:27:52 (00:00)

Pos UTM 31 364881E 5683637N 52m

New Person-Role

Role

First name

Last name

FieldNote MCFE-209813272

Time 1999-08-26 09:27:52 (00:00)

Pos UTM 31 364881E 5683637N 52m

New Person-Role

Role

First name

Last name

Figure 2.3: Examples of Context-Awareness Applications

(a) AT&T Cambridge Lab's Sentient Computing system showing a 3-D visualization of user locations and the portable BAT locator device [132], (b) FieldNote for the PocketPC screenshots showing automatic GPS and time entries [7] location to drive location-specific information services like personal annotations or map data towards the device.

Identity can be established by encoding a user code onto the computing device itself (e.g. authenticating a user after entering a username and password), whether it is a wearable computer, badge, tag, or PDA, and can be transmitted wirelessly to others via infrared or wireless internet. In Europe, users can embed ID chips into their mobile phones to authenticate their transactions. Thus they can buy services and goods via phone, or, combined with infrared or Bluetooth wireless technology, do transactions in-person with the merchant. Also cellular phone providers can pinpoint their subscribers within their service area, and companies are investigating how to provide geolocation services for identified users [68] [160]. Face and person recognition, usually based entirely on computer vision, is an active area, but requires significant processing power and need considerable robustness for a dynamic, mobile context. Despite these challenges, research in wearable computing face and person recognizers include [227] and [187]

There are also computer vision methods for indoor positioning. Aoki et al [6] use a computer vision solution for indoor position awareness, relying on colour histogram footprint for each location of interest. An omnidirectional video camera can grab a 360 degree snapshot to characterize map location [207]. Clarkson et al [43] construct a position signature from low-grade camera and audio.

Outdoor location is often derived using GPS or cellular networks to provide location-specific services. Automated tour guides around a city or campus are common research cases [177] [71] [258] [38]. Burrell and Gay [31] and Persson et al [189] use local area wireless networks to position and situate individual notes based on location. Physical orientation as well as outdoor location can be embedded within digital imagery [35] and zoological field study records [180], providing additional context over timestamps on everyday video cameras. A company, "Information in Place," offers location-based software and services for handhelds and augmented reality for the U.S. Coast Guard, museums, outdoor learning events, and a jazz festival [100].

Ways to sense activity include monitoring the user's own vital signs, and environmental sensors and computer vision recognition. Personal body statistics, such as galvanic skin response, heart rate, blood pressure, etc. can help measure a user's physical activity, and medical status. For instance, a wearable ECG monitor like in [154] could sense if the patient is in danger and automatically alert doctors. Picard and Healey [190] use body statistics to help computers become aware of their users' emotional state and adjust user interaction in response to mood. An example of this is the StartleCam, which takes memorable photos when the user is surprised [191]. Gesture and physical activity recognition through vision and sensors is being pursued in work like [185] [212]. Starner et al [228] demonstrate activity recognition on a wearable platform, where in one case, a user's gestures are analyzed for sign language, and another example uses gestures to interface with an augmented reality game.

While not as "exotic" as location, identity, and activity, context awareness entirely based on time is already a mainstream application. Calendar-based reminders are a heavily used and expected feature in desktop, PDA, and mobile phone based personal information management systems (e.g. Microsoft Outlook, Palm Desktop), particularly for the enterprise. And calendars and time give context for note-taking. PDA calendars like those offered on Palm OS and Microsoft PocketPC platforms allow users to jot notes in their calendars. Dey et al [65] use schedules to enable multiple users to take notes, share their comments, and identify general areas of interest during a conference. Stifelman et al [229] map time and audio with physical pen strokes to create an enhanced notepad that lets users navigate through notes in time or in space (touching parts of the transcript recalls the audio at the time the touched part was written down). Dey and Abowd [64] and Pascoe [180] combine time, position and activity information to create timely reminders and notes.

2.5 Wearable Context Aware Augmented Reality

This chapter gives a general survey of current research in wearable computing and augmented reality. Considerable applications exist in these domains, some commercial, but most still in the research phase. Besides hardware platform issues, the outstanding research areas include context awareness and registration.

A major theme in wearable computing is the user experience, often depicted as a visually enhanced viewport of the world as the user moves around. Context awareness helps identify what characteristics of the current situation, such as location, identity, activity, and time, are important to the user's current task. The wearable computer system identifies the user's current task and finds the appropriate information for the current context. A visually registered augmented reality overlay is presented to the user's head-mounted display, showing the current relevant cues and information based on context.

The big problems in these domain are twofold: hardware challenges and interface challenges. Technological advances in the computing industry, driven by the demand for general purpose computing devices, will address issues like power consumption, display, processing speed, etc. However, to realize the dream of an environmentally aware and seamless computer-enhanced interface, researchers still must achieve fast and accurate augmented reality registration, and rich context awareness. My research in the subsequent chapters explores registration and context awareness to provide collaborative and informative interfaces for wearable computers enabled with augmented reality.

Chapter 3

System Fundamentals

To conduct the exploration into wearable augmented reality as proposed in the first chapter of this work, a testbed platform needs to be defined. This chapter describes the base hardware and software used as a wearable computer testbed in the later chapters. Motivating the hardware and software are a general architecture and target platform, which are described.

3.1 System Architecture

The general data flow encompassing the hardware and software for the wearable augmented reality system is presented in Figure 3.1.

System inputs include a video stream from the worn camera, audio from the microphone, user input from some input device (e.g. a mouse, a miniature hand-held keyboard), and external data transmitted from the network via the wireless modem. System outputs include graphics (that form the virtual overlay), audio feedback, and data to be transmitted back to the network via wireless modem. The key modules of the system are the *wearable computer framework*, *the image registration module*, and *the augmented reality virtual overlay module.*

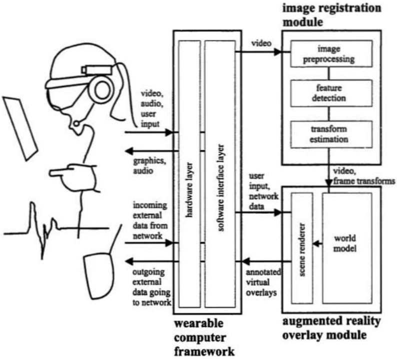


Figure 3.1: General Data Flow in the Wearable Augmented Reality Testbed

The *wearable computer framework* processes all system inputs from the real-world and outputs to the user. This framework embodies the actual hardware of the wearable computer and the software that directly interfaces with the hardware. The framework acts as a convenient foundation to prototype and test the actual algorithms to be used for the registration research problem.

The *image registration module* represents the core of the research, as its purpose is to perform image registration. Taking video from the wearable computer framework module, the image registration module must compute the registration parameters of the video stream in real-time. Three basic steps need to be performed in this module: each video frame is image processed (e.g. to remove noise, to enhance the image for subsequent analysis), distinct features in the frame are identified, and the transformations necessary to register with the video stream are calculated.

The *augmented reality overlay module* presents the actual results of the real-time registration to the user. It is responsible for constructing the virtual overlay by combining user inputs and external data sent from the network through the use of the results generated by the image registration module. User input is needed to allow the user to identify objects to register and to customize the format of presentation. External data from the network allow other users to present and overlay additional information on the virtual overlay, making the wearable computer system a collaborative environment. The image registration module provides the information needed to register the user inputs and external data with respect to the video camera's view of the real world environment. A scene modeler submodule merges the information together, making use of additional data stored about the parameters of the environment via a world model, which updates itself from the image registration output.

3.2 Target Platform

The wearable computer framework consists of a hardware layer, which defines the wearable computer itself, and a software interface layer, which interfaces the hardware with the image registration and virtual overlay modules. Development of the wearable computer framework, specifically the software interface layer, is one of the first tasks in the research, because it provides the foundation for prototyping the other two modules in the architecture.

The first step taken in developing the wearable computer framework was to specify the target platform. The target platform is the vision of the final system's hardware and software at a general level. The target platform was envisaged to consist of:

- see through display headset
- miniature microphone and earphone
- miniature computer
- wearable video camera device
- wireless network connection

With regards to software, the desired underlying operating system running on the target platform was chosen to be Microsoft Windows 95/98. The chief reason for selecting Windows 95/98 is strong prevalence of Windows 95/98 based computers, including those available to the author. Windows 95/98 based laptops dominate in the general computer market, and development of smaller and faster miniature processors for these machines is still active. Also, there is an abundance of programming tools for Windows 95/98 (e.g. Microsoft's Visual C++, Borland C++, etc.), and the availability of free application programming interfaces (APIs) for fast graphics, video, and networking (provided by the Microsoft Direct X and Video for Windows software development kits [156]). The end result is a software framework that would be commercially viable, with support on a fast miniature platform such as wearable computer, and supported by a rich programming environment. Note that Windows NT provides similar advantages to Windows 95, with the additional benefit of better stability. However, Windows NT is not as prevalent as Windows 95 in the general market, and support is lacking for the current version of Direct X.

Alternative operating system environments were also considered. The MIT Wearables Group (and a good portion of the academic wearable computer community) do most of their work using the Linux operating system [87] [176]. Linux is more compact, efficient, flexible, and robust than Windows offerings, and has a sizeable user base in the wearable computer community. However, support for the latest hardware devices, such as cameras, is not guaranteed by the manufacturer (if not available, the driver would have to be custom written, or sought out from the wearable community), and Windows has a greater availability of standardized, supported development kits - which is desirable for rapid prototyping. Also, Linux is not as widespread as Windows in the general market, limiting the audience for commercial applications.

Another option was to forego a specific operating system and use the Java programming language [157] since Java has the promise of "write-once, run everywhere", i.e. platform independence. It also has networking and some multimedia capabilities built in the language itself. However, Java is currently limited in terms of speed when compared to platform specific solutions. Although Java is touted for portable communications devices, only a handful of such computing devices are actually available in the commercial market. In the long term, when Java has matured sufficiently, it may be wise to invest in a Java based solution, but this is beyond the scope of this research.

Instead of a general purpose operating system or scheme like Windows, Linux, or Java, a specialized environment could be used. Environments like Windows CE [50] exist, tailored for portable communication devices like PDAs. Real-time operating systems specialized for industrial, embedded applications like Qnx [141] may be appropriate. However, such environments lack the diversity of tools for a general wearable computer, often require proprietary (i.e. costly) development tools, are limited in popularity (i.e. the actual market for such environments is very specific), and may lack the flexibility or processor power found in more general purpose environments (e.g. Windows CE lacks real-time video support and is currently based on processors far less capable than current generation Pentiums).

3.3 Hardware

Off-the-shelf, commercially available hardware was chosen for this research. This is to focus work on rapid prototyping of software and interfaces. The hardware pieces selected for the wearable testbed are listed as follows (as shown in Figure 3.2):

see through display headset : I-Glasses

Miniature LCD technology has advanced considerably and eyeglasses displays like [49] have been promised for general purchase for years. However, there is still only a handful of see-through displays available, at an affordable price for a researcher. I-Glasses [97], a 320x240 colour see-through display offered the best trade-off in terms of resolution, colour, and price, compared to displays like the TekGear M-1 and M-2 [236] and the Sony Glasstron [195].

wearable video camera device : PC-53XS Camera, QuickClip USB

Numerous CCD and CMOS cameras are available, notably as low-cost webcams. Their bulky housing and integrated circuit boards make them difficult to repackage in a wearable form factor (i.e. nonobtrusive and comfortable enough to be worn on the head) without time and expertise. The Supercircuits PC-53XS CMOS Camera was chosen because of its complete coin-sized form factor and affordable price [230]. The entire camera array can be mounted inobstrusively, with external wires for power

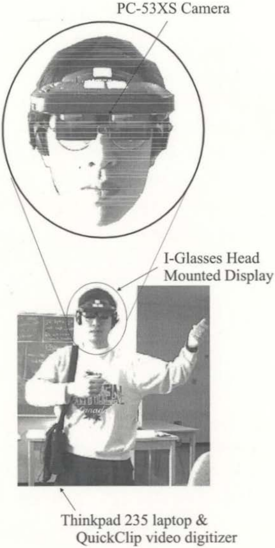

PC-53XS Camera
I-Glasses Head Mounted Display
Thinkpad 235 laptop & QuickClip video digitizer

Figure 3.2: Wearable Augmented Reality Testbed Hardware Components

and NTSC video out. NTSC video was connected to a Connectix (now Logitech) QuickClip video digitizer [140] that feeds a computer's USB port.

miniature computer : IBM Thinkpad 235

Reference designs for wearable computers exist online, such as [86]. With the choice of Windows 95/98, and considering time, simplicity of development, focus on software rather than hardware work, a subnotebook computer was chosen. The IBM Thinkpad 235 (also known as the Ricoh Magio) offered a complete Pentium 233 multimedia computer in a very lightweight and compact package [242] (2.75 lbs, 9.2 by 6.8 by 1.3 inches). Its battery-life, however, was limited to about 1.5 hours using the standard batteries.

Audio can be supplied directly from the Magio's sound ports and built-in sound. Networking is supported by a PCMCIA network card, although this was tethered to a long RJ-45 line. Very late in the research, 802.11 wireless networking was made available.

3.4 MCLGallery : Software Interface Layer

The chief goal of the software interface layer was to facilitate rapid prototyping of research results under Windows 95/98, and to offer a wide variety of services for the multimedia communications activities, while still harnessing the best capabilities available to the operating system. The resulting product was a C++ class library called MCLGallery.

3.4.1 Motivation and Capabilities

MCLGallery (where "MCL" stands for Multimedia Communications Laboratory), was developed by me as a suite of classes built on top of several powerful Microsoft

Windows multimedia APIs. Designed to be used in a rapid application development (RAD) programming environment (in this case, using Powersoft's Power++, formerly known as Optima++, RAD C++ compiler), MCLGallery attempts to hide the complexity of the Windows APIs, and provided simplified but powerful programming interfaces and customizable components. The RAD programming environment helped development further, by providing a graphical means of constructing user interfaces and placing customizable component controls built from MCLGallery classes. MCLGallery made use of currently available (and free) programming APIs that are standards for graphics, networking, and multimedia, namely Microsoft Direct X (providing graphics and networking) and Video for Windows (providing video support).

MCLGallery also addressed the need for a basic software framework for other projects developed in my research lab. The goal of rapid prototyping to final implementation is desirable because demonstration projects can be presented and updated (via supervisor and industrial partner feedback) on an ongoing basis. Hiding or abstracting the complexities of Windows programming means researchers using MCLGallery can concentrate on coding in C++ for areas like face tracking, telewriting, augmented reality, telemedicine, face recognition, and image coding. Although the range of research interests in my group was diverse, there were characteristics common in my work and the rest, which MCLGallery supported, e.g.:

- Manipulate colour and greyscale images
- Capture images from live video
- Transmit data across a network or via modem
- Perform all operations approaching real-time
- Customize the graphical user interface to the needs of the application

The key capabilities provided by MCLGallery, which address what is demanded by the system architecture diagram in Figure 3.1, are:

- real-time video capture
- image processing
- image storage and retrieval
- audio capture and playback
- data transmission and reception via networking (on phone, serial link, TCP/IP, or IPX)
- vector and bitmap graphics rendering

3.4.2 Image Processing and Computer Vision Frameworks

My research group's work, as well as my own, tie closely with the domains of image processing and computer vision. Frameworks in these fields can be classified as programming libraries, visual tools, and component libraries.

Programming libraries conveniently provide prepackaged routines for the researcher. But one problem is to find the appropriate library that can meet the needs of the application, if any [44]. There are also issues of proprietary code and licensing (this issue is also common in the other approaches as well). Non-proprietary public domain libraries do exist, such as CVIPtools [262], and Vista [192]. But these libraries are not suitable, as many target non-Windows platforms like Unix (at the time of MCLGallery's initial development), do not support live video capture or networking, and are oriented at pure research and courseware as opposed to rapid prototyping and product development.

Another general problem with pure programming libraries is the learning curve to fully exploit the capabilities of the library. Too many functions and classes could overwhelm the researcher. A graphical user interface (GUI) "shell" to contain all the library's capabilities, and extensive documentation may remedy this (such as the GUIs and Unix "man" pages found in CVIPtools and Vista). However, GUI shells fix the researcher to a specific user interface, and cannot be readily customized to the needs of a final product, and the sheer size of documentation itself could become intimidating to the reader. Matlab [102] features a powerful shell using a proprietary language, with toolkits for GUI design and C language exporting, but has a restricted licensing and an enterprise-level price model.

Often built on top of programming libraries, visual design tools reduce the researcher's load in learning a library, and use visual icons and visual interconnections to formulate algorithms. Examples of such systems are Ad Oculos [98] and the Cobra/WiT package [44]. Although this is a very powerful medium to develop research algorithms, these systems are often commercial and require licensing, or they are locked to specific image hardware configurations (e.g. Cobra/WiT). Finally, these systems often do not produce actual source code for a standalone application with a custom user interface.

Imaging component libraries have seen great growth recently [136]. Unlike programming libraries, software components can be plugged into an application with minimal linking and compilation effort, and often feature visual controls to help set their parameters through Rapid Application Development (RAD) environments like Microsoft Visual Basic and Borland C++ Builder. An example of a powerful component library for video and imaging is Matrox's Active MiL library [136], although this library was specifically for Matrox hardware at the time of MCLGallery's development (and is still largely biased to Matrox equipment today). Despite the benefits in code reuse and application development times from components [259], developers fear the inability to modify components to fit them to the precise needs of their applications [116].

MCLGallery attempts to harness the strengths of each of these three framework classifications, by providing three levels of programming interface to the researcher. These are bound together in an integrated programming environment. The three levels are: the user interface level, the object level, and the data level. At the user interface level, the researcher prototypes a graphical user interface and deploys MCLGallery components with a RAD tool. At the object level, the researcher programs in C++ with a class library, featuring abstractions of bitmaps, video streams, matrices, etc. At the data level, the researcher manipulates any multimedia information provided by the MCLGallery classes at the byte scale (e.g. for data compression or encryption).

3.4.3 Components and Component Templates

MCLGallery features specialized imaging and networking component templates to encourage rapid development of prototypes. These templates take advantage of the chosen development environment's features, for better integration with the environment and ease of use. Before going to the specifics of MCLGallery component templates, this section explains the development environment and its component template feature.

For work on the user interface level, Powersoft's Power++ was chosen for use with MCLGallery. It has a strong optimizing C++ compiler (based on Watcom C++), and a comprehensive, easy to use RAD environment. Although this restricts development to a specific programming environment, the main goal of MCLGallery is not to be a universal cross-platform, cross-compiler solution. Rather, MCLGallery is to meet the immediate needs of the lab researchers by harnessing the programming environment's specific strengths. The specific strengths of Power++ are enhanced drag-and-drop programming and easy authoring of collections of components, which are known as component templates.

In most RAD environments, such as Visual Basic and Borland C++ Builder, drag-and-drop programming involves the programmer placing a component visually, and setting its general properties via an object inspector window. But the actual underlying code to operate the component is just a bare skeleton for the programmer to fill in. This requires some a priori knowledge of the component application programming interface (API) used by the RAD environment. Power++ overcomes this by generating the underlying code automatically through visual associations between components. The programmer drags a visual component and drops it into the actual source code of another component that will be calling the dropped component. Power++ then shows a list of C++ methods that the programmer can perform with the dropped component, as well as local variables the programmer can use. By selecting a few choices from a list, the underlying code is automatically inserted. Unlike using a "wizard", any combination of components can be dragged and dropped in any order into the source code. Hence, this paradigm is somewhat analogous to having a context sensitive class reference guide available at all times, and reduces the learning time needed to write user interface source code.

Component authoring allows the rapid prototyping, creation, and extension of visual controls available to MCLGallery users. Although component authoring is not unique to Power++, component template authoring is. Components are hard to customize without extensive reprogramming [116]. Also creating components still requires some knowledge of the component API used by the programming environment. For instance, writing an ActiveX component requires some knowledge of the complex Microsoft COM (Component Object Model). Component templates address both of these problems, by working with a collection of components.

Component templates are similar to the idea of software frameworks, as mentioned in [116]. They are essentially a collection of visual components with working underlying source code. Effective component templates can be dropped into any programming project, and without any additional coding, they can run immediately. Component templates combine the strengths of programming libraries, visual tools, components, and the paradigm of programming by example. They harness the capabilities of an underlying programming library via the prewritten source code, as opposed to requiring the programmer to fill in all the underlying code. Component templates use the idea of visual association from visual tools, by presenting a collection of related visual components in the workspace. They can be integrated into a program easily by a mouse click, just like a visual component in a RAD tool.

Component template authoring involves selecting a group of components and then telling the programming environment to create a component template from this group. The selected components and all of their underlying source code form the component template, and the template is added to a common template palette in the programming environment for others to use. To use a component template, the programmer selects the appropriate template from the palette, and clicks the mouse to drop it into the project. The program can run immediately with the template as-is, or the researcher can modify the underlying source code like any other part of the program to meet the needs of his or her project. Thus, using component templates is similar to "copying and pasting" source code.

3.4.4 MCLGallery Component Templates

MCLGallery provides component templates for manipulating images, networking, capturing video, and drawing line graphs. Each component template embodies several components and has working source code for common research tasks such as loading and saving a bitmap, running a video, or performing matrix operations on an image in real time. These visual controls can be selected at design time by the researcher, and their internals can be customized to meet specific application needs. Note that component templates are not related to C++ templates or the Standard Template Library. Instead of being strict language constructs, component templates are a mixed collection of controls with associated source code that are easily copied and pasted.

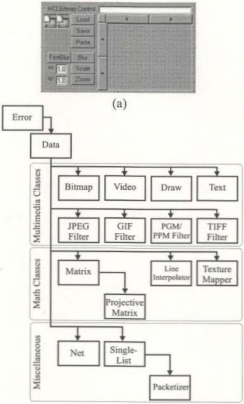
Through the RAD environment, the researcher can rearrange or modify (internally or visually) the components according to the needs of the application. MCLGallery component templates ease the learning curve on the underlying programming library. Because the source code behind a component template is immediately accessible and fully commented, the researcher can learn how to use specific classes and methods by example. Commented code is also provided to demonstrate how component templates can be interconnected (e.g. connecting video to process bitmaps). This can build up practice in using the programming library and can lead to writing source code from scratch.

For example, one of MCLGallery's component templates is the MCLBitmap Control (see Figure 3.3 (a)). This control allows a researcher to load, save, copy and paste, scale, scroll, and zoom images from a variety of formats (.BMP, .GIF, .JPG, .PGM, and .PPM). Also, it features two examples of how to perform a typical image processing operation (blurring) on the loaded image - a simple method using matrices, and a faster method using pointers and a linear frame buffer. All the source code necessary for these tasks can be immediately viewed and modified if the researcher desires it. Thus, the blurring operations could be changed to another algorithm, like edge detection, for instance.

3.4.5 The MCLGallery Class Library

Underlying the components and component templates is a class library (whose basic hierarchy is shown in Figure 3.3 (b)) that comprises the object level of MCLGallery. Classes include bitmap, video, text, networking, line drawing, and matrix math. The researcher can use the classes through component templates or by direct programming. These classes make use of, but hide the complexities of several powerful (yet free) multimedia APIs, namely Microsoft's Direct X and Video for Windows [156], and SciTech's MGL [213].

A standard for games, Direct X is used by MCLGallery to provide networking support (via Microsoft's DirectPlay API). Video for Windows is a widely supported API for Windows video devices. MGL is a powerful API for graphics, used in many high-speed games, and interoperates with Direct X, while offering an easier low-level programming interface than Microsoft's DirectDraw. MCLGallery's internal routines access these APIs to provide fast image and video operations. MCLGallery's class library is programmed using the Win32 API, and thus could be ported to another Windows compatible compiling environment if necessary. Implemented as a DLL,



(a)

(b)

Figure 3.3: MCLGallery Components and Class Hierarchy

(a) The MCLBitmap Component Template, (b) MCLGallery Class Hierarchy (class names have common prefix "MCL"; arrows point from parent classes point to derived classes)

MCLGallery can be easily updated without recompilation of applications.

In the data level of MCLGallery, multimedia classes (e.g. bitmaps) can be broken down into byte buffers, which can be used for transmission across a network. Implemented through Direct X, MCLGallery networking can be done via TCP/IP, IPX, or modem connection. The data level is represented as base byte buffer class, acting as the parent to the MCLGallery multimedia classes. Besides easing network transmission, a base data class providing byte buffer access helps the development of multimedia classes for reading and writing different image formats (e.g. JPEG, GIF).

Underneath all of the classes of MCLGallery is the MCLError base class. MCLError addresses the difficult task of debugging real-time video/image programs in Windows. It features the ability to log messages to file (which remains intact even after the typical Windows catastrophic crash), timing sections of code, and sending messages to a "console window" independent of the Power++ programming environment. The programmer can write console messages for his or her personal reference. Since they are derived from MCLError, all MCLGallery classes output diagnostic messages to the console window when they encounter problems, such as invalid input parameters to class methods. In addition, all MCLGallery classes have a common method, "GetInfo()", which displays the current status and general contents of the class on the console window. This method can help the programmer assess the current state of an object.

It should be noted that the classes in MCLGallery do not provide a comprehensive library for image processing or computer vision. Rather, they provide a baseline framework for researchers to build image processing and computer vision routines (or port existing algorithms). Such image processing and computer vision routines could be built through component templates (which would be extended and exchanged by researchers) as opposed to modifying the actual class library itself.

3.4.6 Recent Developments

Outside this thesis, MCLGallery has been used in a number of graduate and senior undergraduate student projects in the Multimedia Communications Lab at Memorial University. Examples of work done with MCLGallery include: telemedicine imaging across phone lines, an eye tracking study, a telewriting system, image compositing and mosaicing, image enhancement for the visually impaired, colour based face tracking, face warping and error reduction, a card based storytelling demo, a desk enhanced with projections over sheet documents, background reconstruction research using image differences, a telemedicine patient record system, and a networked sketch based image coding demo.

Since its initial development in late December 1996, MCLGallery continued to evolve during the development of this thesis. With each new revision of Microsoft's Direct X, MCLGallery was updated to exploit new functionality, notably in DirectPlay, which provides networking support. Upon SciTech's MGL library public domain release [213], MCLGallery's image blitting routines were changed to use MGL's more efficient and easier-to-interface routines. MCLGallery added support for the Intel Image Processing library, which is speed optimized for Pentium processors on Windows platforms [105].

Other publicly available software toolkits have since appeared, or appeared around the time of MCLGallery's development. These toolkits provide support for augmented reality, computer vision and image processing, or context-awareness on wearable computing platforms. For example, Sulawesi and Context Toolkit are Java-based frameworks for context awareness [168] [210]. The OpenCV Computer Vision Library offers a suite of Pentium optimized computer vision related functions [106]. Microsoft Research has a Vision Software Development Kit [201]. ARToolkit provides a platform for augmented reality using Open GL 3-D graphics and fiducial registration [26]. While these recent toolkits offer a greater depth of features in their specialty area (e.g. OpenCV for computer vision, ARToolkit for AR), MCLGallery offers classes in a broad range of areas in a single environment. Thus, a developer would have to incorporate separate image management, networking, and computer vision libraries from diverse sources to build a complete networkable application. MCLGallery has classes for all of these and more in the same place, under a common level of abstraction.

Chapter 4

Personal Context

As described in Chapter 2, wearable computers employing augmented reality (AR) allow their users to move freely in the environment and interact with virtual information associated with real world objects. Augmentation can gather and act upon sensor readings of the wearable computer user's environment and personal activity, thus building an awareness of the user's context. Context-awareness is an important feature for wearable computer user interfaces, where computer interaction is enabled only when relevant to the current task [228].

Context-aware wearable applications presenting visual displays can use AR registration to anchor virtual information with respect to the user's surroundings. Rather than have information moving with the user, such applications have information spatially organized according to the environment. This enables the whole scene as a canvas for user interaction. For instance, [150] uses walls throughout physical space to distribute a multitude of terminal windows.

From the point of view of a wearable's display overlay, the user's body parts seem no different than any other object in the environment as potential AR registration targets. But unlike any arbitrary object, they are under the user's intimate control, and can always be assumed to be available to the wearable. This chapter explores the notion of using the user's body parts as anchors for virtual information, i.e. personal context, and exploits their distinction from other objects in the environment. Also, I describe the two working personal context proof-of-concept systems that I built (HANDEL and Footprint).

4.1 Defining Personal Context

Sensing when it is relevant to enable interaction can be a challenge. This typically involves analysis of the user's overall environment. Context sensing faces more complications as the wearable computer user moves around, varying the amount of available environmental computing infrastructure to enable sensing (e.g. no wireless networking coverage, going indoors with no GPS information, no "smart room" instrumented with beacons and sensors, etc). But even in the complete absence of environmental support, there remains one physical object available to mediate interaction: the user's own body. To exploit this to enhance a wearable computer user's experience, I introduce "personal context", first as a theoretical thought experiment in the section below, and then as a concrete definition in the subsequent section.

4.1.1 Human-Attentive Computer Interaction and Personal Context Interaction

This section presents a thought experiment regarding a subsection of human-computer interaction, namely human interaction with attentive computer interfaces. This sets a framework where I situate context awareness for a wearable augmented reality system, and from which personal context can be derived. This process sets a theoretical motivation for personal context, and establishes its relationship in the hierarchy of human-computer interaction.

As described in Chapter 2's survey, recent research in wearable and ubiquitous computing have focused on a new paradigm of human-computer interaction. Whereas traditional desktop interfaces demand conscious user attention and interaction with the computer to accomplish tasks, wearable and ubiquitous computing delegates the computer to the periphery, be it somewhere on the body or in the environment. Instead of graphical interfaces immersing the user on a desktop, the user is free to conduct tasks in the real world, while being alerted only occasionally of relevant information by the computer in the periphery. Relevancy can be determined from the user's conscious or subsconscious needs and interests. Such a system is said to be "attentive" [225]. Desktop human-computer interaction requires a two-way exchange of attention between the user and computer. An attentive computer in the periphery has a flow of attention largely biased away from the computer towards the user. For the purposes of this thesis, I would like to focus specifically on the presentation of information to the user in an attentive computer system.

An attentive system approach allows the user to focus on other tasks at hand, and is very suitable for a wearable augmented reality computer interface. Context awareness and augmented reality now become prominent tools in this attentive interface. Because of minimal flow of attention from the user to the computer, an attentive system requires circumstantial data to deduce the user's need for information. Context awareness uses sensors like GPS, clocks, and video, as well as knowledge provided consciously (e.g. a personal profile), or subconsciously (e.g. past usage history and calendaring information), by the user to gather inputs, and correlates them together to deduce a user's need. These data sources can focus on the user, or can be pulled from other sources like online servers, environmental infrastructure, etc. Augmented reality renders the messages from the computer to the user, incorporating virtual information into the user's real world view. Augmented reality is more than a graphics driver, however. Its use of registration allows association of information with objects in the scene, strengthening the sense of information relevancy to the user.

Even with powerful tools like context awareness and augmented reality, the attentive system faces a number of challenges. The major danger is information overload: the attentive system can drive the user back into a desktop-like two-way exchange of attention, if too much information is presented. Augmented reality only magnifies this problem, by allowing excess information to be registered with everything in the real world, much like excess litter on the street. While context awareness filters incoming information, it too can cause problems, where useful information could be accumulated into an unmanageable pile of data over long periods of time. Allowing the computer to diminish reality selectively and augment only the desirable aspects (i.e. mediated reality), as pioneered by Mann [151], suggests a possible solution. This begs the question: how does the computer know what to diminish and what to augment? Directing the computer manually leads to an attention-demanding user interface. Errors made over diminishing and augmenting aspects could result in a "deceptive" system, replacing the "information overloading" problem with an "information misleading" problem.

To deal with this overload problem, the attentive system requires deeper knowledge of the user's immediate task and long term goals. Once tasks are identified and prioritized, the system can choose what information to present at what time, and what information to retire when no longer needed. Longer term goals could be determined from a user's profile information, or analysis of past history data.

A wearable augmented reality system has a difficult time identifying the user's immediate task, where the user is interacting with the real world, and the computer system has limited access to the user's intentions about the task, despite a live feed on all real world information.

A context awareness approach can be employed, taking sensor data from the immediate surroundings around the user to deduce the immediate task. Wearable systems complicate the data gathering because the mobile user could be moving from place to place, performing an unpredictable set of tasks in any possible environment. Despite this problem, there are notable characteristics of mobile human activity. They include a guarantee of the wearable computer user's presence and a body language for physical activity.

A user's presence is guaranteed in a wearable system: since by definition the wearable is attached to the user's body. This differs from instrumented rooms and vehicles, where users can come and go in those spaces. This simple assumption suggests that an attentive interface should be specially tuned to its user's characteristics, such as physical and mental (e.g. personal preferences) parameters. Physical parameters are easier to obtain in a wearable system, where sensors can gather body and environmental status.

The user's task in the real world is accompanied by some body language: often manifesting as some body part interacting with a real world, along with some kind of focus of the user's attention. The focus of user attention in the environment is a cue to an important event, and combined with a body gesture, produces a possible task of interest for the attentive system to augment. This does not encompass all possible tasks (e.g. thinking, daydreaming), but suggests an interesting set of applications for study.

In a general situation where a mobile, wearable augmented reality user is in constant motion and activity in changing environments, the attentive interface has the user's presence and the user's body language as constant data sources. It is here I propose the use for personal context. While context awareness and augmented reality cover a wide range of possible environmental and personal configurations, a subclass of attentive interfaces can simply focus on personal presence and body language.

With regards to contextual information, personal presence information can be gathered by observing the immediate environment around the user, such as using a head mounted video camera. Body language in the context of a task could be restricted to actions involving a focus cue (i.e. *looking at* an object), accompanied by a personal physical gesture. These can be derived by head motion and visual focus from a head mounted camera, and body gesture detection given a priori task information.

With regards to information presentation, personal presence and body cues suggest a convenient "last resort" rendering surface. Instead of registering objects that the attentive system may know nothing about, the system could choose to render on the user's body parts relevant to the immediate task.

Given this foundation of human-computer interaction, I propose the notion of personal context in the next section.

4.1.2 Definition and Scope

Personal context is the contextual awareness of the user's own body - as a stimulus and rendering surface for augmentation and mediation. While a general context can be derived from the environment, a personal context can be derived from an awareness of the user's own body parts with respect to the task at hand (e.g. recognizing a physical procedure from natural hand gestures). Plus, the user's tasks often center around the active body parts, which suggest a natural focus for any virtual information presented (e.g. showing instructions near the hands in a manual task).

Thus a user-centered wearable computer system can always rely on the presence of the user body. Direct sensor measurements or a combination of sensors and pattern recognition can derive personal context from the user's body. Then virtual information can augment the user's first-person experience through a heads-up display, audio, projection, etc., but only enabled (or mediated) by the relevance to the user's task at hand.

For this thesis, interaction is restricted to the hands and feet. Focusing on the hands and feet can be applied to many other hand and feet oriented physical tasks: physical rehabilitation and therapy, choreography, mapping and pathfinding, sports training (e.g. martial arts, tennis, soccer, etc). Also virtual annotations and commands can be defined, moved, sized, stuck, or kicked onto real world objects by hand and feet gestures. For example, framing a shot for video or a photo can be triggered by a two-handed "frame" gesture, where the size and location of the framing gesture defines the parameters of the snapshot (this can also define the placement of a virtual annotation window in 3-D, like in Mann's reality window manager [150]).

It should be noted that actions with hands and feet need the context of the current task and environment. For instance, the hand alone does not suggest output to a computer. Meanwhile in the real world, the hand does act as a context for output for a wide variety of computer and non-computer based information. People use their hands to hold and control personal data assistants (PDAs), as well as paper notepads to view email, addresses, phone numbers, etc. Close to the hand, the wristwatch also provides another data display surface and control interface. Also, people sometimes tie strings on their fingers as reminders, and write directly on their hands to quickly jot down a phone number or the answers to the weekly quiz. The hands even act as a visual medium for entertainment applications such as finger puppets and casting shadows of animals and other creatures. In each of these cases, the hands provide a convenient interface : they require little or no add-on hardware, they can be used covertly, and are available to display information on an as-needed basis.

4.1.3 Distinguishing Personal Context from Other Attentive Interfaces

Given the definition and framework in the previous sections, I now discuss how personal context compares with earlier approaches to other information management schemes related to wearable augmented reality, like context awareness. The discussion emphasizes personal context's distinctive features in comparison to other work.

Context awareness covers a greater span of contexts than personal context. As detailed in Chapter 2, context awareness uses time, identity, place, and activity as triggers and distinctive signatures for wearable interaction [180]. Affective computing [190] drives a computer's response to a user's emotional state, which may, or may not be derived from physical activity. In contrast, personal context centers around the activity of the user's body parts only.

Augmented reality systems also overlay virtual information onto the real world, including first-person applications using head-mounted displays and environmental sensor cues to register the information onto appropriate objects to help direct a user in a task, like servicing a printer in [72] or reading an enhanced book [25]. Personal context is a niche augmented reality application, relying entirely on the user's body parts' interaction with objects and the environment to trigger virtual overlays. So a personal context approach to a printer servicing application or an augmented book would rely on the user's gaze with respect to the hands, rather than building an ultrasonic tracking infrastructure, as in [72], or using specially marked book pages, as in [25].

Gaze detection and eye tracking systems in research and the commercial arena share personal context's interest in finding the user's focus to enhance user interfaces. Unlike a wearable AR system powering a personal context application, such systems have their sensors and cameras at fixed locations. The sensor array is typically around a large monitor to enable control of a desktop application and assumes a user sitting in place staring straight at the screen. The work of Selker et al [215] is an exception, using emitters and imaging sensors embedded in glasses to measure eye gaze from infrared beam reflection off the eye. While this solution could be an interesting platform for accurate gaze tracking in future research, the HANDEL and Footprint systems presented here accomplish user focus detection with a single off-the-shelf camera.

To combat virtual overlay clutter on an augmented reality display, Julier et al [112] introduce "information filtering." This idea is a knowledge-driven means to cull out irrelevant and unnecessary information and highlight relevant and prioritized information for AR displays. Information filtering uses a weighting function constructed from measuring different criteria based on a priori specifications of the augmented reality task. While reported to be effective, this scheme requires a very detailed task and scene object analysis, and function tuning before actual operation of the system. Mann and Fung's Reality Mediator [152] also seeks to eliminate clutter, but not only the virtual. The Reality Mediator erases or replaces undesirable information in the environment with virtual overlays of the user's choosing. Personal context shares the goal of minimizing virtual overlays by concentrating only on the user's body parts as the main foci and triggers for interaction rather than the entire environment. However, personal context's emphasis on body parts gives a more limited source of analysis than taking the entire task and environment in hand for information filtering and Reality Mediator's notion of "diminished reality". On the other hand, personal context has to know less about the task and the users' high level intentions. Thus a personal context has greater flexibility across different tasks and domains (albeit with less in-depth awareness about the tasks and intentions), and can deal with new, unpredictable situations.

Personal context also relies on the user's body as a rendering surface. This does not imply a body-stabilized interface (like a cylindrical or spherical overlay surrounding the user in [21]), but rather an object-centric interface, where the objects are really parts of the user's body, which appears world-stabilized to the user. Unlike a true world-stabilized interface as described in [21], overlays are attached to body parts with little or no attempt to assess a complete world model. So overlay graphics may be attached to the user's hands, but tracking can be done using simple 2-D techniques, with no knowledge of user's physical location, head orientation, etc. Although a complete world model is desirable, a simplified model makes available simple, fast, and perhaps robust, algorithms for tracking.

"Perceptual intelligence" [186] and "visual context" [61] have a broader scope than personal context's single user's experience (e.g. "smart rooms" monitoring and responding to human activity [107]) and can focus on environmental or user-based pattern recognition. Pentland [186] identifies new opportunities for visual contextual analysis from a first-person perspective. Specific applications include a sign-language recognizer [228] and an aid for billiards [111]. Although both cases employ body-mounted sensing to track body parts (i.e. hands gesturing or holding objects), the former lacks any augmented reality overlay and the latter also depends on some environmental awareness. While personal context is interested in detecting body parts and recognizing gestures, it requires a minimal flow of attention from the user to the computer. Although the computer may be hidden in a room or on the body itself, systems like the sign-language recognition and body-based user interfaces that control a projected display in a smartroom require the users to consciously spend attention on the computer interface, which is similar to the two-way exchange of attention on a desktop system (except keyboards and mice are replaced with hand gestures). Smart rooms can respond to subsconscious stimuli from its users, but the users must be physically present within the room, and very specific a priori environmental information is needed (e.g. the architecture of the room is used), and perhaps a priori information about the users as well (e.g. face templates, body measurements). With its user presence characteristic, Personal context cannot assume a fixed environmental model, and demands a priori information about the single user only (the wearer of the personal context system).

I created two systems, HANDEL and Footprint, as an initial investigation into personal context. Both systems infer the user's need for augmentation from personal context, in specific domains: piano playing and private ballroom dance practice, respectively.

4.2 HANDEL

HANDEL, a HAND based Enhancement for Learning piano music, is an example of personal context to assist learning. It uses the hands to trigger an augmented reality overlay onto the hands themselves in the context of piano playing, in essence creating a "hands-up" display.

4.2.1 Hand-Based Personal Context

Considerable research exists in hand-based user interfaces, as well as computer vision techniques used to locate and recognize hand and gestures, such as [186]. Vardy et al [244] explore using wrist-mounted camera to capture finger gestures input. Hardware such as Data Gloves, magnetic trackers, handheld keyboards, and optical sensors can be used to obtain hand pose, orientation, and data entry. However, in these cases, the hand acts solely as an input device. On the other hand, Krueger's work has some examples with interactive graphics merged with hands [129], and Miyasato places small displays on users' hands to ease interaction with a large screen virtual environment [158], allowing the user to "see through" the hands.

Piano teaching tools already exist in the marketplace, including self-help computer software showing keyboard and music layouts and electronic keyboards with lighted keys to guide pianists. Modern acoustic player pianos like the Disklavier allow direct playback on the keyboard from music files or from captured piano key action.

4.2.2 Design and Implementation

HANDEL attempts to help practicing pianists to memorize piano music. In HANDEL, the pianist, equipped with a wearable computer system, sits at a normal acoustic piano with no sheet music. As the pianist attempts to play a piece from memory, the pianist may look down at the hands. Focusing on the hands is the trigger for HANDEL to overlay music. Otherwise, the pianist sees nothing - no graphics clutter the practice session - and without sheet music, the pianist can concentrate on playing from memory, as if in a real recital. When the pianist looks at the right hand, only the right hand's part of the music is shown near the hand, at the current position in the piece, and similarly for the left hand. Thus, HANDEL uses the hand as an input - to trigger when or when not to overlay virtual sheet music to assist the pianist. Because the music is presented near the relevant hand, the hand also acts as context-sensitive display window for sheet music - i.e. presenting information only when needed by the pianist.

Running the wearable testbed described in Chapter 3, HANDEL uses the head mounted video camera to perform scene analysis, and overlays graphics on the see-through head mounted display. Thus, the pianist's hands are totally unencumbered and free to interact normally with the piano. HANDEL uses FFT phase correlation analysis (defined in Chapter 5 and [36]) on consecutive video frames to determine whether the pianist's head is looking to the left or to the right. This is used to assess whether pianist is looking at the right or left hand. A look-up table skin colour detection method is used to detect whether a hand is in view or not (the skin colour scheme is preset with a training set of skin colour beforehand). Skin colour detection is sufficient since it is assumed that the only thing, apart from hands, that the head mounted camera will see is the piano (a non-skin coloured object).

Figure 4.1(a) illustrates HANDEL's general system data flow. On the 233 MHz Pentium wearable platform as described in Chapter 3, HANDEL runs at about 5 frames per second, which is sufficient for slow piano playing.

The practice session begins with the pianist loading the music score into the HANDEL program. In the current implementation, a simple, custom music score language was created to store the music in a text file. Then the pianist dons the head mounted display and sits in front of the piano. The pianist then gives a nod when starting to play the memorized music. HANDEL uses FFT phase correlation to detect a strong vertical displacement (the nod) to begin incrementing an internal counter to keep track of the current position in the piece. In the current implementation, the counter is incremented at a predetermined rate. A future improvement could have the counter's rate follow the actual piano playing through real-time audio analysis.

While the pianist plays the piece, nothing is overlaid on the pianist's heads up display (Figure 4.1(b)-(d)) until skin colour is seen by the head mounted camera. When skin is detected, the program assumes that the pianist is looking down at the hands. A specific hand is chosen based whether the pianist is looking to the left (Figure 4.1b) or to the right (Figure 4.1c), using FFT phase correlation (the same method used to detect the nod at the beginning of the session). The musical score at the current position, for the given hand, is displayed on the head mounted display,

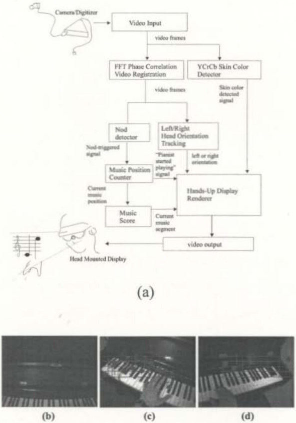

Camera/Digitizer
Video Input
video frames
FFT Phase Correlation Video Registration
YCrCb Skin Color Detector
video frames
Skin color detected signal
Nod detector
Left/Right Head Orientation Tracking
Nod-triggered signal
"Pianist started playing" signal
left or right orientation
Music Position Counter
Hands-Up Display Renderer
Current music position
Music Score
Current music segment
video output
Head Mounted Display
(a)
(b)
(c)
(d)

Figure 4.1: HANDEL Data Flow and Results

(a) HANDEL data flow, (b)-(d) Views from the head mounted display: (b) nothing overlaid when no hand is in view, (c) left hand part displayed for the left hand, (d) right hand part displayed for the right hand

and continues to update itself while the pianist is playing. The score is rendered at a fixed position on the left side of the display for the left hand, and likewise for the right hand (the score is not registered with the hand itself to avoid confusion from seeing musical notes moving with a moving hand). The virtual musical score disappears whenever the hands fall out of view (i.e. when the pianist looks up from the keys).

4.3 Footprint

Footprint, my second personal context application, used the feet instead of the hands, as the focus for computer assistance in a ballroom dancing application.

4.3.1 Foot-based Personal Context

Previous work on foot-based user interfaces falls under hardware based and computer vision based implementations. Applications for such interfaces include dance performance and choreography, motion capture for 3-D animation, and interactive entertainment.

Hardware based schemes rely on either body-mounted miniature magnetic, ultrasonic, or LED devices, often monitoring the motion of the whole body [63]. Exceptions include LED tracked slippers with vibrotactile feedback for games [218], a pressure sensitive carpet [59] to control video, and instrumented dance shoes that control music and artistic presentations [179, 41]. Hardware systems can quickly provide great accuracy and a wealth of data, but require infrastructure or worn equipment.

Computer vision systems make use of a camera or several cameras fixed in the environment, monitoring a specific location for body motion, such as walking and running. While some systems rely on body-placed markers to aid visual detection, many analyze the scene with only an a priori model of the human body [175, 138]. These systems are more interested in entire body motion rather than just foot motion, however. An exception is the work by [137], which derives 3-D motion data from a bicyclist's legs by analyzing specially textured shorts. Computer vision systems often free the human from wearing any special devices but need good lighting conditions and fast computers to process complex algorithms.

Numerous computer dance and choreography applications exist, mainly to empower dancers to create or influence music in their dance performances [164]. Specialized (and complicated) dance notations exist, such as the commonly used Labanotation system [134], which enable an exact description of any kind of dance. An interesting early example of using computers with dance notation for animation is [127], which translates Labanotation into a computer-readable representation and then renders the dance score into 3-D graphics. Instead of a dance notation, a physical model with joint angles and knowledge of human movement kinematics and geometries could be used to represent dance movement [188]. Dance notation and physical model representations are oriented towards specialists rather than a naive user, however. A somewhat more accessible representation specifically for naive users is a finite state machine [56]. The arcade and console game, "Dance Dance Revolution", uses scrolling combinations of left, right, up, and down arrows to direct players to make proper steps on an instrumented mat in sync with popular music, although the game emphasizes exact timing with predefined moves over personal expression [122].

4.3.2 Design and Implementation

Footprint operates on the same wearable computer testbed as described in Chapter 3. A personal context is achieved by having the user's feet trigger computer interaction when they are seen. Feet detection is accomplished by analyzing the frames captured by the video camera and exploiting a priori knowledge of the user's feet.

Footprint's demonstration application is an aid for beginners to practice ballroom dancing steps on their own. At present, the basic waltz steps are used. Figure 4.2 (a) shows Footprint's data flow. A practice session begins when the user, equipped

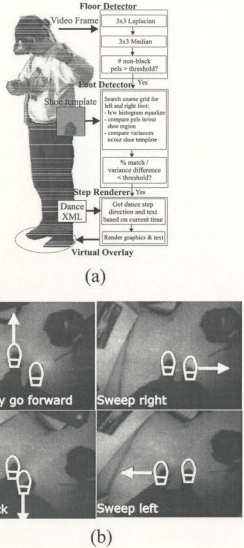

Floor Detector
Video Frame
3x3 Laplacian
3x3 Median
# non-black pels > threshold?
Yes
Foot Detector
Shoe template
Search coarse grid for left and right foot:
- b/w histogram equalize
- compare pels in/out shoe region
- compare variances in/out shoe template
% match / variance difference < threshold?
Step Renderer
Yes
Dance XML
Get dance step direction and text based on current time
Render graphics & text
Virtual Overlay
(a)
Smoothly go forward
Sweep right
Step back
Sweep left
(b)

Figure 4.2: Footprint's Data Flow and Results

(a) Footprint's data flow, (b) Dance step instructions as seen by the user's head mounted display with the wearable computer system, starts the application and loads the system settings and dance information. An internal timer is activated, allowing Footprint to synchronize dance steps to time. The user then performs the dance to music supplied by the computer. Whenever the user needs help, s/he simply looks down at the feet. Graphics and text indicating where the feet should move next are then presented on the head-mounted display (see Figure 4.2 (b)). This information disappears when the user looks back up. As a consequence, looking down at the feet provides a natural means to interact with the computer. Like in HANDEL, information is only shown when needed, minimizing graphical clutter on the limited-resolution head-mounted display.

The feet detection algorithm assumes that the user is wearing dark shoes against a fairly uniform floor. Thus, the algorithm applies a 3x3 Laplacian edge detection filter onto the current video frame from the camera, followed by a 3x3 median filter. For a uniform floor, such a process would yield a largely zero pixel image, with point noise being filtered out by the median filter. Floors with some structured patterns may yield some white edge pixels, whereas scenes with people, motion, objects, etc. would appear as numerous white pixels. Thus, the algorithm compares the total number of non-zero pixels against a threshold, and images below the threshold indicate the user is looking at a uniform floor. Otherwise, Footprint assumes the user is not looking at a uniform floor, and will not perform any foot detection.

If the current video frame passes the "floor test", then a predefined shoe template is matched against a coarse grid on the current frame. To avoid false matches and complications when feet overlap, the grid is set to the left half of the image to search for the left foot. At each grid position, the local rectangular region to be compared against the template is histogram equalized into black and white, and the number of matched pixels (with respect to the template), and local variances within the shoe and outside the shoe area are calculated. If the difference of variances between the pels within the shoe area and outside the shoe fall below a threshold (indicating the texture inside and outside the shoe are the same), or if the total difference within

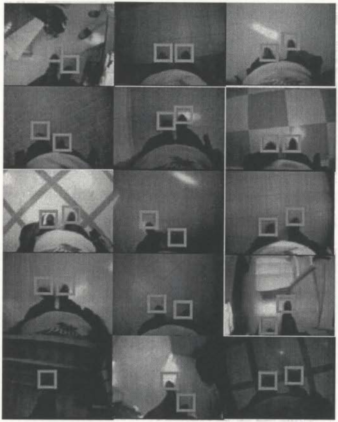


Figure 4.3: Foot Detection Under Various Conditions

Footprint's foot detection under different lighting and floor conditions, as seen by the head-mounted camera. Detected feet are highlighted by rectangles shoe area against the template exceeds a threshold (indicating the shoe area does not have a dark shoe), then no foot is detected. Otherwise, a measure proportional to the match against the template divided by the difference of variances is computed. The grid position with the smallest measure that still falls under a threshold is classified as a foot. The process then repeats itself to find the right foot, except the grid is set to the right of the discovered left foot position. Figure 4.3 illustrates the output of the foot detection algorithm under different lighting and floor conditions. On subsequent steps after the first, the system searches around the last detected coordinates first before performing a full coarse grid search.

The dance itself is represented as an XML text file, using custom markups. As seen in Figure 4.4, the dance moves are clearly represented, sufficient for a ballroom dancing application. The dance steps are given in sequence, using common ballroom dance step speed denotations ("quick", "slow", etc). Text descriptions are provided with each movement. This new "dance markup language" is similar to SMIL, a markup language for synchronized multimedia [47]. All the parameters controlling Footprint are stored in another XML text file. The XML representation is convenient for this particular application as opposed to a more general but complex dance notation system like Labanotation.

Footprint runs at about 4 frames per second on a Pentium 233 laptop, which includes all the image processing, video capture, and graphics rendering required by the ballroom dancing task. It detects the feet well and runs effectively with the basic waltz.

4.4 Discussion

HANDEL was tested successfully by the author on an acoustic piano for a short musical piece. While it proved to be very comfortable to use, there are numerous improvements that can be made: The virtual music score should constantly show the

```
<dance>
  <title>Basic Waltz</title>

  <step name="advance" duration="quick">
    Smoothly go forward
    <leftfoot direction="forward">Left first</leftfoot>
  </step>

  <step name="right" duration="quick">
    Sweep right
    <rightfoot direction="right">Right first</rightfoot>
  </step>

  <step name="right wait" duration="quick">
    Close
    <leftfoot direction="hold">Left arrives late</leftfoot>
  </step>

  <step name="back" duration="quick">
    Step back
    <rightfoot direction="back">Right first</rightfoot>
  </step>

  <step name="left" duration="quick">
    Sweep left
    <leftfoot direction="left">Left first</leftfoot>
  </step>

  <step name="left wait" duration="quick">
    Close
    <rightfoot direction="hold">Right arrives late</rightfoot>
  </step>
</dance>
```

Figure 4.4: The Dance Markup File for the Basic Square-Step Waltz

current clef to remind the pianist if the score refers to the left or right hand, as well as the current key signature. There is ambiguity when both hands are visible, since the system assumes only one hand will be in view at a time. This can be resolved by counting the number of skin coloured blobs in view, and showing the music score for both hands when two blobs (hands) are detected. The notes displayed on the head mounted display are large. Smaller notes, comparable to the size on real sheet music, would be desirable, if the resolution of the head-mounted display permits. Displaying proper piano fingering, having the computer listen and adjust the music to the pianist's playing and recording and playback of practice sessions would be useful additional features.

HANDEL's FFT phase approach, combined with skin detection and the assumption of a seated pianist in a front of a nearby piano, is sufficient enough for distinguishing left from right, and to detect a hand. A future improvement would be to employ affine or projective based scene analysis (such as in the work specific to wearable camera systems in [149]). With a priori knowledge of a flat piano keyboard, this could form a richer interface with a pseudo 3-D world model, allowing for effects like 3-D texture-mapped coloured keys responding to hand motion and different musical visualizations while playing (e.g. a 3-D frequency analyzer plot).

Footprint can benefit from a faster computer, foot pose recognition, and further user tests to optimize the dance instruction presentation over more types of dances. Other modes for computer-assisted teaching can be explored, such as having Footprint measure the feet movements to assess a proper step. Extending the system to recognize and coordinate with a live partner would also be desirable.

Foot detection can be improved by using multiple templates to account for varied foot orientations. Combining the template matching algorithm with active contour modelling may yield more accurate detection and foot pose estimation. The simplicity, robustness, and efficiency of the foot detection algorithm demonstrates the usefulness of computer vision for personal context user interfaces.

The piano itself can be seen as a workspace for pianist or a composer, as the desktop is for an office worker. Both environments use paper based documents. Thus, Live Paper technologies [204], which are being used to enhance the desktop, could be applied to the piano. The piano and its sheet music could be augmented, either by a head mounted display or an external projection system coupled with a video camera.

Besides ballroom dancing, other foot-based personal context applications can be developed, such as for various sports and martial arts, mapping and pathfinding, physical exercise, and walking therapy for the injured and disabled, and performance dance. In general, body-mounted cameras and a personalized model of the user's body running on a wearable computer promise new opportunities and new approaches for traditional vision and image processing problems.

The use of an XML based dance step file to represent content and an XML configuration file as a "style sheet" casts Footprint as a browser for a personal context wearable computer interface. Because the dance markup language is a simple description of the needed dance steps, it can be interpreted for different purposes on other platforms. For instance, another wearable computer could create XML-based data on the fly from streaming sensor data. A 3-D capable XML desktop browser could translate the dance step file into a dancing computer-generated character that could be incorporated into a virtual reality environment or a computer graphics movie. Online XML database engines could index and catalogue the dance step file in a repository, allowing for text-based searches for human gesture and motion. In general, context-aware applications can exploit XML as a foundation to create readable, portable, and indexable notations for human gesture, motion, and interaction with the real-world. Since gesture, motion, and interaction vary over time and depend on different conditions, context-aware notations might adapt properties and behaviours from scripting languages and temporal-based notations (such as SMIL).

Chapter 5

The Mosaic is the Interface

In Chapter 4, I use the idea of Personal Context to create an interface for a wearable computer that provides timely and relevant information to the user. Timeliness is achieved by having the wearable computer "always on" (one of the key wearable characteristics in Chapter 2) and relevance is achieved by sensing the activity user's body parts in the context of some assumed task (e.g. playing piano or dancing, as in Chapter 4). The results are presented through augmented reality, with directions from an automated "expert" (really a preprogrammed set of instructions running against a timer) being registered with the scene.

In this chapter, I replace the "automated expert" with a remote human expert. This impacts the wearable's sensing, and requires a new interface for the remote expert. I introduce the notion of "the mosaic is the interface" as a solution to these issues.

5.1 Wearable Telecollaboration

Telecollaboration is a commonly cited use for wearable computers and augmented reality. Some examples are presented in Starner et al [227] and Azuma [11]. A wearable could be a simple networked terminal, and the user could exchange information with remote colleagues by text messaging. Or a wearable could be a video conferencing hub, running on a wireless network and using head mounted camera with audio to stream video with a remote participant. The wearable user and remote user can collaborate with a virtual telepointer.

Field repair with remote help is the standard scenario for wearable AR telecollaboration. The wearable computer user could be performing on-site repairs, while taking advice from a remote expert via a wireless video connection (as in the collaborative field repair of a bicycle in [79]). Such human-based task assistance can provide a rich dynamic between the wearable user and remote expert, and offers better flexibility against unforeseen complications than automated task centered solutions (like the printer assistant in [72]). This story can be recast into a scenario with a paramedic with remote telemedicine, a search and rescue system guided by experts, and a means for strategists to support soldiers or police in the midst of a crisis. Chapter 2 has more scenarios involving wearable computer and augmented reality collaboration.

Fussell et al [79] and Bauer et al [16] present detailed user studies on the benefits of audio-video in a shared visual collaboration. Fussell et al [79] compares results against physical presence of the expert and an audio-only remote link. Bauer et al [16] have cases that feature full video and audio, but measures performance of a telepointer and freeze-frame feature for the remote expert. The freeze-frame feature allows the remote expert to concentrate on important views and deal with confusing head motion in the video. The telepointer allows the remote expert to point and gesture at actual objects that can only be referenced indirectly in an audio-only collaboration.

The testbeds used in both studies are similar to the wearable testbed in Chapter 3, and the studies features a repair task scenario with remote assistance. Fussell et al [79] conclude that shared visual collaboration using video taken from the wearable user's point of view is beneficial, moreso than audio-only, although not quite as good as a live colocated colleague. The limitations of the video implementation include a limited field of view, lack of telepointers or means to indicate gestures from the remote user onto the video, and an inability to read the remote user's facial expressions. The last limitation can be addressed easily by incorporating a standard "talking head" video frame like in [22]. Bauer et al [16] present solutions for the other two limitations: freeze-framing and a telepointer. Although neither solution improved task completion time significantly versus no solution at all, users overwhelmingly preferred using these features over using voice alone.

5.2 Registration for Collaboration

I am interested in going beyond the direct video conferencing methods in [79] and [16], to mediate the collaboration between the worker and the expert by leveraging AR registration. The model of this collaboration, illustrated in Figure 5.1, is as follows. The field worker, wearing a system like the wearable testbed in Chapter 3, looks at something of interest. A digital camera mounted on the worker's head captures live video of what the worker is looking at. The live video is captured by the wearable computer, and transmitted to the remote expert's desktop. The remote expert writes or draws an annotation on one of the images in the video stream (e.g. circling and labelling something of interest). This annotation is sent to the field worker's wearable computer, and overlaid as a graphic on the field worker's see-through head mounted display. The graphic is anchored with respect to the part of the scene it was associated with. Regardless of how the user moves or turns, the annotation remains registered with the tagged part of the scene. This ability to draw onto the video will enable remote users to gesture and type onto the scene and address one of the limitations in [79].

There is a need for a video-based registration algorithm for use in the above collaborative augmented reality scenario. All registration algorithms may be assessed in terms of their precision, robustness and speed, but the relative importance of these depends on the application. Traditionally, real-time operation (speed) has been secondary to precision in applications such as remote sensing and medical image

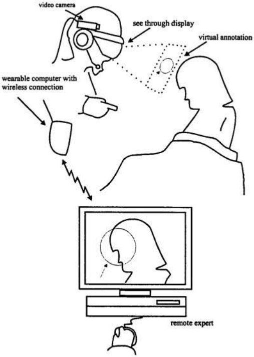


Figure 5.1: A Collaborative Wearable Computer Scenario with Augmented Reality

fusion. In the wearable computing context, speed is a top priority, precision (at least in many applications) is not critical, and robustness can be handled cooperatively between the computer and the user. Robustness refers to the registration algorithm's reliability in the context of "unusual" scene conditions such as changing illumination, motion that violates theoretical assumptions, etc.; while a degree of robustness is important in wearable computing, the user can play an active role if errors are made, and assist the algorithm in fast recovery.

Amongst the registration options presented in Chapter 2.3.2(page 28), mosaicing is the most appealing choice. It elegantly combines registration with a user interface medium, i.e. "the mosaic is the interface". Mosaicing methods have been used in a wide variety of areas. There are photoimagery applications like personal panoramic photos from commercial packages like VideoBrush [52] and Apple Quicktime Panorama Maker [45], and iMove [99]. Panoramic video shots can build worlds for virtual reality such as [39]. Planetary photographs and weather maps are stitched together from sequences of photos from satellites and spacecraft. Detailed mosaic scans of the interior of the human body help doctors in medical visualization [120]. Mosaicing techniques can provide super-resolution image enhancement [261] and video indexing, compression, and 3-D scene reconstruction [135]. Chapter 6 surveys and compares several mosaicing methods in detail.

An image mosaic can be constructed automatically with knowledge about the scene around the camera (see Section 6.1 for details). Thus, the set of transformations used to generate a unified view from a series of individual snapshots can form the coordinate system to register a remote user's telepointer and gestured drawings. The single final image combining all the transformations and snapshots gives the remote user a greater field of view into the wearable computer's surroundings than a normal video conferencing link. A large mosaic can overcome video instability from constant head motion. Unlike manually freezing one video frame in [16], incorporating a live stream of video frames automatically in real-time ensures the remote user sees a single montage of activity in the past and present, before any making future decisions. This feature addresses the field of view issue in [79].

Using the wearable testbed in Chapter 3, I built an augmented reality telecollaboration system, AugR, as a working proof-of-concept prototype that uses mosaicing for collaboration and registration.

5.2.1 Related Work in Registration

Using image mosaics as in AugR is just one approach to the registration problem. For some applications, other technologies are suitable. For instance, Feiner et al use GPS to obtain the user's position to register information with buildings at a university campus [71], and Feiner and MacIntyre use ultrasonic sensors to obtain position/orientation for a laser printer maintenance example [72]. However, in surveying the hardware and software issues in augmented reality registration, Azuma notes that even the best hardware has inherent limitations in precision and performance [11]. For instance, GPS is not useful for short distances or indoors, ultrasonic sensors require line of sight operation, and radio-frequency and magnetic trackers suffer from interference from nearby metallic objects. This research concentrates on a visual-based registration scheme, which can eventually be integrated with other modalities.

Existing registration methods can be classified as feature based and global image techniques (a detailed survey of image registration methods can be found in Brown [30]). Feature based registration techniques detect and track special features in an image. The main challenge with these techniques is finding good features to track. One approach is to search for features of interest and compare them with a database of features. This database of "feature templates" can produce real-time performance, like in Uenohara and Kanade [241]. However, considerable time must be spent to construct the actual database, which does not lend itself well to uncontrolled environments with unknown features and objects. Another feature based registration strategy is found in the use of artificial fiducials or markers, which are placed into the scene [40]. Such fiducials can be designed with very distinguishable characteristics, such as a unique colour or shape (e.g. ring) for easy and fast feature detection in an image. A hardware variant is to place LED beacons into the scene, as demonstrated in work like [14]. But this requires the user to place the fiducials onto the scene in advance, which is again not suitable for an uncontrolled environment.

Instead of using artificial fiducials, some registration algorithms search for natural features found in an image that can provide a reliable motion estimate between image frames. A corner is an example of a natural, reliable feature - a junction of markedly different textures in an image. Previous work includes methods such as Zoghlami et al [260], whose geometric corner detection handles large frame displacements, and Morimoto and Chellappa [161], which specializes in estimating rotation. However, these methods are not real-time. If found, environmental fiducials can provide fast and easy image registration as with artificial fiducial methods. The problem lies in quickly finding suitable fiducials. I initially experimented with fast fiducial image registration methods, but found them to be very sensitive to local scene motion as opposed to the general camera motion (e.g. fiducials would often lock onto moving people in a room rather than the background of the room). Global image techniques estimate the motion of all pixels of the image, and typically make use of some kind of optical flow estimation or minimization the overall squared difference between image frames like [166]. These algorithms produce precise results, and they are robust. Their strength lies in processing every pixel in the image frame. But as a result, many algorithmic operations are required per pixel, for solving a set of linear equations, computing the optical flow, estimating errors between frames, or building Gaussian pyramids. My work avoids the complexity of multiple iterations to find a optimum fit by doing a single pass over an image in the frequency domain, which can be computed simply by a few Fast Fourier Transforms (FFTs).

While traditional augmented reality like "Windows on the World" in [72] and collaborative wearable AR systems like Kato et al's Shared Space interactive tabletop [114] and Schmalstieg et al's multi-user, multi-device augmented workspace [211] feature registered overlays in the wearable user's field of view, none of the registration is leveraged to benefit a remote collaborator on a desktop computer and expand the field of view beyond a video camera's field of view. Mosaicing for wearable augmented reality is featured in other research, such as [149], where mosaicing is used to register labels on people and objects. Starner et al [227] mention the potential of telepointing on a static final mosaic image generated on a server and seen by a remote user, whereas I am interested in a live mosaic generated "on-the-fly" and locally by the wearable testbed. Mann's recent work in [152] and [150] use a method driving image mosaicing to exploit flat surfaces in the scene to render overlays as 3-D texture maps covering walls and signs. This research concentrates on simple 2-D overlays, registered against a flat mosaic.

Kourogi et al's work in [124] and [126] feature a wearable testbed somewhat similar to the testbed in chapter 3, and use mosaicing for fast registration. They rely on several multi-processor servers to perform the registration calculations but deal with scaling and user movement issues by matching the head camera video against multiple pre-generated mosaics for different locations. Kourogi et al's latest work in [125] follows up the previous research by hybridizing the system with inertial sensors to improve registration.

5.2.2 Image Registration by Phase Correlation

AugR employs image registration by phase correlation, a method that aligns displaced images by matching distinctive textures, as opposed to individual pixel comparisons. Frequency domain based image registration has a long history, but has largely been restricted to only estimating translation between image pairs through calculating phase correlation [130]. By computing the phase correlation via the Fast Fourier Transform, robust results are achieved, which can be implemented on real-time hardware [182]. Faster computation can be achieved at the cost of robustness and accuracy if the phase correlation is computed only with one-dimensional FFTs [3]. DeCastro and Morandi [62] and Lucchese et al [142] propose frequency domain algorithms that can search for rotation and affine transformations respectively, but they are computationally expensive. However, the image rotation (about the optical axis) and scaling information of an image can be found in the magnitude spectrum independently of translation. Reddy and Chatterji [199] exploit this property to implement an image registration algorithm that can compute translation, rotation, and scaling with only three FFTs. I use this algorithm for our image registration application. The method is described, following [199]:

First consider two consecutive images from a video sequence, f_1 and f_2. If they differ only by a displacement (Δx, Δy), then

$$f_2(x, y) = f_1(x - \Delta x, y - \Delta y) \tag{5.1}$$

Applying the Fourier transform and the Fourier shift theorem gives:

$$F_2(u, v) = e^{-2\pi j(u\Delta x + v\Delta y)} F_i(u, v) \tag{5.2}$$

Then note that the cross-power spectrum of F_1 and F_2 (where F_2^* being the complex conjugate of F_2) is

$$\frac{F_1(u, v)F_2^*(u, v)}{|F_1(u, v)F_2^*(u, v)|} = e^{2\pi j(u\Delta x + v\Delta y)} \tag{5.3}$$

This result shows that the translation information between f_1 and f_2 can be found entirely in the cross-power spectrum. Ideally, the inverse Fourier transform of the above result gives an impulse located at (Δx, Δy). With real images, there will be many "impulses", due to different motions in the scene (e.g. people moving, parallax effects) as well as noise. I am only interested in the motion of the camera, and so I take the largest "impulse", which corresponds to the dominant scene motion (assumedly from the camera). The process of obtaining this result is the general phase correlation algorithm. Finding the dominant scene motion by just locating the largest correlation peak demonstrates the robustness of this method.

Now consider the case where two consecutive images f_1 and f_2 differ by a displacement, $(\Delta x,\ \Delta y)$, and a rotation, Δq, about the optical axis of the camera, i.e.

$$f_2(x,y) = f_1(x\cos\Delta\theta + y\sin\Delta\theta - \Delta x, -x\sin\Delta\theta + y\cos\Delta\theta - \Delta y) \tag{5.4}$$

By the Fourier shift theorem and the rotation property, the Fourier transform of the above equation becomes

$$F_2(u,v) = e^{-2\pi j(u\Delta x + v\Delta y)} F_1(u\cos\Delta\theta + v\sin\Delta\theta, -u\sin\Delta\theta + v\sin\Delta\theta) \tag{5.5}$$

Since f_1 and f_2 are real images (i.e. real 2-D signals with no imaginary components), the rotation information is entirely in the magnitude portion of the above result. That is, the magnitudes of F_1 and F_2 are related by

$$|F_2(u,v)| = |F_1(u\cos\Delta\theta + v\sin\Delta\theta, -u\sin\Delta\theta + v\cos\Delta\theta)| \tag{5.6}$$

The magnitude of F_2 is a rotated version of the magnitude of F_1. So, this is converted to polar coordinates, then the rotation becomes a shift, and I can apply the phase correlation method here to find $\Delta\theta$.

Finally, let us consider the case if f_2 is a scaled version of f_1, i.e.

$$f_2(x,y) = f_1(s_x x, s_y y) \tag{5.7}$$

where (s_x, s_y) is the scaling factor along the x and y axes. By the Fourier scaling property, the Fourier transform of the above equation is

$$F_2(u,v) = \frac{1}{|s_x s_y|} F_1(\frac{u}{s_x}, \frac{v}{s_y}) \tag{5.8}$$

I can convert the axes to a logarithmic scale, which turns scaling into a shift (ignoring the factor $\frac{1}{|s_x,s_y|}$), i.e.

$$F_2(\log u, \log v) = \frac{1}{s_x s_y} F_1(\log u - \log s_x, \log v - \log s_y) \tag{5.9}$$

If I let $u' = \log u$ and $v' = \log v$ and take the magnitude of the above equation, then

$$|F_2(u', v')| = |\frac{1}{|s_x sy|} F_1(u' - \log s_x, v' - \log s_y)| \tag{5.10}$$

As with the rotation case, I can apply the phase correlation method here to find the "shifts", $\log s_x$ and $\log s_y$, from which I can get the scaling factor.

Knowing the results of the rotation and scaling cases, and letting $s = s_x = s_y$, I can combine them to obtain

$$|F_2(r, \theta)| = |\frac{1}{s^2} F_1(r - \log s, \theta - \Delta\theta)| \tag{5.11}$$

Thus, applying the phase correlation method on the polar representation of the magnitude spectra can obtain the log of the scaling factor and rotation angle. This information can then be used to scale and rotate the original image f_1. The phase correlation can then be reapplied between f_2 and the scaled and rotated version of f_1 to find the translation estimate (Δx, Δy).

I implemented the phase correlation method following the implementation guidelines given in [199]. I added new modifications to accommodate the type of images AugR uses, i.e.

1. I obtain sub-pixel accuracy in the phase correlation method by finding the maximum peak through a cubic interpolation around the discrete maximum peak region in the cross-power spectrum.

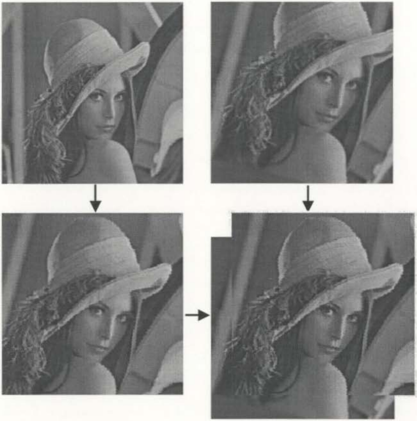


Figure 5.2: Registering "Lena"

original image (upper left) registered with a shifted, clipped, scaled, and rotated version of itself (upper right). The scaling and rotation are first computed and applied (bottom left) before translation is estimated and mosaic is generated (bottom right)

2. Before scaling and rotation are estimated, images with the dimensions not equal to a power of 2 are clipped to a centered square of width and height equal to the smaller dimension, and then scaled to the nearest lower power of 2 (e.g. a 160x120 pixel image would be clipped to a centered 120x120 pixel square, and then scaled to 64x64). This uses more image information for alignment, albeit in a scaled form. Just taking a subset at the closest power of 2 (e.g. the 64 by 64 pixel square in the middle of the picture) could mislead the phase correlation.

3. Before translation is estimated, the images are rescaled to a square with a dimension equal to the nearest lower power of 2 (e.g. a 160x120 image would be scaled to a 64x64 square). Again, this uses as much image information as possible. Clipping is not required at this stage, because only translation is involved, which can be accommodated in a square through scaling factors.

Figure 5.2 demonstrates the image registration algorithm for the original "Lena" image (256x256 pixels) and a shifted, scaled, clipped, and rotated version of itself. There are a few misalignment errors only around the left and bottom boundaries, suggesting a correct translation estimate, but problems with scaling. The angle of the face appears aligned with the transformed version of the original image. In the initial stage of the algorithm during the rotation and scaling estimate, some inaccuracy is introduced when comparing the original image against the new information in the rotated frame (the bottom left of the rotated image). While single pixel errors from a translational estimate amounts to small perceivable differences, a few pixels of error in polar coordinate space can amount to a few degrees or scaling factors of change. Such differences translates to many pels of error proportional to the arclength traced in polar space, more notably for scaling, since it is a multiplicative term. Compounding this problem, scaling in this polar transformation scheme is susceptible to discretization errors. Because the polar coordinate view of the image is essentially a radial distance map overlaid against a regular horizontal/vertical grid of pixels, accurate pixel measurements can be found closer to the center, while interpolation is needed for pixels at distant radii. Since radial distance is associated with scale, the polar transformation for phase correlation is very sensitive to scaling errors, and cannot be relied upon for large scaling effects.

5.3 AugR Implementation

AugR went through two incarnations as an augmented reality prototype system. It was originally designed to provide a tour of the lab for a single user, with preset "hot-spots" that activated videos when gazed upon by the user (see Figure 5.3).

The original system used an earlier image registration algorithm using a feature-based scheme (see Chapter 6.3, page 110). Block matches were made to obtain motion estimates of identified feature points between the reference image and the current image. From the motion estimates and the former positions of the "corner" points, the projective transformation that registers the image pair was computed. In our early trials, I discovered that this method is very vulnerable to moving objects and people, since it cannot distinguish the foreground from the general background when choosing feature points.

Because of this, I investigated the Fourier based method of registration. Upon implementing the Fourier based image registration algorithm, I incorporated the new algorithm, and redesigned the augmented reality prototype system to realize the field worker / remote expert scenario. As shown in Figure 5.4a, the application begins by setting up a TCP/IP session for a remote expert or field worker.

Once a network connection is established between the remote expert and the field worker, the field worker application continuously transmits video to the remote expert as a stream of RGB 160x120 pixel JPEG images and updates to the image mosaicing world model using the Fourier based image registration algorithm. The remote expert application constructs the resulting image mosaic, upon which the user can draw annotations (see Figure 5.4c), which are transmitted back to the field worker, and

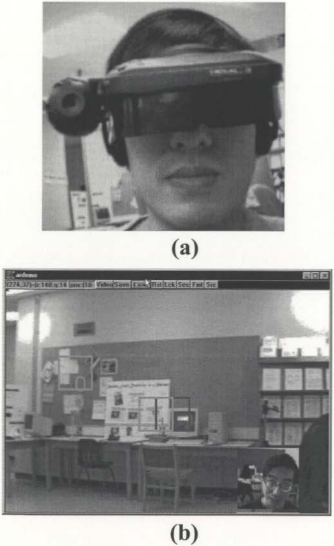
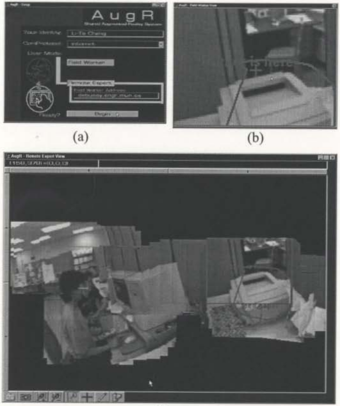


Figure 5.3: Prototype Augmented Reality System

(a) Virtual I-O Glasses + Camera (b) Screenshot with Overlaid Graphics & Video



Your Identity: Li-Te Cheng

Comm Protocol: internet

User Mode: Field Worker / Remote Expert

Field Worker Address: debussy.engr.mun.ca

(a) (b)

(c)

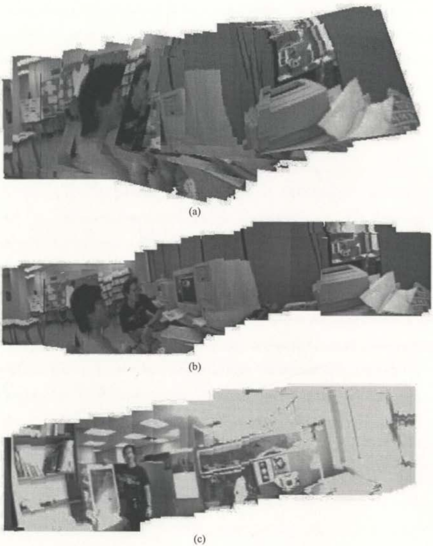
Figure 5.4: The Current Version of AugR

(a) Setup screen (b) Field Worker View with Registered Annotations (c) Remote Expert View with Image Mosaic and Annotations (the field worker's current viewport is indicated by the small rectangle near the printer) properly registered in the field worker's display (see Figure 5.4b). The remote expert also can zoom in, zoom out, and scroll around the image mosaic, as well as save the entire mosaic to file and capture the video stream as a series of JPEG files. The total speed (image capture, registration, network transmission, scaling, rotating, and rendering) of this program without optimization averages about 1 frame per second. If only 2-D translation using phase correlation is used, the speed is about 3-4 frames per second.

Figure 5.5 illustrates more image mosaicing results. Figures 5.5a and 5.5b compare the original feature-based method with the Fourier image registration scheme over the same video sequence. The sequence shows a wide camera pan with perspective distortion due to objects at a variety of distances from the camera. Both people in the scene are moving (one person is working at the computer and the other gets up from his chair, walks around, and comes back). Also, for brief instants, other people in the lab walk through the scene. The perspective distortion and human motion create the malformed image mosaic in Figure 5.5a. The perspective distortion is still apparent, but the Fourier image registration algorithm is able to compensate for extraneous motion, producing the better-looking mosaic in Figure 5.5b. Figure 5.5c shows a mosaic generated by the Fourier image registration algorithm under variable lighting conditions. The camera begins on the left, where fluorescent lighting in the lab dominates, and moves to the right, where sunlight from the windows eventually overwhelm the camera. Despite these conditions, the algorithm is still able to construct a comprehensible image mosaic.

5.4 Discussion

Figures 5.4c, 5.5a, and 5.5b show the effectiveness of the Fourier based image registration algorithm, but the most apparent limitation is the lack of perspective correction, since only translation, rotation, and scaling are accounted for. Rotational and scaling terms are sensitive to error, more so for scaling. Despite these problems and the low



(a)

(b)

(c)

Figure 5.5: Mosaicing Results

Same video sequence with human motion using (a) feature points, (b) Fourier registration, and (c) another sequence demonstrating the Fourier registration algorithm with variable lighting frame rate, the camera can move at a moderate rate and the algorithm can produce comprehensible mosaics, because the Fourier method does not need large overlap between frames. Also, the remote user can draw registered annotations effectively onto the live video mosaic being generated "on the fly" by the wearable testbed. Informal use of the system by the author and a few others noted a perceptible lag (around one second depending on the network traffic) but found AugR usable for short demonstrations.

With further optimization, this may be a registration algorithm suitable for a wearable computing application. But what about robustness? Regardless of how powerful an algorithm may be, it is inevitable that some unforeseen pathological case will cause the registration to fail. Even automated compensation methods could fail. Eventually user intervention may be needed to recalibrate the system. In the current system, the remote expert is responsible for recalibration, since any errors will become readily apparent in the image mosaic. The remote expert can recalibrate by dragging the field worker's current viewport in the image mosaic (the smaller rectangle around the head in Figure 5.4c) to the desired position. This effectively corrects the current image registration by a displacement. This form of manual recalibration makes the image mosaic into an intuitive user interface for correction.

As a proof-of-concept, AugR implements fully the proposed vision of a wearable augmented reality telecollaboration system, with its simple phase-correlation registration and presents a user interface driven by mosaicing entirely on a wearable platform. It uses mosaicing to compute registration information, and to render a user interface for remote collaborators to interact with wearable computer users. By allowing remote experts to draw annotations onto a live mosaic, AugR overcomes the camera field of view and interaction problems cited in earlier video collaboration studies. The next chapter considers mosaicing as a general AR registration engine, and compares AugR's phase correlation scheme against existing methods.

Chapter 6

An Evaluation of World Modeling Methods

In Chapters 4 and 5, I presented a range of wearable augmented reality systems. The Personal Context systems in Chapter 4 present an interface from only the user's activity and AugR in Chapter 5 leverages mosaicing as a two-way telecollaboration medium. Underlying these diverse interfaces is a simple algorithm to register virtual overlays into the wearable user's field of view on a head mounted display. Specifically, HANDEL presents left or right hand specific musical notation, Footprint attaches instructions to the user's feet, and AugR places the remote expert's annotations onto the wearable user's viewpoint.

Thus, an AR registration scheme powered by sensory data inputs (e.g. from a camera) is the computer's model of the real-world around the user, and puts virtual information in sync with the tangible environment. With no registration and no sensory input, information is totally detached from physical reality, existing as a phantom that always shows up with no regard to what the user is doing or anything in the environment. Examples of this are typical PDA tasks on a wearable, like calendaring, messaging, and web browsing. With information purely sensor driven with no guiding registration model, information has some empathy with the user and environmental awareness, but lacks any sense of physical space and still remains detached from objects of interest in the scene. Examples of this are sensory visualization and notification, like reading a temperature or GPS coordinate readout, being reassured by an affective user interface monitoring galvanic skin response and heart rate, or receiving a reminder of a person's name from a wearable face recognizer. Sensory data feeding registration produces information that can be attached to scenery giving a strong sense of spatial context. For instance, this can manifest as a GPS enabled map with the user's location and directions to places of interest and virtual directions from a Personal Context system or from a remote expert teleoperating a camera attached to an object to repair.

My research investigates using mosaicing as an AR registration technique. As discussed in Chapter 5, mosaicing benefits telecollaboration user interfaces as well as the registration of virtual annotations. The raw motion information derived from the same mosaicing scheme in Chapter 5 also power the Personal Context systems in Chapter 4. This chapter systematically compares a variety of mosaicing algorithms as engines for augmented reality registration for wearable systems.

6.1 The Image Mosaicing Approach

Image mosaicing constructs a single high resolution image, called the mosaic, by combining together a series of low resolution images. To build the mosaic, a transformation mapping each image frame into the next is estimated. The transformation contains a motion estimate of the camera. The inter-frame motion estimates that form the mosaic provide the orientation information of the camera during an image sequence. Composing these estimates as perspective matrix transformations lets mosaicing become an AR registration model, mapping between the virtual and the real world reference frames. Such a mosaic could provide a reference image for transmission to other users via the network or for video database archival [108]. Combining the transformations with interpolation can lead to supersampled super-resolution image mosaics [261]. Also, such a mosaic can form the basis for constructing a virtual environment, possibly with depth information [234].

6.1.1 Camera Motion Assumptions and Projective Transforms

Projective matrix transformations contain translation on the image plane, perspective chirp, zoom, and rotation around the depth axis. For distant scenery, the watched scene can be treated as a flat plane and 2-D translation on the camera plane can be modelled by projective transformation with little impact from 3-D object parallax. Mosaicing schemes tend to use some form of projective transform or a subset (e.g. affine, translation-only, etc) since the transformations map well to camera motion in many picture-taking conditions (e.g. non-moving photographer), and the transforms compute nicely in linear algebra [153]. Even a 360 degree panoramic mosaic taken from spinning the camera around a fixed vertical axis can be modelled as a series of translation-only transforms along the image x-axis.

The fundamental problem behind any image mosaicing algorithm is to find the set of projective transformations that will map each pair of images (a current frame and a reference frame in a video sequence) such that the transformed images form into one single, seamless image composite. The distinguishing factors of the algorithm are:

- how it finds the transformations between frames
- how fast it can find the transformations
- how much error is apparent in the final composite
- how well it can deal with different types of video sequences

6.1.2 Basic Algorithm

The process of creating an image mosaic is typically as follows:

1. Image Acquisition

   A series of images are captured from a camera, video, or any source of image files. Acquisition may occur in real-time or separately from the other steps.

2. Image Processing

   Some or all of the acquired images from the previous step are processed by some kind of image processing. The image processing can serve a number of purposes: to reduce noise, to simplify or accelerate the transform estimate in the next step, etc. For example, a Gaussian Pyramid may be generated to iteratively improve transform estimation from the global to local scale.

3. Transform Estimation

   The images are compared to each other (usually image frames adjacent in time) and a transformation to merge them together into a single mosaic is estimated. Many different schemes could be used to construct such a transformation. Schemes could be as simple as a translational motion estimate or as complex as a perspective warping transformation. A complex model may produce more accurate transformations, but is typically more complex to compute (e.g. solving the 8 parameters of a perspective transformation versus a 2 dimensional block match).

4. Mosaic Construction

   Once the transformations are estimated, the final estimated transformations are applied on all the captured images and the result should be a single, unified image mosaic. Often the transformed images are blended together to eliminate seams between frames arising from effects like placement errors, moving objects, and lighting variations over the scene and time. Some schemes even use the constructed mosaic to do further estimation refinements.

The above steps can be refined further. A general image mosaicing algorithm, loosely based on the approach described by Mann and Picard [153] and [88] (as illustrated in Figure 6.1) is:

1. Construct a Gaussian Pyramid for a video sequence (repeatedly lowpass filter then downsample each video frame, generating quarter-sized, sixteenth-sized, and smaller versions of the original)
2. For each image pair (a current image and a reference image) in the sequence:
   (a) From the lowest resolution to the highest resolution images in the pyramid, refine an estimate for the correspondence of points in the two images.
   (b) Solve for the perspective mapping between the finalized and predicted sets of points from step (a).
   (c) Apply the perspective transformation on the current image

The Gaussian Pyramid provides a global-to-local search structure for mosaicing parameters [33]. The highly blurred images at the highest level in the pyramid leave only major image features. By going from the highly blurred to the original image in the pyramid, the algorithm attempts to avoid local minima of correspondence by beginning with an image dominated only by major features and refining an iterative search for the best correspondence points.

The simplest version of this general algorithm would be as follows: Four points are chosen, either arbitrarily or from some kind of feature selection scheme, depending on the actual mosaicing algorithm. Once four points are determined, their corresponding location in the reference frame is needed. Depending on the mosaicing scheme, a feature matching scheme or some approximate motion model can be used to find an area in the reference frame that best matches a corresponding area around or defined by the feature points in the current frame. The areas forming the "best match" define the feature points in the reference frame. The feature points in the current and reference frame can be applied to solve for the perspective transformation parameters. These parameters are solved by substituting the four original and four transformed

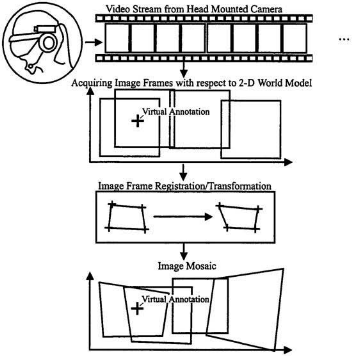

Video Stream from Head Mounted Camera
...
Acquiring Image Frames with respect to 2-D World Model
Virtual Annotation
Image Frame Registration/Transformation
Image Mosaic
Virtual Annotation

Figure 6.1: General Image Mosaicing Algorithm

feature points ($(x_1, y_1), \ldots, (x_4, y_4)$ and $(x'_1, y'_1), \ldots, (x'_4, y'_4)$ respectively) into the general perspective transformation equation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \frac{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}{\begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + 1} \tag{6.1}$$

where the perspective parameters $a_{11}, \ldots, a_{22}$ define shear/rotation, b_1, b_2 define translation, and c_1, c_2 define perspective distortion. The substitution forms the following system of linear equations with eight equations and unknowns:

$$\begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x'_4 & -y_4x'_4 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_4y'_4 & -y_4y'_4 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ b_1 \\ a_{21} \\ a_{22} \\ b_2 \\ c_1 \\ c_2 \end{bmatrix} \tag{6.2}$$

By solving this system of equations, the perspective parameters can be found and the perspective transformation is constructed. Composing the transforms from the beginning to any point in the video sequence generates a mapping from the image display's coordinate frame to the current video frame's reference system. The question of whether four point correspondences can be found accurately and reliability will be discussed later. Clearly, an error in any would cause a significant error in the transform, suggesting that a many point correspondence method with least squares deviation of the transform parameters might be more robust.

Once the transforms are determined, a series of video snapshots can be warped by composite transforms to form a single image mosaic. A remote expert's virtual annotation drawn on desktop rendered mosaic can be mapped to the current video frame seen on a head mounted display via an inverse composite transform. In Chapter 4's HANDEL, the translation components of composite transformation matrix provides head-motion estimates to infer whether the user is looking towards the left or right.

Simpler inter-frame motion can be derived from affine or simpler transformations (e.g. translation-only, rotation-only). This limits the possible kinds of camera motion that can be modelled, but reduces the number of required points to match and the complexity of the set of linear equations. For example, a 2-D phase correlation (described in 6.2.4) only computes a displacement vector, which is then applied to all points in the image.

6.1.3 2-D and Cosmetic Issues

Due to 3-D object parallax, in-scene object motion, camera optics, and inaccuracies in the registration scheme, the final, composited mosaic usually is not truly seamless. Often there is lighting variation amongst image frames, producing a "light-dark patched" effect in the final image and sharp seams between composited frames. There are many solutions to this problem. A weighted image blending filter around the joined regions can be applied as in [32]. Szeliski and Shum's "deghosting" method [219] is a variation on the weighted blending filter idea, and Davis uses Vornoi regions to help map proper blending areas [60]. Peleg [184] spreads the edge effect on the seams to a larger area iteratively. Histogram equalization can nicely balance the lighting level amongst the frames [89], and can be applied during the initial motion estimation process to reduce error from lighting effects. Camera distortion can be compensated by modelling the optics with some "a priori" transform (often the inverse of a radial distortion filter) on incoming image frames. In his everyday wearable imaging system, Mann exploits auto-gain control to integrate multiple images taken at different exposures to produce a single lighting compensated result [151].

In my research, with the emphasis on registration rather than presentation, and fast and simple methods, I did not use blending to clean up the final image mosaics. Being independent of the actual registration scheme, such methods could be applied later to improve the appearance of the mosaics for final rendering.

6.2 Existing Mosaicing Methods

While the mosaicing process can be generalized as a series of perspective transformations, the actual methods in practice vary in how they compute the individual transformations between frames. Typically the interframe motion estimates are derived from consecutive frames, as suggested in the generalized steps in the previous section, although some methods match against a global composite image (e.g. [60]). This section will describe several approaches.

6.2.1 Predetermined or Manually Determined

The trivial solutions to the interframe motion estimation problem include having a human plan the motion in advance, or assess the motion "post-hoc" (or combine both of these solutions). For instance, the early versions of Apple QuickTime VR [37] assumed a single 360 degree camera pan to capture a cylindrical panoramic mosaic, and some of the early packages for QuickTime VR allowed mosaics to be stitched together when the user manually selected corresponding control points between frames. Neither predetermined or manually determined methods would work well for a wearable augmented reality system, since the user's motion is not predictable and manually selecting control points is a tedious and distracting task.

6.2.2 Brute Force

The brute force approach is an exhaustive search on all parameters in the perspective transformation matrix and match one frame against the next. The search can be driven by a greedy algorithm like a least-squares error comparison of the frame using a proposed estimated transformation versus the target frame. Additional constraints can be applied to narrow the search. For instance there can be geometric considerations (e.g. setting limits on perspective distortion, scaling, rotation, scaling, etc) and Gaussian pyramids to narrow the search from a global to local scale. Despite these measures, the brute force approach is a computationally time consuming process, and not suitable for real-time use on a wearable platform.

6.2.3 Feature Tracking

The global motion between consecutive frames to mosaic can be estimated by matching corresponding local features between frames. This assumes that unique, distinctive features are present in both consecutive frames. Tracking the positions of four features from one image frame to the next gives the eight points needed to solve for a perspective transformation. Distinct and small features like corners, block regions, and templates can be found by a variety of efficient methods [27] [198] [205] [260] [217] [80] [131].

Although simple and fast, feature-detection schemes suffer from a number of drawbacks. They can be thwarted by the absence of features in a scene, such as a uniformly coloured or oversaturated wall or floor. Noise or texture, like vegetation and wallpaper, could introduce false features. Automatically and individually detected features might not correspond to geometrically constrained planar objects (e.g. a set of corners might be found on multiple objects like a moving person, a table, and the floor).

6.2.4 Frequency Domain

As detailed in Chapter 5.2.2 (page 85), global motion can be found using frequency domain methods like Fast Fourier Transform (FFT) phase correlation. An alternative to using the FFT is the wavelet transform such as the registration method in [96].

The benefits include simple computation and robustness against local motion. Some noise can be filtered (e.g. low-pass and high-pass) conveniently in the frequency domain. A distinguishing characteristic of all of these methods is the use of the entire image to form an estimate, whereas feature-based approaches usually consider local neighbourhoods.

On the other hand, Chapter 5.2.2's FFT phase correlation can only estimate translation and rotation. Rotation estimation in practice is very limited. Phase correlation also requires a significant amount of overlap (50 percent) between frames. Computing the FFT does require more memory resources than block based schemes. Davis [60] presents a robust FFT method combined with iterative search that derives a full perspective transformation, although rotation is still limited to a maximum of 45 degrees. Instead of matching two consecutive images, Davis matches a video frame against the current composite of images. This increases the potential overlap area to register with, but increases the memory requirements for the algorithm. Badra et al [12] use another frequency domain scheme (Zernike moments) to compute motion estimates efficiently, with less overlap. This scheme can robustly deal with large changes in rotation and translation and some zoom, but does not account for perspective distortion parameters.

6.2.5 Iterative Search

Iterative approaches to mosaicing try to trace the path of every pixel between reference frames and summarize these individual local motions as a global perspective transformation. Optical flow is a classic way to trace pixel motion [95]. Variations of optical flow use different models to solve and optimize the flow equations, such as Shum and Szeliski's gradient descent scheme [219] and Peleg et. al's image strip alignment [109] [183]. Mann and Picard [153] iterate an estimated versus an exact model of pixel motion derived from optical flow.

These mosaicing methods often use phase correlation or block matching to provide an initial translational estimate and bypass any local minima, and then use iteration to refine the estimate and discover the complete perspective transformation. Also their creators report robust results against noise and in-scene motion. In addition, 3-D and depth information can be derived from local pixel motion information [167] [234]. Computation effort for each pixel may be an issue, although methods like Mann's report speeds of 5-10 frames per second [152]. Techniques like optical flow assume consecutive frames have as little motion as possible (i.e. maximum overlap, or a high video frame capture rate) because the equations estimating the pixel motion are derived from an infinitessimal sub-pixel grid. However, using phase correlation to give an initial translation "boost" tends to ensure the frames are closely aligned.

Mann and Picard's method has been presented as a "repeated multiscale estimate, relate, and resample approach" for mosaicing [151]. However, the method in practice applies some frequency-based techniques besides using a phase correlation to initialize the iterative search with a translation. For each pair of images, a pure translation is first assumed, and obtained by phase correlation. If the MSE between images from translation has improved, the translation is kept. Then the frames are assumed to be rotated/zoomed. The rotation/zoom transformation is estimated by equation 5.11 in section 5.2.2. If the MSE between images is improved, the transformation is kept. Finally, the frames are assumed to differ only by a perspective chirp in the x-direction. This is treated as a camera "pan", thus the images are transformed into cylindrical coordinates, and the x-chirp is found by phase correlation in the cylindrical space. If the MSE improves, the transformation is kept. The same is applied for a vertical pan (y-chirp). In conclusion, although it depends on iteration to finalize the exact transformation, Mann and Picard's initialization using frequency-based methods suggest the method is more of a hybrid of iterative and frequency techniques.

6.3 An Initial Investigation into Feature Tracking

Early in my research, I investigated several approaches to feature tracking for mosaicing. Feature point selection determines the reference points to use for motion estimation. In general, I estimated local edge activity, and candidate feature points were moved towards regions with higher estimated edge activity. The goal of estimating the local edge activity was to place feature points in areas where a motion estimator can obtain an accurate motion estimate (as opposed to flat uniform regions from which no motion could be inferred). The actual edge activity estimator was based on a local Laplacian, i.e. a subtraction of the Gaussian low-pass information from the original local pixel values. The edge estimator algorithm can be described as follows:

- Obtain a local Laplacian around the current feature point (at the beginning, the "current" point defaults at the corner of the image)
- Sum the absolute differences from the local Laplacian to get a local edge activity estimate for that point
- Repeat steps 1 and 2 for blocks neighbouring the feature point
- Move the feature point to the block with the highest local edge activity estimate
- Stop if the corner point no longer moves, or after a set number of iterations

Summing the absolute differences of the local Laplacian was not a true estimate of edge activity, however. Rather it was an estimate of local non-uniformity. Also, this scheme did not guarantee the final corner points rest on edges - only that they are in regions with some kind of edge activity. However, this method was simple to implement, and fast to compute. The block sizes used were 16, 8, and 4 pixels for the original and two levels of Gaussian pyramid. In a compromise between a precise estimate and speed, neighbouring blocks were selected at distances of half the current block size away (i.e. 8, 4, and 2 pixels away).

Two of the most successful methods I examined and their results are presented in the next two sections below. The other techniques were variations of the first method (translational block matching).

Translational Block Matching

Block matching compares the pixel values in a block around each feature point in the current frame with blocks around the feature point in the reference frame (see Figure 6.2 (a)). In order to handle non-translational motions from a camera (e.g. rotation), block matching required a very small motion between frames. Also, block matching required the local neighbourhood to have non-uniformity (e.g. edges) in order to distinguish any motion.

The goal of translational block matching is to estimate the overall translation between the current and original image frames. The translational matching scheme used was a variation of the basic block matching method as described by Clarke in [42]. Instead of block matching each feature point individually, the algorithm examined four feature points together. Thus, each relative direction with respect to the corner points was examined in turn and the total summed absolute difference of the blocks around each of the corner points in that direction was taken. The total summed absolute difference in each direction with respect to the corner points were compared and the direction with the lowest difference was selected. This ensures that there was an overall agreement in block matching predictions, which provided the basis of an overall translation. All four feature points moved equally in the same direction as a result.

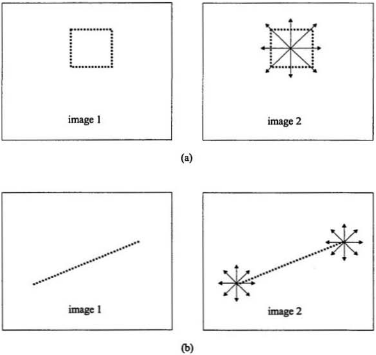


Figure 6.2: Block matching and line matching

(a) estimating motion using block matching between image 1 (reference frame) and image 2 (b) estimating motion using line matching between image 1 (Reference frame) and image 2

Line Matching

Whereas block matching attempted to find the best pixel match in a block, line matching attempted to find the best line match in a set of lines. A reference line in the current image frame was defined by two end points, and a set of lines for comparison was generated by moving one end point within a predefined block (see Figure 6.2 (b)). An initial translation before line matching was important because line matching did not model translation particularly well (fixing an endpoint and moving another resembled stretching, shearing, and possibly rotation operations). The actual algorithm used was as follows:

1. Perform a translational block match and translate the current image
2. For each edge of the current image (defined by the corner points)
    (a) Define a reference line in the current image, defined by the current edge
    (b) Define a set of lines, by fixing an end point, and varying the other within 2 pixels
    (c) For each line in the comparison set, compute the mean summed absolute difference between the pixels in the reference line (in the current image) and the pixels in the current line (in the reference image)
    (d) Select the line with the least mean summed absolute difference, and update corner points to match this line

The order in which edges were visited affected the results. Furthermore, certain images could have more useful matching information in areas outside the edges (e.g. an image could have blank edge regions, which would cause line matching to fail). Thus, as a refinement, the line matching was applied iteratively with four different patterns: the first pattern was a clockwise traversal around the image, the second was counterclockwise, the third was a crisscross: upper left to lower right corner then upper right to lower left corner, while the last was the crisscross done backwards. The use of complementary patterns helped verify results, and the variety of these patterns ensured the scheme was not restricted to specific areas of the image. At any point during the iteration if the summed difference was estimated to become worse, the iterating was stopped.

When applied with a fast line generation algorithm, such as Bressenham's run-slice line drawing algorithm [2], the line matching scheme was computationally fast. Line matching had the advantage of comparing longer regions between frames, whereas block matching was focused on concentrated blocks. However, the compared regions were only 1 pixel in thickness, thus the local context around the line was lost. To compensate for this, the line matching algorithm was combined with a translational block match. The translational block match was done initially to provide a translational estimate. Then the line matching algorithm was applied to provide a perspective correction.

6.3.1 Feature Tracking Results

Figures 6.3, 6.4, 6.5, and 6.6 show the results of applying translational block matching and line matching. The first two were image mosaics generated from 320 pixel by 240 pixel sequences from Steve Mann's web site [146]. The "Alan Alda" sequence in Figure 6.3 was produced by a camera pan, with some tilt. The "Claire" sequence in Figure 6.4 consisted mostly of a camera panning left to right to left with varying tilt to revisit parts of the image. The translational block matching and line matching algorithms were compared to the results reported at that site using Mann and Picard's algorithm. The image mosaics in Figure 6.5 and Figure 6.6 were generated from 320 pixel by 240 pixel sequences taken by the author. The "Hallway" sequence in Figure 6.5 was a camera pan similar to "Alan Alda", but taken farther away from the scene. The "Laptop" sequence in Figure 6.6 was taken from a head mounted camera, undergoing motions analogous to the "Claire" sequence (pan, tilt, revisiting) along

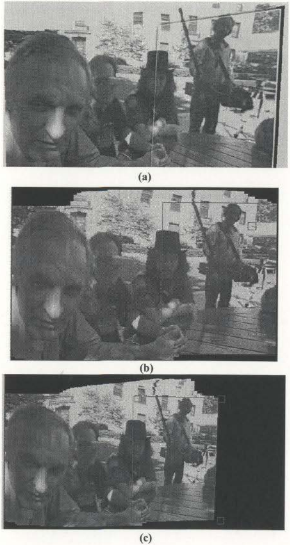
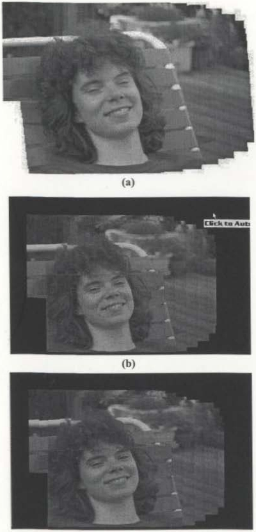


Figure 6.3: "Alan Alda" Sequence

(a) Image mosaic using Mann and Picard's Algorithm, (b) Image mosaic using 32x32 Translational Block Matching, (c) Image mosaic using 32x32 Translational Block Matching with Line Matching



(a)

(b)

(c)

Figure 6.4: "Claire" Sequence

(a) Image mosaic using Mann and Picard's Algorithm, (b) Image mosaic using 32x32 Translational Block Matching, (c) Image mosaic using 32x32 Translational Block Matching with Line Matching

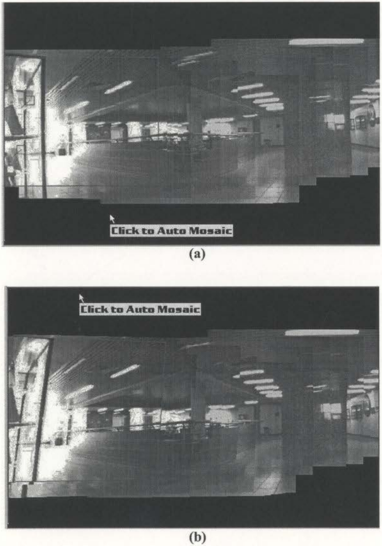


Figure 6.5: "Hallway" Sequence

(a) Image mosaic using 32x32 Translational Block Matching, (b) Image mosaic using 32x32 Translational Block Matching with Line Matching

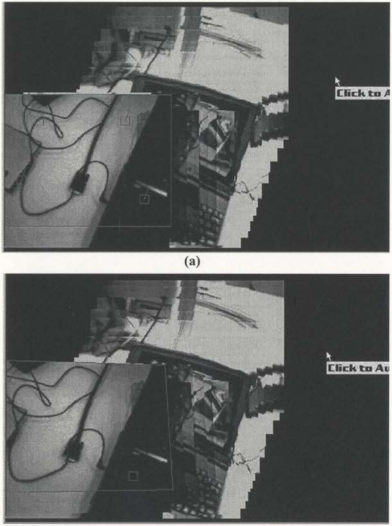


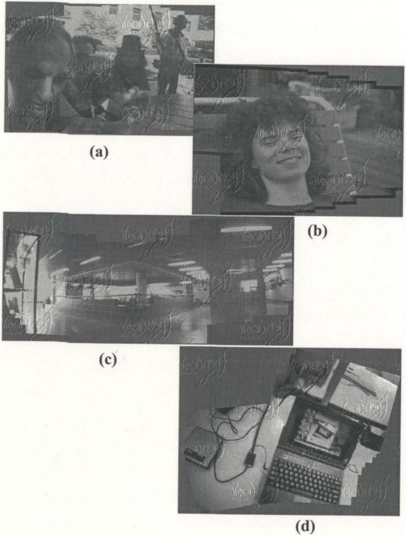
Figure 6.6: "Laptop" Sequence

(a) Image mosaic using 32x32 Translational Block Matching, (b) Image mosaic using 32x32 Translational Block Matching with Line Matching with rotation, translation, and some zoom. The translational block matcher and line matcher algorithms operated with a 32x32 pixel window size.

The first two results show that the translational block matching and line matching algorithms were comparable to Mann and Picard's algorithms, but errors were still apparent. The latter two cases, Figure 6.5 and Figure 6.6, varied in results. Figure 6.5 produced adequate results, but image errors were visible. Figure 6.6 produced very noticeable errors, due to the rotation and zoom. Despite their failure to provide exact orientation information for each frame, the algorithms were able to obtain general position estimates, which resulted in "choppy" but somewhat accurate image mosaics.

The average speed performance of the translational block matcher and line matcher was in the same order of magnitude, within 10 seconds per frame (including rendering time) on a Pentium 166 with 32 Mb of RAM, Microsoft Direct X version 3.0, 2 Mb of video RAM. This compared favourably with Mann and Picard's 6 seconds per iteration on a 3-4 level Gaussian pyramid on a HP735 [148] (they report using 2-3 iterations per level, so the total speed estimation would be 36-72 seconds per frame). However, 10 seconds per frame was still a long way from real-time performance. On a later test with a Pentium Pro 200 workstation, with 64 Mb of RAM, Direct X version 5.0, and 4 Mb of video RAM, the line matching algorithm ran at 3.4 seconds per frame for the "Alan Alda" sequence.

For another comparison, the "Alan Alda", "Claire", "Hallway", and "Laptop" sequences were run through a demonstration version of the commercial image mosaicing program, VideoBrush Panorama [52], which uses Peleg and Herman's strip-based algorithm [183]. The results are presented in Figure 6.7 (note that the demonstration version of the software inserts a watermark into the background). Speed measurement on the same Pentium Pro 200 workstation configuration as described earlier gives 13 seconds for 30 frames in "Alan Alda", or 2.3 frames per second (0.43 seconds per frame), 19 seconds for 17 frames in "Hallway", or 0.89 frame per second (1.11 seconds per frame), 19 seconds for 15 frames in "Claire", or 0.78 frame per second (1.27 sec-



(a)

(b)

(c)

(d)

Figure 6.7: Image Mosaics Produced by VideoBrush Panorama

(a) "Alan Alda" sequence, (b) "Claire" sequence, (c) "Hallway" sequence, (d) "Laptop" sequence onds per frame), and 24 seconds for 30 frames in "Laptop", or 1.25 frames per second (0.8 seconds per frame). Although the speed varies depending on the "difficulty" of the image (i.e. a short range pan like in "Alan Alda" gave faster results than the pan, tilt, rotate, and zooms in "laptop"), the overall speed was superior, and the quality of the final image mosaics match Mann and Picard's results.

6.3.2 Constrained Feature Matching: A Hybrid Approach

As observed in Section 6.2.3, a major shortcoming of any feature based approach, including the ones examined here, is that any error at any feature point can lead to disastrous estimates. Thus, noise, sudden foreground motion, flat scenery, or anything else that can fool the local motion estimate, can produce a poor perspective transformation result. Each feature is being treated independently rather than as a set of points constrained together. Nonetheless, this investigation has inspired the development of a new hybrid algorithm that addresses these shortcomings with claims of speed and robustness benefits [203]. The new hybrid mosaicing scheme combines the strength of block matching to derive a translation estimate and applies iterative searching over a grid of local candidate regions. The algorithm is summarized as follows:

First a translational estimate is computed by performing a full image block match over an image pyramid. The block matching process is similar to that described in Section 6.3, but uses the overlapping region between a reference image and displaced image over each pyramid level rather than a local neighbourhood. The block matcher computes mean differences between overlapping image regions for candidate displacements ranging up to half the image size. Estimates at each level of the image pyramid seed the search at the next pyramid level for further refinement. Lighting variations (e.g. from automatic gain control) are compensated by computing the mean greylevel of the overlapping region at each pyramid level and using this to adjust the pixel values for subsequent levels. Besides compensating for global lighting effects, the resulting translational estimate gives good initial placement for an iterative search relying on local features.

The second part of this method places a grid of "windows" to compute candidate transformations and to obtain local pixel information from the images to mosaic. The sampling windows combined together must span an image frame, and are constrained to not overlap. Each window's extents are dictated by an associated candidate perspective transformation. The goal of the second part of the method is to compute the optimum transformations of these windows to match the two input frames. This is done by applying a simplex optimization search, using the transformation parameters as the variables to solve for, and using the sum of absolute pixel differences between corresponding windows in the image frames as the driving function.

Basically as the optimization progresses, the shapes and positions of these search windows in the input frames slowly change, as if a projective transformation is being applied to them, until the pixels within one image's window match up closely to those in the warped window in the other image frame. The perspective motion estimate between image frames are then derived from the optimum transforms found in the windows.

The use of simplex optimization puts this method more in common with iterative search schemes, but the use of a moving grid of windows to examine local pixel disparity harkens back to the earlier attempts in feature matching. Robinson and Cheng [203] go into further detail and discussion about this new hybrid scheme.

6.4 An Evaluation of Mosaicing Methods

The mosaicing methods listed here are only a sample of what is available in the literature. Behind each method are claims of speed and robustness, but no quantitative comparison against other techniques. I selected a diverse sampling of mosaicing algorithms to evaluate, and measure and compare their performance against wearable augmented reality criteria. The selected methods are:

- **Mann and Picard's Video Orbits algorithm [153]**

  This iterative frequency domain method has been used to support wearable augmented reality work such as texture mapping annotations over advertising [152]. In the subsequent sections, I will refer to this algorithm as "Mann". Version 1.08 of the algorithm is used in this evaluation, which is available online at [146].

- **Davis' perspective FFT algorithm [60]**

  This frequency domain method goes beyond the phase correlation algorithm I used in Chapter 5.2.2 by finding all the perspective transformation parameters. It has additional robustness by matching against a global mosaic rather than individual frames. In the subsequent sections, I will refer to this algorithm as "Davis".

- **Videobrush**

  Unlike all the other methods examined here, Videobrush is a commercial mosaicing product. It is based on the iterative methods described in [109] [183]. In the subsequent sections, I will refer to this algorithm as "Videobrush".

- **FFT Phase Correlation**

  Although it only computes translational estimates, the FFT phase correlation estimator used in AugR in Chapter 5 is included, for comparison purposes. In the subsequent sections, I will refer to this method as "FFT".

- **The Hybrid Approach**

  The technique described in section 6.3.2 is a new technique that primarily relies on perspective constraint driven optimization. In the subsequent sections, I will refer to this method as "Robinson".

6.4.1 Criteria

There are three basic evaluation criteria important for a registration system on a wearable platform:

- **speed**

  The ideal algorithm should process image frames at real time speeds for any virtual overlays to be usable on a wearable computer. Although the processing could be given to a powerful server and the results be delivered to the wearable, this requires a reliable and responsive network connection. It is preferable that all processing is local on the wearable, minimizing any lag time, and making the system independent of any supporting infrastructure.

- **accuracy**

  Balancing speed is the need for an accurate world model inferred by the mosaicing algorithm. Accuracy determines how well the algorithm's model matches against the real world. For instance, an annotation drawn with AugR in Chapter 5 should remain stable in the scene, and should not wander as the user moves around.

- **reliability**

  Somewhat tied to accuracy, reliability refers to how often a mosaicing algorithm can consistently produce estimates below an error threshold in any situation. An algorithm could be very accurate but unreliable, for example, if it can give excellent estimates for only a constrained set of images and environmental conditions. On the other hand, a reliable but somewhat inaccurate algorithm could produce passable or semi-passable results for all cases.

Unlike desktop mosaicing applications like remote sensing, speed is a significant criterion because the mosaicing algorithm impacts the wearable user interface's responsiveness. Accuracy is also important, notably in scenarios demanding precision (e.g. surgery), but a user may be more forgiving while in motion using a wearable system [36]. Reliability's importance varies with application. A system could be tuned for a guaranteed level of confidence for a specialized environment (e.g. a training room, a lab), whereas a user may be more forgiving to a general-purpose system (e.g. everyday use in the real world).

The existing literature focuses on presenting final mosaics from long series of image frames, and all algorithms produce seamless mosaics. Rather than repeat these efforts in demonstrating overall effectiveness over multiple frames, I concentrated on testing the algorithms systematically over specific perspective transformations (translation, rotation, zoom, and perspective distortion) using image pairs. This evaluation focuses on robustness and accuracy versus motion. I do not consider measuring the impact of noise and lighting variation, which can be a worthwhile study, but these can be compensated against (as described in section 6.1.3).

6.4.2 Procedure

The evaluation procedure is as follows, and illustrated in Figure 6.8:

1. A data set of image pairs is extracted from a series of photographs. For each photograph, a series of known perspective transformations are applied to extract image pairs. The transformations themselves are small, moderate, and extreme translations, zooms, rotations, and perspective distortions. The transformation parameters are chosen to represent different extrema of motion, but are bounded to ensure all the mosaicing algorithms have a fair chance of succeeding.

2. The set of image pairs are run against the evaluated algorithms. All the methods are automated except for Videobrush, which requires manual use, being a commercial package.

3. The time taken to estimate and construct mosaics is measured for all methods. Note that although Videobrush's time is measured, the measurement is partially

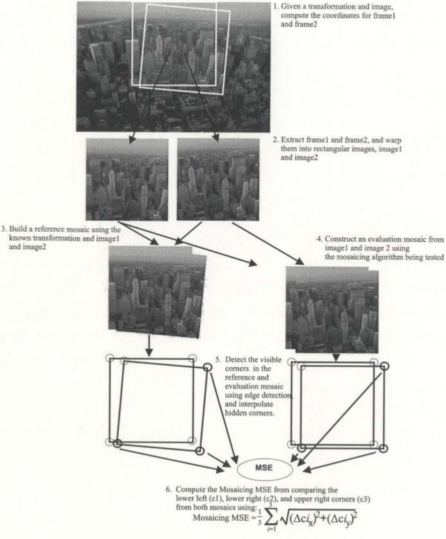
limited by human speed rather than a computer's, because the Videobrush software must be manually operated.

4. The accuracy is measured for each mosaiced image pair by computing the mean squared error (MSE) of three of the visible corners of the transformed frame against the real corner positions from the extracted frames in step 1. Three corners are only available for comparison because of the overlap between the image pairs. Visible corner positions rather than perspective matrices are compared because VideoBrush, being a commercial package, does not return transformation parameters. Reliability is derived from an analysis of the mean squared errors, and will be discussed in Section 6.5

Other approaches to evaluating mosaicing and augmented reality registration in general include [117], [113], [118], [216], [92], and [166], but they often take advantage of "inside information" provided by their systems software and hardware to obtain accurate measures of performance. For example, one could use the transformation matrix output from an algorithm or magnetic trackers to get performance measurement data.

I took a more neutral experimental approach, treating the evaluated systems as "black boxes". Thus, all methods are treated to take in pictures, and return mosaiced pictures. This provided a less rich dataset less inclined to precise quantitative analysis than taking a more traditional procedure, and required me to resort to image processing for evaluation. But this experimental regime is not partial to open-sourced software applications or closed commercial products.

The reference transformations were deliberately selected to provide a balance between a variety of cases, and a reasonable set of cases that could be processed by all the tested methods. As seen in Figure 6.9, all transformations include a translational shift to ensure a similar "upper-left" to "lower-right" rectangular mosaic. The reference mosaics were all checked against the corner-finder to guarantee a set of reference corner positions in all cases. Five basic perspective transformations, rotation,



1. Given a transformation and image, compute the coordinates for frame1 and frame2

2. Extract frame1 and frame2, and warp them into rectangular images, image1 and image2

3. Build a reference mosaic using the known transformation and image1 and image2

4. Construct an evaluation mosaic from image1 and image 2 using the mosaicing algorithm being tested

5. Detect the visible corners in the reference and evaluation mosaic using edge detection, and interpolate hidden corners.

6. Compute the Mosaicing MSE from comparing the lower left (c1), lower right (c2), and upper right corners (c3) from both mosaics using:

$$\text{Mosaicing MSE} = \frac{1}{3}\sum_{i=1}^{3}\sqrt{(\Delta c i_x)^2 + (\Delta c i_y)^2}$$

Figure 6.8: Steps to Evaluate a Mosaicing Algorithm

translation, shear, chirp (perspective distortion), and scale, are varied at three levels, denoted as "close", "far", and "extreme". The close and far cases attempt to test response to small and moderate motion, whereas the extreme cases seek to push the limits, but hopefully still produce intelligible results. Three rather than four corner points are computed in the mean squared error, to minimize confounding the data with any interpolation errors from computing a hidden forth corner in the reference mosaics. The same corner finding process is applied to generate the reference corners and the evaluated mosaics' corners.

6.4.3 Data Set

The photographs chosen to form the data set are a series of high quality greyscale JPEG pictures over 600x390 in size (see Figure 6.10). The extracted image pairs are 256x256 in resolution and saved in BMP or PGM format (depending on the algorithm used). The selected photos are from the Corel Draw 6 clip art CD, and represent a diverse set of real world imagery that a wearable computer user would encounter. Indoor spaces, cityscapes, outdoor wilderness, people in action, individuals, and crowds are depicted. The same set of transformations illustrated in Figure 6.9 were applied to all the images, producing a data set of 135 image pairs (9 images, 5 transformations, 3 transformation magnitudes). Given the five methods being evaluated (Mann, Davis, Videobrush, FFT, and Robinson), 675 mosaics were generated and examined.

6.5 Results and Discussion

6.5.1 Overall Accuracy and Reliability

Figure 6.11 summarizes the accuracy and reliability performance of the tested methods. The graph plots how many mosaics had mean squared errors (MSEs) below a MSE threshold from 0 to 100. So, a point such as (4,100) means that particular

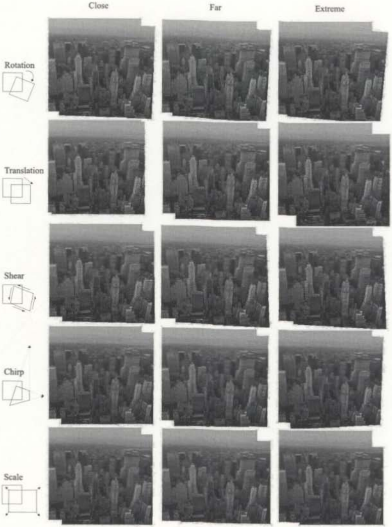


Figure 6.9: Reference Mosaics for all Evaluated Transformations in City Image Sequence

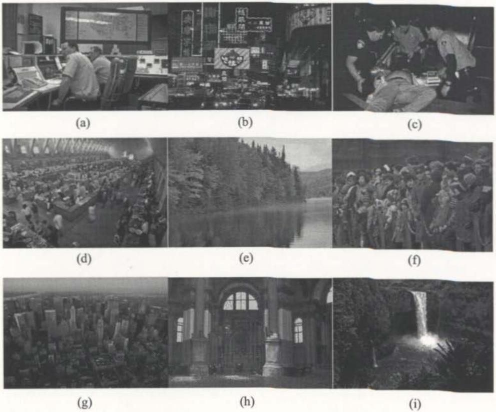


Figure 6.10: Image Set Used for Mosaicing Evaluation

Images from the Corel Draw 6 Clip Art CD ROM: (a) Police, (b) Neon, (c) Medical, (d) Market, (e) Forest, (f) Crowd, (g) City, (h) Cathedral, (i) Waterfall algorithm created 100 mosaics with a MSE below or equal to 4. Each curve shows a different mosaicing method. A lower MSE score denotes a close match between an evaluated mosaic against its corresponding reference mosaic. Thus, curves with a near vertical slope near the y-axis illustrate very accurate algorithms. As the MSE threshold increases, the curves reach a plateau, since eventually all mosaic cases, good and bad, will be included with a high enough MSE. So, methods that plateau at some point offer a certain level of reliability (a consistent percentage of good or bad mosaics). And mosaicing methods with curves that reach the maximum plateau closer to the y-axis indicate an algorithm with strong accuracy and reliability.

With these considerations in mind, the Robinson and Mann methods have the strongest slopes for small MSE thresholds and reach a plateau earlier than the other methods. Thus, both offer strong accuracy results at a guaranteed level of reliability. However, the Robinson method has a significantly higher plateau (which is eventually reached by the other methods for larger MSE thresholds) than Mann, suggesting stronger reliability performance. Mann's curve is still slowly growing over MSE thresholds, and never reaches the plateau achieved by the other methods at the end of graph, which suggests the method has a bi-modal performance: good results given good cases, but catastrophic results otherwise.

The Davis and Videobrush methods share somewhat similar performance curves, with less vertical slopes early on than either Mann or Robinson, but eventually attain Robinson's plateau. Thus they offer somewhat less overall accuracy but eventually achieve a consistent level of reliability, with Davis being slower to achieve this level than Videobrush. The FFT method, being a translation-only mosaicing method, fared the most poorly, but does achieve a higher plateau than Mann in the graph.

It should be noted that for higher MSE thresholds, it is harder to compare the "goodness" of any algorithm's mosaicing results. Larger MSE thresholds suggest larger displacement errors, but the comparison becomes an analysis of "bad" and "worse" results, both of which can correspond to equally unintelligible distorted im-

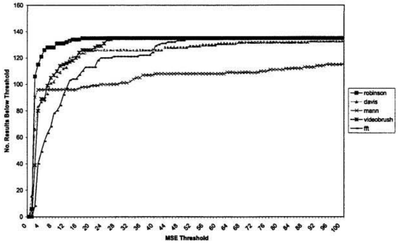


Figure 6.11: Total Mosaicing Results Over MSE Thresholds

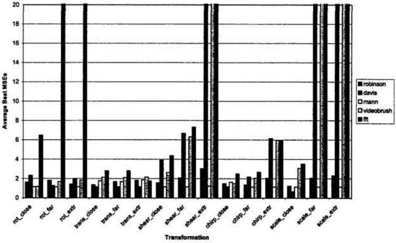


Figure 6.12: Average Mosaicing MSE under a MSE Threshold of 8

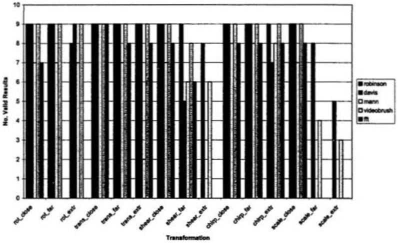


Figure 6.13: Number of Mosaicing Results under a MSE Threshold of 8

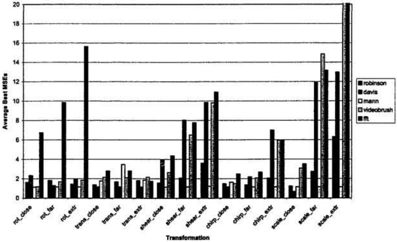


Figure 6.14: Average Mosaicing MSE under a MSE Threshold of 16

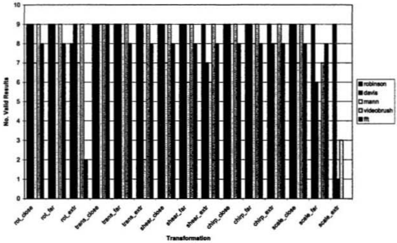


Figure 6.15: Number of Mosaicing Results under a MSE Threshold of 16

agery. In Figure 6.11, all of the methods slope upwards consistently within a MSE threshold range of 0 to 8, thus it should be said that all the evaluated methods are capable of creating good mosaics and none of them fail completely during my evaluation.

Figures 6.12, 6.13, 6.14, 6.15 take a deeper look at the tradeoffs between accuracy and reliability by examining average MSEs and counts under specific thresholds. Figure 6.11 shows the majority of methods with a sharp incline and beginning their approach to plateau in the 8 to 16 MSE range. Figures 6.13 and 6.12 takes a snapshot of the number of mosaics under MSE 8 for different transformations, and the average MSEs of those mosaics under MSE 8 over different transformations respectively. Figures 6.13 and 6.12 do the same thing respectively for a MSE threshold of 16. Cases with no mosaics under the threshold are denoted as columns going above MSE 20 in figures 6.12 and 6.14.

Comparing 6.13 to 6.15, there is a sharp rise or appearance of columns in the charts, indicating a marked increase of mosaics under the MSE threshold of 16 rather than 8 for methods like Mann and FFT. Other methods like Robinson roughly maintain the same profile, indicating they reached their plateau. The high column values indicate a majority of mosaics falling under the MSE threshold, further confirming the case for consistently strong mosaicing results for such methods.

Comparing 6.12 to 6.14, there is a similar pattern, with a sharp rise of average MSEs in cases with higher inaccuracies. Methods like Mann and Robinson, illustrate their good accuracies within the thresholds by maintaining lower MSE averages despite a change in threshold.

6.5.2 Performance Per Transformation and Picture Set, and Failure Modes

The next set of charts, Figures 6.22, 6.18, 6.16, 6.20, 6.24, show the performance of the algorithms over specific transformations and pictures. In essence, each chart gives a algorithm-specific "profile", indicating which transformations and pictures the algorithm can handle best and worst. The charts plot the MSE result for each transform over every picture set. Areas of the charts with low bars indicate strong accuracy results, whereas high bars show erroneous cases. A high frequency of high bars suggests consistent problems, while a high frequency of low bars suggests good accuracy and good reliability. The x-axis uses abbreviations of the transformations, e.g. "rot_far" means "rotation far".

Figure 6.16 shows the profile of the Mann mosaicing method. The results reflect the hard failure/success case shown in Mann's performance curve in Figure 6.11. For almost all the transformations except extreme translation, the Mann method has a number of very successful mosaics around a MSE of 1 or lower. But the method gives very high MSEs for specific image sets. This suggests the Mann method is very good, and possibly superior to, other methods if given images similar to its successes shown on the graph, or if its configuration parameters are well-tuned for the desired application. However, unforseen cases can result in unpredictable behaviour, whereas other methods offer more graceful failure modes (e.g. the MSE for the Davis method rises with increasing shear).

In figure 6.17, a few mosaics generated by the Mann method are shown for high MSEs. The topmost illustrates one example of the Crowd image set, where for every transformation except for close translation, there is a consistently high MSE. There appears a tendency in Mann to apply a chirp transform, which appears to overwhelm the final mosaics in the high MSE cases. The other two images show this bias to chirping noticeably, but less so. The chirping bias may create false optimization paths for an iterative scheme like Mann's in certain images.

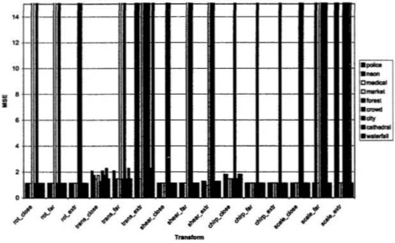


Figure 6.16: Mosaicing MSE Profile for the Mann method

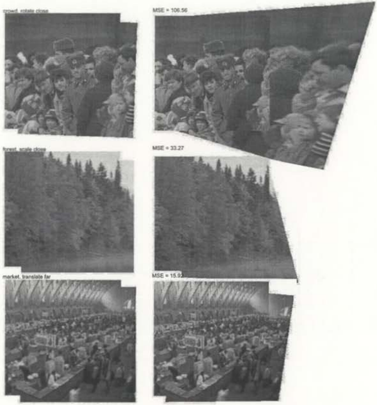


Figure 6.17: Example Mosaicing Failures for the Mann method

Left images show reference mosaics, right images show evaluated mosaics from the Mann method

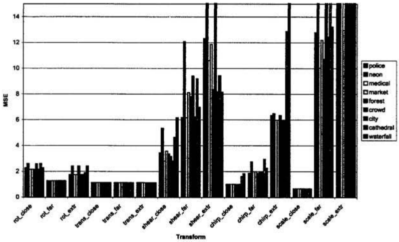


Figure 6.18: Mosaicing MSE Profile for the Davis method

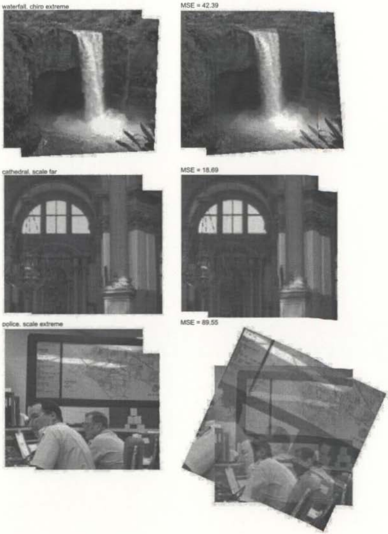


Figure 6.19: Example Mosaicing Failures for the Davis method

Left images show reference mosaics, right images show evaluated mosaics from the Davis method

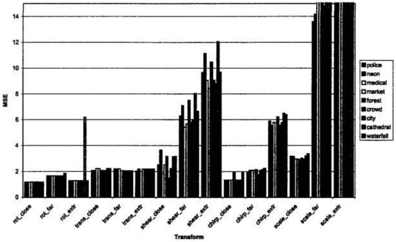


Figure 6.20: Mosaicing MSE Profile for the Videobrush method

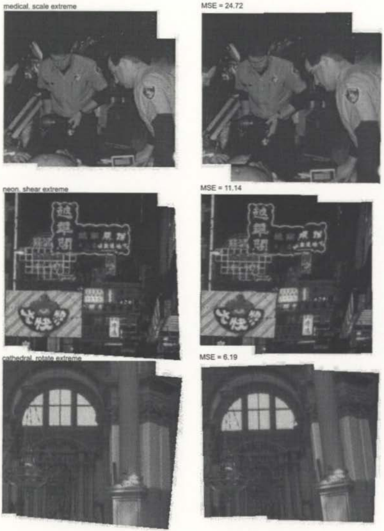


Figure 6.21: Example Mosaicing Failures for the Videobrush method

Left images show reference mosaics, right images show evaluated mosaics from the Videobrush method

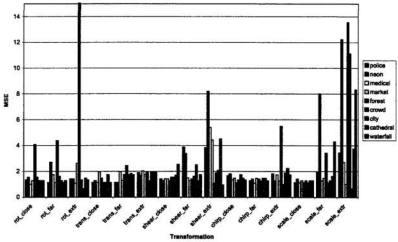


Figure 6.22: Mosaicing MSE Profile for the Robinson method

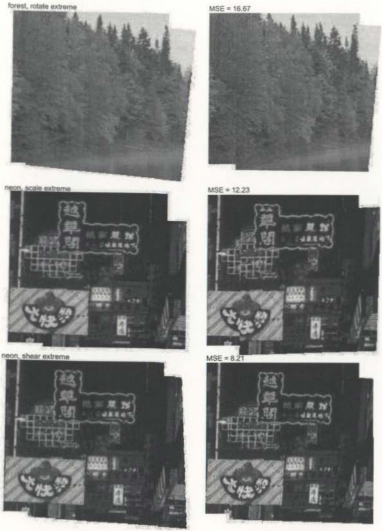


Figure 6.23: Example Mosaicing Failures for the Robinson method

Left images show reference mosaics, right images show evaluated mosaics from the Robinson method

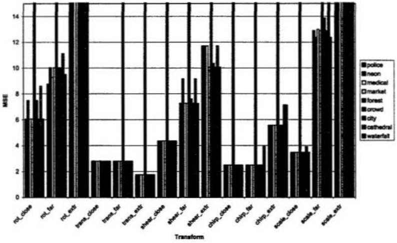


Figure 6.24: Mosaicing MSE Profile for the FFT method

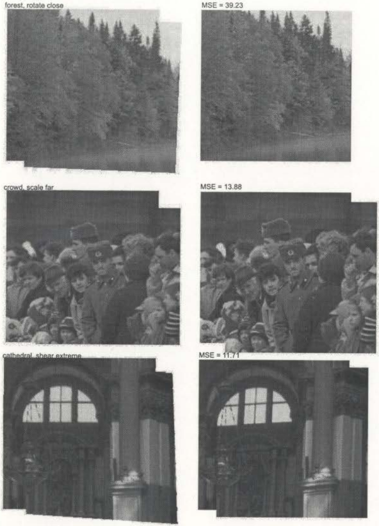


Figure 6.25: Example Mosaicing Failures for the FFT method

Left images show reference mosaics, right images show evaluated mosaics from the FFT method

Figure 6.18 shows the profile of the Davis mosaicing method. The Davis method offers low MSE results around 1 for many images and transformations, and some around a MSE of 2. Davis consistently has problems with the far and extreme shearing and scaling, as well as the extreme chirping transformations over all image sets. Thus, it can be said the Davis method performs consistently well for rotations, translations, and some moderate chirp and scaling, but does poorly over all image sets otherwise. These results are consistent with the use of phase correlation in the Davis method. Phase correlation provides a reliable translation estimate, and to a lesser extent, rotation (note that Davis has slightly higher MSEs in rotation) by phase correlating a polar-coordinate version of the image.

In figure 6.19, a few high MSE cases for the Davis method are shown. They all show error in the initial displacement. Once seeded with a poor initial displacement, the Davis method follows up with rotation and perspective effects, which it does somewhat well for the top two cases. Visually, they appear "correct", thanks to its post-estimation blending and refinements. However, the result does not match the transformation intended by the reference frame. In the bottom case, the two frames are different enough by a scaling factor to necessitate a good initial displacement to have any hope of proper alignment. Lacking a good initial displacement, Davis tries what it can to align the two images - it obtained a scaled version of the second frame, but incorrectly assumed a rotation.

Figure 6.20 shows the profile of the Videobrush mosaicing method. Videobrush's profile is similar to the results from Davis, with problems arising for far and extreme shearing and scaling, and extreme chirping. Also like Davis, its high MSE cases (see Figure 6.21) are due to errors in the initial displacement estimates. These results are not too surprising, considering that Videobrush was intended to be a commercial desktop imaging application, reading in panoramic background shots from still and fixed cameras. Such packages recommend a steady camera sweep at a largely static, distant scene, which usually would not have shearing or scaling effects. They apply blending techniques to produce visually appealing mosaics, which are suitable for desktop use, but their actual transformations do not correspond to the intended case.

Figure 6.22 shows the profile of the Robinson mosaicing method. Overall, the Robinson method offers consistent MSE results under 2 over most images and transformations. It has trouble with rotating and scaling the Forest image set, and with some extreme cases of shearing and scaling. Therefore, the Robinson method appears to perform consistently well for all transformations and image types except for rotating images like Forest, and extremely sheared and scaled pictures. This supports Robinson's strong reliability and accuracy curve in Figure 6.11.

Robinson's high MSE cases are shown in Figure 6.23. One striking difference when compared to other methods is the lower MSE values for these cases. Whereas Robinson's worst cases range in the 8 to 16 MSE range, others like Davis and Mann feature examples over a 80 MSE. This fits with the higher reliability observation from Robinson's performance curve in Figure 6.11. While iterative techniques are susceptible to false optimization paths, Robinson's iterative optimization is coupled with the performance of its local "window" grids, which are all tightly constrained to fit on the image frame and a cumulative perspective transformation. Thus, the multiple grids and their coupling offer some robustness even when in a failure mode.

Figure 6.24 shows the profile of the FFT mosaicing method. As expected, the FFT gives worse results than the other methods overall, with its best cases in translation, which it was designed to handle. Even there, translation MSEs are notably higher than the other techniques. Since the FFT phase correlation is a single-pass technique and processes the entire image frame to obtain a motion estimate, it lacks the refinement steps offered by other iterative methods (and often these methods just use a phase correlation to give a rough initial displacement estimate). Even the FFT's closest relation, the Davis method, fares better, since it also does a global transformation refinement against the current image mosaic. The FFT actually does better for extreme translation, which relates to the technique's global approach in computing correlations. Small displacements are harder to detect in phase correlation, since they may be considered as "noise", or not sufficiently global changes to warrant attention by the FFT.

Figure 6.25 illustrates some of the cited shortcomings of the FFT method. Images with a heavy semi-uniform texture like the Forest image set (topmost image), suggest an unchanging background to a global method like FFT, even when there is actual motion within the frame. The other cases show the FFT trying its best to replace the 6 other missing parameters of a perspective transformation with a 2-parameter displacement.

6.5.3 Speed

The next figure 6.26 show the average times to generate a mosaic from two images. The times are constant regardless of transformation, but there are significant order of magnitude differences in speed among the methods. The timing standard deviations over different transformations and methods are consistently around 10 percent for each of the averages. Timing measurements only include the time to estimate the perspective transformation, not the times for image loading, saving, and mosaic compositing.

Davis and Mann are the slowest, averaging 21 and 11 seconds respectively, on a Pentium II 400 computer. Davis is understandably slowest due to its Matlab implementation. Mann's method uses the C implementation of the VideoOrbits 1.0 code available online [146], although a recent version described in [152] suggest a more optimized and faster implementation.

Videobrush is after Davis and Mann in slowness, averaging 1.7 seconds. The slowness is somewhat influenced by the need for human intervention to control the Videobrush application and presentation of a desktop user interface. But it is still optimized as a commercial application, giving some speed compared to the other methods.

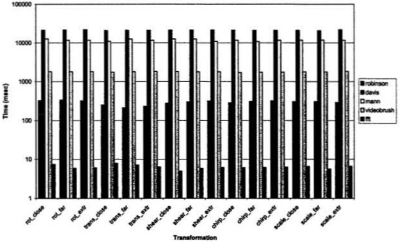


Figure 6.26: Average Mosaicing Times per Image Pair

Note that due to the order of magnitude differences in speed, a logarithmic time scale in milliseconds is used

Robinson is the second fastest method, balancing accurate and reliable results through optimization with a well optimized search path and robust initial block-based translation estimate. Clocking at 0.29 s on average, the method could provide optimum performance on a current generation computer platform.

The FFT gave the most dramatic speed result, giving an average 6.3 msec. The speed is largely influence by the implementation's dependence on the Intel Image Processing Library [105] to compute the FFT, which provides a machine-level optimized library for each separate Pentium CPU. Since the only significant steps of phase correlation technique involves a single pass with a fast-fourier transform and inverse transform, it is not surprising there is a high speed result given an optimized library. The slower frame rate times described in Chapters 4 and 5, where phase correlation is used in the Handel and AugR applications, would suggest slowdowns due to graphics subsystem rendering, networking, and other issues. Nonetheless, an implementation on a current generation platform with graphics acceleration hardware could yield a real-time wearable experience.

6.6 Conclusions

Figure 6.27 summarizes the discussion of mosaicing evaluation results. Given the three driving criteria of accuracy, speed, and reliability, the results show strong tendencies in each of these areas for different methods. Reliability is computed as a percentage of number of mosaics under a threshold over the maximum number of possible mosaics. The horizontal bars span left to right for each method, from a accuracy/timing/reliability measurement with a MSE Threshold of 8, and the same measurements for a MSE threshold of 16.

General purpose methods intended for desktop graphical use, such as Davis and Videobrush take a middle ground between accuracy, speed, and reliability. They do not need to be extremely accurate against a reference transformation, since they

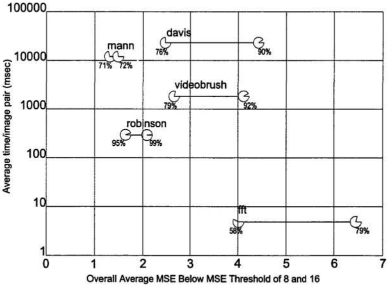


Figure 6.27: Mosaicing Accuracy Timing and Reliability

Each line goes from a accuracy,timing,reliability measurement at MSE Threshold 8 (left) to MSE Threshold 16 (right). Partially filled circles indicate percentage reliability (numerical value also shown as a percentage).

are primarily interested in nice looking image mosaics. Desktop applications are not held up by real-time constraints, thus speed is less of a priority. Consistency can be achieved at the expense of time, despite a lessened need for accuracy, and thus reliability is not as strong as other methods.

The FFT method provides incredible speeds, but at great cost to accuracy and reliability. Its lower-right location on the graph shows its speed advantage. At a MSE threshold of 16, the line spans much further right and still carries a low reliability of 79 percent, suggesting a higher frequency of inaccuracy than other methods.

Mann gives a tight line, on the left side of the graph. It provides excellent accuracy, but somewhat variable reliability. While other methods eventually achieve 90 percent or more reliability, Mann still stays around the 70 percent range. This may reflect more of its tight tuning for cases in real-world mosaicing on a head-mounted camera platform, rather than encompassing all possible imaging applications. Timing is a notable issue, but this may be better in Mann's current implementations.

Robinson shares a somewhat similar accuracy range with Mann, but with significantly greater reliability and better timing results. These suggest an excellent combination of attributes for a robust, general purpose application, including a wearable augmented reality system, but not limited to that case. In any case, these favourable results suggest further investigations with this new method.

In conclusion, I outlined the motivation for mosaicing as a engine for AR world modelling, and the fundamentals behind mosaicing schemes. After surveying the general categories of techniques, I have selected examples in each category for a comparative evaluation. Taking a "black box" experimental approach, the evaluation compared mosaicing accuracy, reliability, and speed using MSE measures, counts below MSE thresholds, and millisecond timings respectively. The FFT, Mann, and Robinson methods stand out with respect to these three criteria. The FFT demonstrates significant speed while sacrificing reliability and accuracy. Mann illustrates strong accuracy, whereas the new Robinson method presents a powerful balance of all three criteria.

Chapter 7

Conclusions

As outlined in Chapter 1, the goal of this research is to investigate how augmented reality interfaces can be achieved using imaging techniques. Image registration is used to form a world model, and simple visual detection methods are used to assess a user's intention and environmental awareness. From this, three areas are explored: the idea of Personal Context, the use of the mosaic as an interface, and a systematic comparison of world modelling methods. In this chapter, I will summarize and discuss the highlights from each of these explorations and propose recommendations for improvement and directions for future work.

7.1 Personal Context

Personal Context addresses the challenge of human-computer interaction, and follows the direction in ubiquitous, wearable, context-aware, and pervasive computing where computers take a more passive role than an "in your face" desktop experience, but are attentive to a user's needs and interests [225]. Like context-awareness in particular, personal context gauges the user's interests with current activity, and employs augmented reality to present timely and relevant information in the user's viewport of the environment. However, personal context takes a narrower focus than context awareness and AR, by concentrating its efforts on the user. Namely, the user's physical interactions with the environment are measured as potential cues for activity and intention, which triggers an augmented reality response if deemed sufficiently relevant to the current task and user's interests.

Two personal context applications, HANDEL and Footprint, were implemented and demonstrated in this work. Their common theme is to have the wearable computer pay attention to the user's hands and feet, and deduce whether assistance was needed in known tasks. For HANDEL and Footprint, the tasks were piano playing and dancing, respectively.

HANDEL and Footprint were tested successfully on an acoustic piano for a short musical piece and for a short waltz respectively. While they proved to be very comfortable to use, there are numerous improvements that can be made in the presentation of musical and dancing context, and only focused on very basic examples.

In summary, the user's attention on body parts for guidance is the basis of my demonstrations of personal context. This is a natural gesture in many tasks, thus a user can simply concentrate on the task as if the mobile computing device was not there in the first place. With only simple computer vision techniques, HANDEL and Footprint demonstrate such natural human-computer interaction in their specific application areas. And the use of XML in Footprint illustrates the potential of XML as a portable format to represent human activity in context for specialized wearable computer applications and general purpose desktop computers.

7.2 The Mosaic is the Interface

Personal context, as well as context awareness in general, relies on the computer to judge when it is appropriate to place timely, relevant information into an wearable augmented reality user interface. Telecollaboration systems, on the other hand, focus on computerized means to facilitate human-human interaction. I propose image mosaics as a computer-supported medium for such collaboration in complex tasks, where personal context could not play a significant role.

As discussed in Chapter 5, there exists a body of work in telecollaboration, including wearable-based collaboration. They conclude with interesting problems to be solved, notably a narrow field of view offered by an shaky, moving head mounted camera and a limited toolset for remote experts to explain their directions to a busy wearable computer user. 'The mosaic as the interface" answers these challenges, by leveraging the mosaicing algorithm's inherent AR registration ability to manifest the system's world model as live, growing big picture composite of the scene. The picture also becomes a drawing surface for the remote user to create annotations.

A traditional video window can also be drawn upon, but is really a visualization of one instant of time perceived by the cameraperson. Thus it is a naturally passive, one-way medium - used to great effect in other applications, like movies and television, but assumed a centralized authority geared towards broadcasting. Hence the cited difficulties with telepointing in Chapter 5: the remote expert in a telecollaboration system has to seize control of that instant normally managed by the field worker. On the other hand, approaches like freeze framing have the side effect of disrupting a common time reference between remote expert and field worker.

The mosaicing interface demonstrated by AugR presents a live world model of the scene, continuously organizing the scene seen by the camera. The spatial organization is not only a physical map of the scene, but also a temporal map. The latest composited image frame is like a traditional video window. However, the earlier composited images show a past history of the user's activity (mostly head motion). Thus a mosaic user interface not only offers a live view on the current scene, but also a unified spatial and temporal view of the scene and current activity. Furthermore, the remote expert's annotations on this visualization are registered against the field worker's frame of reference. These annotations are constantly shown in the field worker's time frame (due to AR registration) but try to maintain spatial constancy.

In comparison, a freeze-frame video conferencing window gives a limited spatial and outdated temporal view, and a rewindable movie sequence provides a temporal view without spatial relationships between movie frames. Annotations on neither offer any spatial constancy needed by a field worker who must exist in latest time frame.

Thus, AugR, as a demonstration telecollaboration system powered by mosaicing, steps up from videoconferencing based collaboration schemes with temporal/spatial visualization and annotations for the remote expert that are spatially and temporally relevant for the field worker. It uses "off-the-shelf" FFT phase correlation to mosaic, which can already be used to run traditional AR registration models. While the wearable user's experience is not radically changed from previous wearable AR systems, the remote expert has a greater toolset and view to enrich the human-human collaboration experience further.

7.3 Systematic Comparison of World Modelling Methods

Mosaicing has a central role to creating a world model for registration and visualization, as evidenced in Chapter 5 and discussed in Chapter 6. Even personal context systems can benefit from world modelling, such as in HANDEL in Chapter 4. Thus, it is important to evaluate mosaicing techniques against criteria suitable for a wearable AR experience.

The evaluation of mosaicing methods in Chapter 6 presents a systematic study into this need, focusing on the criteria of accuracy, speed, and reliability. By taking a neutral experimental approach, the evaluation examined MSE errors and timings for a wide range of possible transformations and images handled by mosaicing methods. Accuracy is related to MSE error against a reference, reliability is determined to be the percentage of test mosaics below a MSE error threshold, and speed is perspective transform estimation time for two successive image frames.

For a wearable AR experience, speed and accuracy are paramount compared to reliability for specific applications. Fast speed allows real-time responsiveness. High accuracy ensures correct placement of AR annotations. Reliability really depends on the current task. It is easier to tune an algorithm for specific tasks and restricted conditions, or perhaps use a suite of difference algorithms for different situations, rather than use a single method for every case. What can be said about reliability, however, is that it should offer a graceful failure mode: as conditions become more extreme it should be gradually apparent there is a problem. The system can then indicate a failure, possibly with a visual indication of the current uncertainty, or try to compensate (e.g. use another method).

The evaluation examined five mosaicing methods, including the phase correlation scheme used in AugR from Chapter 5. The methods varied in performance against different image types and transformations. Overall, the FFT phase correlation, Mann's VideoOrbits technique, and the new Robinson method stood out against the three criteria.

AugR's FFT phase correlation shows great speed, which makes it suitable for real-time use on a wearable platform. However, its restriction to translation estimation meant much more limited accuracy and reliability compared to all other evaluated methods. Other methods use phase correlation to provide a good initial estimate before applying more accurate refinements, and thus a future version of AugR would benefit more from other methods.

Mann's method demonstrates excellent accuracy but the tested implementation suffered reliability problems. It failed dramatically with certain image sets where other methods succeeded without any tuning. With a faster implementation, tuned for everyday wearable use, Mann's method is desirable for a wearable augmented reality platform, particularly for interactive applications using AR registration.

The Robinson method is a recently developed method, but its initial implementation shows great promise in speed, accuracy, and reliability. While slower than the

FFT by an order of magnitude, it was faster than all the other tested methods, and provided accuracy and reliability rivalling the best in those criteria. This would suggest the Robinson method is quite suitable for a general purpose platform, including a wearable augmented reality system.

The evaluation focused on pair-wise image estimation rather than entire video sequences, and used a limited selection of methods, although the dataset consisting of all permutations of pair-wise images over multiple videos, transformations, and algorithms would be daunting to generate and analyze. The use of MSEs against reference mosaics allows the use of closed commercial products in the testing, but yields less precise information than actual transformation matrix information available in open source code.

7.4 Future Directions

There are a number of future directions for this research, which are categorized as short term and long term directions. Short term directions address various limitations in this thesis, while the long term directions proposes new avenues and applications of the ideas introduced here.

7.4.1 Short Term Directions

While they illustrate the point behind personal context and mosaicing interfaces, the various prototypes implemented in Chapters 4 and 5 lack user data to refine their user experiences and evidence to strengthen the claims behind personal context and mosaicing as an interface. Besides critiquing and suggesting user-interface improvements (e.g. graphics, responsiveness, interaction), formal user studies need to examine HANDEL, Footprint, and AugR's benefits (or detriments) to learning piano music, improving dancing, and enhancing remote expert-field worker telecollaboration. HANDEL and Footprint can benefit from more complex and longer examples of music and dance. A study of AugR should also compare user performance against earlier investigations in video telecollaboration described in Chapter 5.

Since the human-human telecollaboration should be as responsive as face to face interaction, AugR's lag time and overall performance need to be assessed. Also, AugR's use of an interactive spatial/temporal mosaic is susceptible to information clutter. The remote user must consciously manage the placement and deletion of annotations, or otherwise cloud the field worker's view. A possible improvement would have a personal context system, tuned to the field worker's needs, filter out dated and irrelevant annotations automatically. The use of a mosaicing algorithm should be compared against other faster and more accurate registration schemes, like hardware tracking or hybrid hardware-algorithm methods, where mosaicing parameters could be derived.

With regards to the evaluation of mosaicing methods, while a considerable number of test cases were examined over various methods, more comprehensive evaluations are needed. The testing could benefit from a greater variety of imagery, including frames captured from live wearable camera footage. Comparing image pairs could be expanded to comparing long image sequences. It would be worthwhile to compare the results presented in the thesis against a conventional non-black box experiment. Exact mosaic corner positions can be computed from transformation parameters taken directly from the evaluated algorithms, and the same MSE evaluation scheme and analysis in Chapter 6 can be applied. Also, a more general study focused on AR modelling schemes, should introduce comparisons against non-mosaicing approaches, such as pure hardware tracking systems and 3-D motion estimation computer vision algorithms.

7.4.2 Long Term Directions

This research has treated personal context and mosaicing as two sides of wearable augmented reality user interfaces. Personal context focuses on the user to create user-computer interaction, whereas mosaicing uses the environment to generate user-user interaction. Future research could examine how both can complement each other in a single wearable user experience.

For example, mosaicing can provide a world model to stabilize annotations on world objects, and personal context can enable interaction between body-centered annotations with world-centered annotations. I did a preliminary investigation into this using a modified version of AugR, which followed the use of coloured fiducials and coloured objects in work like [40] [227]. Figure 7.1 shows a gloved hand that moved a virtual annotation created and registered with AugR. A colour detection algorithm identified the hand and its overlap with the annotation to allow the annotation to attach itself to the hand. Once attached, the annotation is no longer registered with AugR's mosaicing scheme, and relies on the colour tracker (thus, the annotation becomes part of the user's personal context). The annotation is deposited back into the scene by quickly swiping the hand out of the camera's view, which forces the annotation to rely on AugR's registration to stay fixed in the scene.

The earlier discussions recommended AugR be improved with a newer platform and faster methods and a systematic usability evaluation. The suite of tools for a remote expert and field worker could be expanded, such as enhanced and customizable telepointers, 3-D object, speech, etc. Some of the concerns about accuracy and reliability highlighted by the mosaicing algorithm evaluation can be addressed by a user interface confidence indicator and tools to introduce manual correction (like the manual scheme mentioned in section 5.4). The mosaicing algorithms could be modified to incorporate and learn from user correction. Better blending techniques and frame-by-frame control could allow remote experts to see a more seamless mosaic than in the current AugR system, and allow navigation to specific past frames (which are normally covered up in a live, updating mosaic). In addition, AugR lacks any tools for the field worker to interact back with the remote expert. Besides a live video and audio link, gestural inputs and personal context effects (e.g. glancing at body parts) could form the basis of such tools. Also, AugR was tested for one expert

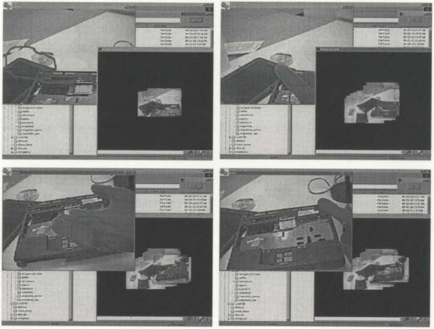


Figure 7.1: An example of a hand-based personal context with a mosaicing world model

Each screenshot shows the field worker's view on the upper left and the remote expert's composite mosaic on the lower right. From left to right, top to down, the screens show a virtual annotation created by the remote expert being grabbed by the field worker's hand, and placed a different location and one worker. A study on the behaviour and development of toolsets supporting a small team of several co-located and separated experts and workers, as well as a larger community of such individuals would be interesting future work. Similarly, an examination of how multiple personal contexts can interact with each other, or with general purpose context awareness is an open area for research. A simple case is how a system like Footprint could support a pair of dancers, or a ballroom full of dancers with and without personal context assistance.

With regards to personal context itself, there are potential explorations into the use XML as the language to model for human-wearable interaction. Spohrer [224] proposes using an XML-based model of annotating all objects in the world. The A-TON system [101] uses XML to model spatial position behind a navigation solution using handhelds and augmented reality for the U.S Coast Guard. The G-XML project is developing an XML specification for applications integrating Geographic Information System (GIS) spatial data [58]. Dey et al [65] use XML to represent a user's notes taken within a note-taking context awareness application. Ryan [208] and Ryan et al [209] detail an XML representation of context data between a field reporting system and server, but is biased for location and time context types. Kortuem et al [123] have an XML format for user profiles which can be exchanged between wearable computer users to facilitate mobile collaboration and discovery of colleagues with similar interests. Whereas these other XML representations cover identity, time, and location, Footprint's XML notation represents activity. QuickSet in [178] is similar, but more higher level, being a logical framework for representing multimodal user interface interaction.

Also, at the time of the writing of this thesis, OASIS, an international consortium promoting and developing XML standards, has announced a technical committee to develop the Human Markup Language, HumanML [170]. Taking a very general approach to human contextual awareness, HumanML hopes to represent cultural, social, kinesic (body language) psychological, and intentional features within XML information. OASIS foresees HumanML applications in a wide variety of domains, including artificial intelligence, virtual reality, conflict resolution, psychotherapy, art, workflow, advertising, cultural dialogue, agent systems, diplomacy and business negotiation. Future research into personal context could leverage HumanML and similar work to create notations and annotated datasets for a wearable computer user's interactions with other people and the world. Given a standard notation for human activity outside the desktop, future work can examine how to tie such data with XML data stored on traditional desktops and servers (e.g. user profiles, shopping preferences, online transaction histories) to create powerful human-computer interfaces.

7.5 Final Remarks

In conclusion, this research achieved its goals, in presenting a number of novel augmented reality prototypes using image techniques for registration and user activity detection. While not operating under real-world conditions with optimum accuracy, reliability, and speeds, the prototypes showed real uses for the proposed concepts of "Personal Context" and "Mosaic as the Interface". All the prototypes operated entirely on a self-contained wearable computer platform, that while not as power efficient or outdoors-hardened as other systems, was portable, networked, video-enabled, and capable of limited registration and graphical overlays. A systematic evaluation of different mosaicing registration methods was also conducted, revealing potential successors to the phase correlation method used for AugR's mosaicing system. Finally, a number of improvements and directions for future research are presented, delving into further extensions of personal context, mosaicing, and how both can be combined for future wearable augmented reality systems.

Bibliography

[1] G.D. Abowd, A.K. Dey, R. Orr, and J. Brotherton. Context-awareness in wearable and ubiquitous computing. Technical report, GVU Technical Report, GIT-GVU-97-11, May 1997. http://www.cc.gatech.edu/fce/pubs/iswc97/wear.html.

[2] M. Abrash. *The Zen of Graphics Programming, 2nd edition.* Coriolis Group Books, Scottsdale, 1996.

[3] S. Alliney and C. Morandi. Digital image registration using projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):222–233, March 1986.

[4] Xybernaut North America. Xybernaut at work: BOC gases case study. Technical Report CS-BOC-01, Xybernaut Corporation, 1999. http://www.xybernaut.com/downloadables/xybcsboc.pdf.

[5] Xybernaut North America. Xybernaut at work: Maintenance application brief. Technical Report AB-MAIN-01, Xybernaut Corporation, 1999. http://www.xybernaut.com/downloadables/manufacturingbrief.pdf.

[6] H. Aoki, B. Schiele, and A. Pentland. Realtime personal positioning system for a wearable computers. In *Proceedings of the Third International Symposium on Wearable Computers*, pages 37–43, San Francisco, USA, October 18-19 1999.

[7] Mobile & Context-Aware Computing Research Group at University of Kent at Canterbury. Fieldnote fieldwork software. http://www.cs.ukc.ac.uk/projects/mobicomp/Fieldwork/Software/index.html.

[8] N. Ayache, editor. *Computer Vision, Virtual Reality, and Robotics in Medicine (CVRMed '95)*, number 905 in Lecture Notes in Computer Science, Nice, France, April 1995. Springer-Verlag.

[9] R. Azuma and G. Bishop. Improving static and dynamic registration in an optical see-through HMD. In *Proceedings of SIGGRAPH '94*, pages 197–204, Orlando, July 1994.

[10] R. Azuma, B. Hoff, H. Neely, and R. Sarfaty. A motion-stabilized outdoor augmented reality system. In *Proceedings of IEEE Virtual Reality*, pages 252–259, Houston, USA, March 13-17 1999.

[11] R.T. Azuma. A survey of augmented reality. *Presence*, 6(4):355–385, August 1997.

[12] F. Badra, A. Qumsieh, and G. Dudek. Robust mosaicing using zernike moments. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(5):685–704, May 1999.

[13] P. Bahl and V. Padmanabhan. Radar: An in-building RF-based user location and tracking system. In *Proceedings of IEEE INFOCOM 2000 Conference on Computer Communications, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 775–784, Los Alamitos, CA, 2000. IEEE Computer Society.

[14] M. Bajura and U. Neumann. Dynamic registration correction in video-based augmented reality systems. *IEEE Computer Graphics and Applications*, 15(5):52–60, September 1995.

[15] L. Bass, C. Kasabach, R. Martin, D. Siewiorek, A. Smailagic, and J. Stivoric. Design of a wearable computer. In *Proceedings of Conference on Human Factors in Computing Systems (CHI 97)*, Atlanta, Georgia, March 22-27 1997. http://www.acm.org/sigchi/chi97/proceedings/paper/ljb1.htm.

[16] M. Bauer, G. Kortuem, and Z. Segall. "Where Are You Pointing At" A study of remote collaboration in a wearable videoconference system. In *Proceedings of the Third International Symposium on Wearable Computers*, San Francisco, USA, October 18-19 1999.

[17] R. Behringer. Registration for outdoor augmented reality applications using computer vision techniques and hybrid sensors. In *Proceedings of IEEE Virtual Reality*, pages 244–251, Houston, USA, March 13-17 1999.

[18] T. Bentley. Body wearable computer applications, 2000. http://science.ksc.nasa.gov/payload/projects/borg/.

[19] J.W. Berger, M.E. Leventon, N. Hata, W.M. Wells III, and R. Kikinis. Design considerations for a computer-vision-enabled ophthalmic augmented reality environment. In *CVRMED/MRCAS*, Grenoble, France, 1997.

[20] S. Bilcar, M. Curry, G. Kortuem, R. Toub, and P. Walser. Mediwear : CIS 650 - software engineering winter '95 final project, 1995. http://www.cs.uoregon.edu/research/wearables/MediWear/.

[21] M. Billinghurst, J. Bowskill, N. Dyer, and J. Morphett. An evaluation of wearable information spaces. In *Proceedings of IEEE Virtual Reality Annual International Symposium (VRAIS 98)*, pages 20–27, Atlanta, March 14-18 1998.

[22] M. Billinghurst, J. Bowskill, M. Jessop, and J. Morphen. A wearable spatial conferencing space. In *Proceedings of the Second International Symposium on Wearable Computers*, pages 72–83, Pittsburgh, USA, October 19-20 1998.

[23] M. Billinghurst and H. Kato. Collaborative mixed reality. In *Proceedings of the First International Symposium on Mixed Reality*, Yokohama, Japan, March 9-11 1999. Ohmsha-Springer Verlag.

[24] M. Billinghurst, H. Kato, and I. Poupyrev. Magicbook SIGGRAPH 2000 demo media. http://www.hitl.washington.edu/magicbook/media.html.

[25] M. Billinghurst, H. Kato, and I. Poupyrev. The Magicbook-moving seamlessly between reality and virtuality. *IEEE Computer Graphics and Applications*, 21(3):6–8, May/June 2001.

[26] Mark Billinghurst. Human interface technology lab - ARToolkit/shared space download page. http://www.hitl.washington.edu/research/shared_space/download/.

[27] T. Blaszka and R. Deriche. Recovering and characterizing image features using an efficient model based approach. Technical report, INRIA Technical Report 2422, November 1994. ftp://ftp.inria.fr/INRIA/tech-reports/RR/RR-2422.ps.gz.

[28] Boeing. Boeing technology focus. http://www.boeing.com/assocproducts/art/tech_focus.html.

[29] Boeing. Boeing wearable computer workshop breakout session summary, August 19-21 1996. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/vuman/www/boeing/index.html.

[30] L.G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, December 1992.

[31] J. Burrell and G. Gay. Collectively defining context in a mobile, networked computing environment. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI 2001)*, pages 231–232, Seattle, WA, March 31-April 4 2001.

[32] P. Burt and E. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[33] P.J. Burt. The pyramid as a structure for efficient computation. In A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, pages 6–35. Springer-Verlag, Berlin, 1984.

[34] A. Butz, J. Baus, and A. Kruger. Augmenting buildings with infrared information. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality 2000*, pages 93–96, Munich, Germany, October 5-6 2000.

[35] N. Campbell, H. Muller, and C. Randel. Combining position information with visual media. In *Proceedings of the Third International Symposium on Wearable Computers*, pages 203–205, San Francisco, USA, October 18-19 1999.

[36] L. Cheng and J. Robinson. Dealing with speed and robustness issues for video-based registration on a wearable computing platform. In *Proceedings of the Second International Symposium on Wearable Computers*, pages 84–91, Pittsburgh, USA, October 19-20 1998.

[37] S. Cheng. Quicktime VR : An image-based approach to virtual environment navigation. In *Proceedings of the 22nd Annual ACM Conference on Computer Graphics (SIGGRAPH 95)*, Los Angeles, CA, USA, August 6-11 1995.

[38] K. Cheverst, N. Davies, K. Mitchell, and A. Friday. Experiences of developing and deploying a context-aware tourist guide: the guide project. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MOBICOM 2000)*, pages 20–31, Boston, MA, USA, August 6-11 2000.

[39] C. Chiang, A. Huang, T. Wang, M. Huang, Chen Y., J. Hsieh, J. Chen, and T. Cheng. PanoVR SDK-A software development kit for integrating photo-realistic panoramic images and 3-d graphical objects into virtual worlds. In

Proceedings of the ACM Symposium on Virtual Reality Software and Technology '97, pages 147–154, Lausanne, Swizterland, September 15-17 1997.

[40] Y. Cho, J. Park, and U. Neumann. Fast color fiducial detection and dynamic workspace extension in video see-through self-tracking augmented reality. In *Proceedings of the Fifth Pacific Conference on Computer Graphics and Applications*, pages 168–177, Seoul, Korea, October 13-16 1997.

[41] I. Choi and C. Ricci. Foot-mounted gesture detection and its application in a virtual reality environment. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 5, pages 4248–4253, Orlando, USA, October 12–15 1997.

[42] R. Clarke. *Digital Compression of Still Images and Video.* Academic Press, London, 1996.

[43] B. Clarkson, K. Mase, and A. Pentland. Recognizing user context via wearable sensors. In *Proceedings of the Fourth International Symposium on Wearable Computers*, pages 69–76, Atlanta, GA, October 16-17 2000.

[44] P. Colet. Beyond the "shrink-wrapped library"versus "ground up" development choice. *Advanced Imaging*, 13(2):68–70, February 1998.

[45] Apple Computer. Quicktime VR Authoring Studio. http://www.apple.com/quicktime/qtvr/authoringstudio/stitcher.html.

[46] ARVIKA Consortium. ARVIKA home page. http://www.arvika.de.

[47] W3C World Wide Web Consortium. Synchronized multimedia. http://www.w3.org/AudioVideo/.

[48] Handykey Corporation. Handykey Corporation Home Page. http://www.handykey.com.

[49] MicroOptical Corporation. Eyeglass display FAQ. http://www.microopticalcorp.com/eyefaq.htm.

[50] Microsoft Corporation. Windows CE home page. http://www.microsoft.com/windowsce.

[51] Microvision Corporation. Microvision home page. http://www.mvis.com.

[52] Sarnoff Corporation. Videobrush PC stitching applications. http://www.sarnoff.com/government_professional/vision_technology /products/videobrush_pc_stitching.asp.

[53] Xybernaut Corporation. Germany's first web-reporter describes how the wearable pc from xybernaut improved her journalistic work. http://www.xybernaut.com/wear/case_sub10.htm.

[54] Xybernaut Corporation. Welcome to xybernaut corporation. http://www.xybernaut.com.

[55] Xybernaut Corporation. Xybernaut public relations photos. http://www.xybernaut.com/public/pub_grph.htm.

[56] C. Crowley. A finite state machine for western swing. *SIGPLAN Notices*, 16(4):33–35, April 1981.

[57] M. Darty. Maintenance and repair support system (MARSS), June 1995. http://web-ext2.darpa.mil/ETO/SmartMod/Factsheets/MARSS.html.

[58] Japan Database Promotion Center. G-XML project home page, May 31 2001. http://gisclh.dpc.or.jp/gxml/.

[59] G. Davenport. Visions and views: Curious learning, cultural bias, and the learning curve. *IEEE Multimedia*, 5(2):14–19, April/June 1998.

[60] J. Davis. Mosaics of scenes with moving objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 98)*, pages 354–360, Santa Barbara, USA, June 23-25 1998.

[61] A. De Angeli, L. Romary, and F. Wolff. Ecological interfaces: Extending the pointing paradigm by visual context. In *Modeling and Using Context, Second International and Interdisciplinary Conference, CONTEXT 99*, pages 91–104, Trento, Italy, September 1999.

[62] E. DeCastro and C. Morandi. Registration of translated and rotated images under finite Fourier transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):700–703, September 1987.

[63] B. Delaney. On the trail of the shadow woman: The mystery of motion capture. *IEEE Computer Graphics and Applications*, 18(5):14–19, September/October 1998.

[64] A. Dey and G. Abowd. Cybreminder: A context-aware system for supporting reminders. In *Handheld and Ubiquitous Computing, Second International Symposium (HUC '00)*, Bristol, England, September 25-27 2000.

[65] A. Dey, D. Salber, G. Abowd, and M. Futakawa. The conference assistant: Combining context-awareness with wearable computing. In *Proceedings of the Third International Symposium on Wearable Computers*, pages 21–28, San Francisco, USA, October 18-19 1999.

[66] A.K. Dey and G.D. Abowd. Towards a better understanding of context and context-awareness. Technical report, GVU Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology, USA, 1999. ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf.

[67] D. Drascic. Stereoscopic vision and augmented reality. *Scientific Computing and Automation*, 9(7):31–34, June 1993.

[68] Ericsson. Ericsson: Mobile internet - offerings - location-based services. http://www.ericsson.com/mobileinternet/offerings/loc_services.shtml.

[69] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the world : 2D windows for 3D augmented reality. In *ACM Symposium on User Interface Software and Technology*, pages 145–155, Atlanta, Georgia, November 3-5 1993.

[70] S. Feiner, B. MacIntyre, and T. Höllerer. Wearing it out: First steps toward mobile augmented reality systems. In *Proceedings of the First International Symposium on Mixed Reality*, Yokohama, Japan, March 9-11 1999. Ohmsha-Springer Verlag.

[71] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster. A Touring Machine: Prototyping 3D mobile augmented reality systems for exploring the urban evironment. In *Proceedings of the First International Symposium on Wearable Computers*, pages 74–81, Cambridge, MA, October 13-14 1997.

[72] S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):53–62, July 1993.

[73] S. Feiner, A. Webster, and B. MacIntyre. Augmented reality for construction, 1996. http://www.cs.columbia.edu/graphics/projects/arc/arc.html.

[74] S. Feiner, T. Webster, T. Krueger, B. MacIntyre, and E. Keller. Architectural anatomy, 1994. http://www.cs.columbia.edu/graphics/projects/archAnatomy/architecturalAnatomy.html.

[75] S. Fickas, G. Kortuem, and Z. Segall. Software organization for dynamic and adaptable wearable systems. In *Proceedings of the First International Symposium on Wearable computers*, Cambridge, Massachusetts, October 13-14 1997.

[76] G.W. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, 36(7):39–49, July 1993.

[77] A. Fuhrmann, H. Löffelmann, D. Schmalstieg, and M. Gervautz. Collaborative visualization in augmented reality. *IEEE Computer Graphics and Applications*, 18(4):54–59, July/August 1998.

[78] M. Fukumoto and Y. Tonomura. "Body Coupled FingeRing": Wireless wearable keyboard. In *Proceedings of Conference on Human Factors in Computing Systems (CHI 97)*, Atlanta, Georgia, March 22-27 1997.

[79] S. Fussell, R. Kraut, and J. Siegel. Coordination of communication: Effects of shared visual context on collaborative work. In *Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work (CSCW 2000)*, pages 21–30, Philadelphia, PA, December 2-6 2000.

[80] D. Gibson and M. Spann. Robust motion trajectory estimation for long image sequences with applications to motion compensated prediction. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(5):781–802, May 1999.

[81] C. Gorman. Pocket-size medicine. *Time*, 148(14):46, September 23 1996.

[82] W.E.L. Grimson, T. Lozano-Perez, W.M. Wells III, J.G. Ettinger, S.J. White, and R. Kikinis. An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 94)*, pages 430–436, Seattle, Washington, June 21-23 1994.

[83] Computer Vision Group. Project on image guided surgery: A collaboration between the MIT AI lab and brighamand women's surgical planning laboratory. http://www.ai.mit.edu/projects/vision-surgery/surgery_home_page.html.

[84] Future Computing Environments Group. Cyberguide project page, 1997. http://www.cc.gatech.edu/fce/cyberguide/index.html.

[85] GeorgiaTech Wearables Group. Gatech wearables page. http://wearables.gatech.edu/EPSS.asp.

[86] MIT Wearable Computing Group. The MIT wearable computing web page. http://wearables.www.media.mit.edu/projects/wearables/.

[87] MIT Wearable Computing Group. Wearable computing FAQ version 1.0. http://wearables.www.media.mit.edu/projects/wearables/FAQ/FAQ.txt.

[88] S. Gümüstekin. An introduction to image mosaicing, July 1999. http://www.pitt.edu/~sevgum/research/mosaicing/.

[89] S. Gümüstekin and R. Hall. Mosaic image generation on a flattened gaussian sphere. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, pages 50–55, 1996.

[90] R. Handley. Neuro-navigation: Bringing new techniques to the (operating) table. *Advanced Imaging*, 16(7):52–57, July 2001.

[91] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MOBICOM'99*, pages 59–68, Seattle, Washington, USA, August 1999.

[92] B. Hoff and R. Azuma. Autocalibration of an electronic compass in an outdoor augmented reality system. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality 2000*, pages 159–164, Munich, Germany, October 5-6 2000.

[93] W. Hoff and T. Vincent. Analysis of head pose accuracy in augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 6(4):319–334, 2000.

[94] R. Hooke. *Micrographia; or Some physiological descriptions of minute bodies made by magnifying glasses, with observations and inquiries thereupon.* Dover Publications, New York, 1961. unabridged, fascimile reproduction of University of Pennsylvania Library copy of the first edition, published in 1665.

[95] B. Horn and B. Schunck. Determing optical flow. *Artificial Intelligence*, (17):185–203, 1981.

[96] J. Hsieh, H. Liao, K. Fan, and M. Ko. A fast algorithm for image registration without predetermining correspondences. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume I, Track A, pages 765–769, Vienna, Austria, August 25-29 1996.

[97] i O Display Systems. i-O display systems home page. http://www.i-glasses.com.

[98] DBS Imaging. The imaging souce: AdOculos. http://www.dbs-imaging.com/prod/soft/adoculos/adoculos.htm.

[99] iMove. iMove Home Page. http://www.imoveinc.com/03products/.

[100] Information in Place. IIPI products. http://www.informationinplace.com/jnav_products/current_products.html.

[101] Information in Place. Visual aids to navigation (V-ATON). http://www.informationinplace.com/current_products_vaton.html.

[102] Mathworks Inc. The mathworks : Developers of MATLAB and Simulink for technical computing. http://www.mathworks.com.

[103] Mixed Reality Systems Laboratory Inc. Aquagauntlet description page. http://www.mr-system.co.jp/project/aquagauntlet/index.html.

[104] ViA Inc. Via Inc. the flexible PC company. http://www.via-pc.com.

[105] Intel. Intel image processing library. http://developer.intel.com/ software/products/perflib/ipl/.

[106] Intel. Open source computer vision library. http://intel.com/research/mrl/research/opencv/.

[107] S. Intille, J. Davis, and A. Bobick. Real-time closed-world tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 97)*, pages 697–703, San Juan, USA, June 17-19 1997.

[108] M. Irani and P. Anandan. Video indexing based on mosaic representations. ftp://ftp.wisdom.weizmann.ac.il/pub/irani/PAPERS/VideoIndexing/TextPages/, 1997.

[109] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using region alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):268–272, March 1997.

[110] H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits, and atoms. In *Proceedings of Conference on Human Factors in Computing Systems (CHI 97)*, pages 234–241, Atlanta, Georgia, March 22-27 1997.

[111] T. Jebara, C. Eyster, J. Weaver, T. Starner, and A. Pentland. Stochasticks: Augmenting the billiards experience with probabilistic vision and wearable computers. In *Proceedings of the First International Symposium on Wearable Computers*, pages 138–145, Cambridge, MA, October 13-14 1997.

[112] S. Julier, M. Lanzagorta, Y. Baillot, L. Rosenblum, S. Feiner, T. Höllerer, and S. Sestito. Information filtering for mobile augmented reality. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality 2000*, pages 3–11, Munich, Germany, October 5-6 2000.

[113] S. Kang and R. Weiss. Characterization of errors in compositing panoramic images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 97)*, pages 103–109, San Juan, USA, June 17-19 1997.

[114] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top AR environment. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality 2000*, pages 111–119, Munich, Germany, October 5-6 2000.

[115] S. Kaufman, I. Poupyrev, E. Miller, M. Billinghurst, P. Oppenheimer, and S. Weghorst. New interface metaphors for complex information space visualization: a ECG monitor object prototype. In *Proceedings of Medicine Meets Virtual Reality V*, 1997.

[116] D. Kiely. Are components the future of software? *IEEE Computer*, 31(2):10–11, February 1998.

[117] H.S. Kim, H.C. Kim, W.K. Lee, and C.H. Kim. Stiching reliability for estimating camera focal length in panoramic image mosaicing. In *Proceedings of the International Conference on Pattern Recognition*, Barcelona, Spain, September 3-8 2000.

[118] H.W. Kim and K.S. Hong. Robust image mosaicing of soccer videos using self-calibration and line tracking. *Pattern Analysis and Application*, 1(1):9–19, 2001.

[119] S. Kirsner. Wired news: Booting up something more comfortable. http://www.wired.com/news/news/culture/story/7660.html.

[120] Y. Kita, D. Wilson, J. Noble, and N. Kita. A quick 3D-2D registration method for a wide-range of applications. In *Proceedings of the International Conference on Pattern Recognition*, Barcelona, Spain, September 3-8 2000.

[121] J. Kollin. A retinal display for virtual-environment applications. In *Proceedings of the 1993 International Symposium of the Society for Information Display*, volume 24, page 827, Playa del Rey, California, USA, 1993.

[122] Konami. Dance Dance Revolution. http://www.konami-arcade.com/Music/Ddr.

[123] G. Kortuem, Z. Segall, and T. Thompson. Close encounters: Supporting mobile collaboration through interchange of user profiles. In *Handheld and Ubiquitous Computing, First International Symposium (HUC '99)*, pages 171–185, Karlsruhe, Germany, September 27-29 1999.

[124] M. Kourogi, T. Kurata, J. Hoshino, and Y. Muraoka. Real-time image mosaicing from a video sequence. In *Proceedings of the 1999 International Conference on Image Processing (ICIP 99)*, volume 4, pages 133–137, Kobe, Japan, October 24-28 1999.

[125] M. Kourogi, T. Kurata, K. Sakaue, and Y. Muraoka. Improvement of panorama-based annotation overlay using omnidirectional vision and inertial sensors. In *Proceedings of the Fourth International Symposium on Wearable Computers*, pages 183–184, Atlanta, GA, October 16-17 2000.

[126] M. Kourogi, T. Kurata, K. Sakaue, and Y. Muraoka. A panorama-based technique for annotation overlay and its real-time implementation. In *Proceedings of the IEEE Conference on Multimedia and Expo 2000*, New York City, NY, USA, July 30 - August 2 2000.

[127] M. Kozaburo and Y. Ohno. A system for the representation of human body movement from dance scores. *Pattern Recognition Letters*, 5:1–9, January 1987.

[128] R.E. Kraut, M.D. Miller, and J. Siegel. Collaboration in performance of physical tasks: Effects on outcomes and communication. In *Proceedings of CSCW 96*, pages 57–66, Boston, Massachusetts, November 16-20 1996.

[129] M.M. Krueger. Environmental technology: Making the real world virtual. *Communications of the ACM*, 36(7):36–37, July 1993.

[130] C.D. Kuglin and D.C. Hines. The phase correlation image alignment method. In *Proceedings of the 1975 International Conference of the Cybernetics Society*, pages 163–165, New York, USA, 1975.

[131] K. Kutulakos and J. Vallino. Affine object representations for calibration-free augmented reality. In *Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS 96)*, pages 25–35, Santa Clara, USA, March 30 - April 3 1996.

[132] AT&T Cambridge Research Lab. Sentient computing research. http://www.uk.research.att.com/spirit/.

[133] MIT Media Lab. MIT MITHRIL project. http://www.media.mit.edu/ wearables/ mithril/photos.html.

[134] R. Laban. *Laban's Principles of Dance and Movement Notation.* MacDonald & Evans Ltd., London, UK, 1975.

[135] T. Lao, K. Wong, K. Lee, and S. Or. Creating virtual walkthrough environment from vertical panoramic mosaic. In *Proceedings of the International Conference on Pattern Recognition*, Barcelona, Spain, September 3-8 2000.

[136] N. Lecky. Board level industrial imaging/machine vision markets now: The impact of getting easier. *Advanced Imaging*, 13(2):10–14, February 1998.

[137] F. Lerasle, G. Rives, and M. Dhome. Tracking of human limbs by multiocular vision. *Computer Vision and Image Understanding*, 75(3):229–246, September 1999.

[138] M. Leung and Y. Yang. First sight: A human body outline labeling system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):359–377, April 1995.

[139] M.A. Livingston and A. State. Magnetic tracker calibration for improved augmented reality registration. *Presence*, 6(5):532–546, October 1997.

[140] Logitech. Logitech home page. http://www.logitech.com.

[141] QNX Software Systems Ltd. QNX software systems ltd. home page. http://www.qnx.com.

[142] L. Lucchese, G.M. Cortelazzo, and C. Monti. Estimation of affine transformations between image pairs via fourier transform. In *Proceedings of the 1996 International Conference on Image Processing (ICIP 96)*, pages 715–718, Lausanne, Switzerland, September 1996.

[143] B. MacIntyre and E. Coelho. Adapting to dynamic registration errors using level of error (loe) filtering. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality 2000*, pages 85–88, Munich, Germany, October 5-6 2000.

[144] B. MacIntyre and R. Kooper. The real-world wide web browser: An interface for a continuously available, general purpose, spatialized information space. In *Proceedings of the Second International Symposium on Mixed Reality*, Yokohama, Japan, March 14-15 2001. Ohmsha-Springer Verlag.

[145] P. Maes, T. Darrell, B. Blumberg, and A. Pentland. The ALIVE System: Wireless, full-body interaction with autonomous agents. Technical report, M.I.T. Media Laboratory Perceptual Computing Technical Report No. 257, Cambridge, Massachusetts, 1995. ftp://whitechapel.media.mit.edu/pub/tech-reports/TR-257.ps.Z.

[146] S. Mann. Video orbits of the projective group: A new perspective on image mosaicing. http://www.eyetap.org.

[147] S. Mann. Wearable, tetherless computer mediated reality: Wearcam as a wearable face recognizer, and other applications for the disabled. In *Proceedings of the AAAI Fall Symposium on Developing Assistive Technology*, November 9-11 1996.

[148] S. Mann. Wearable computing: A first step toward personal imaging. *Computer*, 30(2):25–32, February 1997.

[149] S. Mann. Wearable intelligent signal processing. *Proceedings of the IEEE*, 86(11):2123–2151, November 1998.

[150] S. Mann. Mediated Reality: University of Toronto RWM project. *Linux Journal*, (59):50–57, March 1999.

[151] S. Mann. *Intelligent Image Processing.* John Wiley and Sons, Inc., New York, USA, 2002.

[152] S. Mann and J. Fung. Videoorbits on eye tap devices for deliberately diminished reality or altering the visual perception of rigid planar patches of a real world scene. In *Proceedings of the Second International Symposium on Mixed Reality*, Yokohama, Japan, March 14-15 2001. Ohmsha-Springer Verlag.

[153] S. Mann and R.W. Picard. Video orbits of the projective group: A simple approach to featureless estimation of parameters. *IEEE Transactions on Image Processing*, 6(9):1281–1295, September 1997.

[154] T. Martin, E. Jovanov, and D. Raskovic. Issues in wearable computing for medical monitoring applications: A case study of a wearable ECG monitoring device. In *Proceedings of the Fourth International Symposium on Wearable Computers*, pages 43–50, Atlanta, GA, October 16-17 2000.

[155] J.P. Mellor. Realtime camera calibration for enhanced reality visualization. In N. Ayache, editor, *Computer Vision, Virtual Reality, and Robotics in Medicine (CVRMed '95)*, number 905 in Lecture Notes in Computer Science, pages 471–475, Nice, France, April 1995. Springer-Verlag.

[156] Microsoft. Microsoft developer network : DirectX. http://msdn.microsoft.com/ directx.

[157] Sun Microsystems. Java home page. http://www.javasoft.com.

[158] J. Miyasato. See-through hand. In *Proceedings of the Eigth Australian Conference on Computer Human Interaction (OzCHI 98)*, Adelaide, Australia, November 29-December 4 1998.

[159] D. Mizell. Wearable computer systems with head-mounted displays for manufacturing, maintenance, and training applications, August 1995. http://web-ext2.darpa.mil/ETO/SmartMod/Factsheets/BoeingTRP.html.

[160] mobilePosition. mobilePosition home page. http://www.mobileposition.com.

[161] C. Morimoto and R. Chellappa. Fast 3D stabilization and mosaic construction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 97)*, pages 660–665, San Juan, USA, June 17-19 1997.

[162] L. Najjar, J.C. Thompson, and J.J. Ockerman. Using a wearable computer to improve the performance of quality assurance inspectors in a food processing plant. In *Workshop on Wearable Computing, Proceedings of Conference on Human Factors in Computing Systems (CHI 97)*, Atlanta, Georgia, March 22-27 1997.

[163] L. Najjar, J.C. Thompson, and J.J. Ockerman. A wearable computer for quality assurance inspectors in a food processing plant. In *Proceedings of the First International Symposium on Wearable Computers*, pages 163–164, October 13-14 1997. http://mime1.marc.gatech.edu/MiME/papers/ISWC_Larry.html.

[164] L. Naugle. Visions and views: Digital dancing. *IEEE Multimedia*, 5(4):8–12, October/December 1998.

[165] United States Navy. Battlefield AR system. http://www.ait.nrl.navy.mil/vrlab/projects/BARS/BARS.html.

[166] U. Neumann, S. You, Y. Cho, J. Lee, and J. Park. Augmented reality tracking in natural environments. In *Proceedings of the First International Symposium on Mixed Reality*, Yokohama, Japan, March 9-11 1999. Ohmsha-Springer Verlag.

[167] U. Neumann and Suya You. Integration of region tracking and optical flow for image motion estimation. In *Proceedings of the 1998 International Conference on Image Processing (ICIP 98)*, volume 3, pages 658–662, October.

[168] N. Newmann and A. Clark. Sulawesi: A wearable application integration framework. In *Proceedings of the Third International Symposium on Wearable Computers*, pages 170–171, San Francisco, USA, October 18-19 1999.

[169] Nonin Medical Inc. Nonin Medical home page. http://www.nonin.com.

[170] OASIS. Oasis members form technical committee to develop human markup language. http://www.oasis-open.org/news/oasis_news_08_21_01.shtml.

[171] J.J. Ockerman, L. Najjar, and J.C. Thompson. Wearable computers for performance support: Initial feasibility study. In *Proceedings of the First International Symposium on Wearable Computers*, pages 10–17, October 13-14 1997. http://mime1.marc.gatech.edu/MiME/papers/ISWC_Jenn.html.

[172] K. Oeler. Now, computers are wearable, October 2 1997. http://www.news.com/News/Item/0,4,14850,00.html.

[173] Australian Institute of Marine Science. WetPC photostory. http://www.aims.gov.au/pages/wetpc/wpcphotostory.html.

[174] Australian Institute of Marine Science. WetPC, 1997. http://www.aims.gov.au/pages/wetpc/wetpc.html.

[175] J. Ohya, J. Kurumisawa, and R. Nakatsu. Virtual metamorphosis. *IEEE Multimedia*, 6(2):29–39, April/June 1999.

[176] Linux Online. The Linux home page. http://www.linux.org.

[177] R. O'Rafferty, M. O'Grady, and G. O'Hare. A rapidly configurable location-aware information system for an exterior environment. In *Handheld and Ubiquitous Computing, First International Symposium (HUC '99)*, pages 334–336, Karlsruhe, Germany, September 27-29 1999.

[178] S. Oviatt and P. Cohen. Multimodal interfaces that process what comes naturally. *Communications of the ACM*, 43(3):45–53, March 2000.

[179] J. Paradiso, K. Hsiao, A. Benbasat, and Z. Teegarden. Design and implementation of expressive footwear. *IBM Systems Journal*, 39(3 and 4):511–529, 2000.

[180] J. Pascoe. Adding generic contextual capabilities to wearable computers. In *Proceedings of the Second International Symposium on Wearable Computers*, pages 92–99, Pittsburgh, USA, October 19-20 1998.

[181] J. Pascoe, N. Ryan, and D. Morse. Issues in developing context-aware computing. In *Handheld and Ubiquitous Computing, First International Symposium (HUC '99)*, pages 208–221, Karlsruhe, Germany, September 27-29 1999.

[182] J.J. Pearson, D.C. Hines, S. Golosman, and C.D. Kuglin. Video-rate image correlation processor. In *SPIE Proceedings: Applications of Digital Image Processing*, volume 119, pages 197–205, San Diego, USA, 1977.

[183] M. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 97)*, pages 338–343, San Juan, USA, June 17-19 1997.

[184] S. Peleg. Elimination of seams from photomosaics. *Computer Graphics and Image Processing*, 16:90–94, 1981.

[185] A. Pentland. Perceptual intelligence. In *Handheld and Ubiquitous Computing, First International Symposium (HUC '99)*, pages 74–88, Karlsruhe, Germany, September 27-29 1999.

[186] A. Pentland. Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):107–118, January 2000.

[187] A. Pentland. Perceptual intelligence. *Communications of the ACM*, 43(3):35–44, March 2000.

[188] K. Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):5–15, March 1995.

[189] P. Persson, F. Espinoza, and E. Cacciatore. Geonotes: Social enhancement of physial space. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI 2001)*, pages 43–44, Seattle, WA, March 31-April 4 2001.

[190] R.W. Picard and J. Healey. Affective wearables. In *Proceedings of the First International Symposium on Wearable Computers*, Cambridge, USA, October 13-14 1997.

[191] R.W. Picard and J. Healey. Startlecam: A cybernetic wearable camera. In *Proceedings of the Second International Symposium on Wearable Computers*, pages 42–49, Pittsburgh, USA, October 19-20 1998.

[192] A. Pope and D. Lowe. Vista: A software environment for computer vision research. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 94)*, pages 768–772, Seattle, Washington, June 21–23 1994.

[193] G. Priest-Dorman. Wearable computing, 1997. http://www.cs.vassar.edu/~priestdo/wearable.html.

[194] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MOBICOM 2000)*, pages 32–43, Boston, MA, USA, August 6-11 2000.

[195] Sony Computing Products. Sony computing products: PLM-S700 PC Glasstron. http://www.ita.sel.sony.com/products/av/glasstron/.

[196] J. Rantanen, N. Alfthan, J. Impiö, T. Karinsalo, M. Malmivaara, and R. Matala. Smart clothing for the arctic environment. In *Proceedings of the Fourth International Symposium on Wearable Computers*, pages 15–24, Atlanta, GA, October 16-17 2000.

[197] A. Rastogi, P. Milgram, and J.J. Grodski. Augmented telerobotic control: A visual display for unstructured environments. In *Proceedings of the 1995 KBS/Robotics Conference*, October 16-18 1995.

[198] S. Ravela, B. Draper, J. Lim, and R. Weiss. Adaptive tracking and model registration across distinct aspects. In *Proceedings of IEEE International Conference On Intelligent Robots and Systems (IROS)*, pages 174–180, Pittsburgh, USA, August 1995.

[199] S. Reddy and B.N. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271, August 1996.

[200] IBM Research. IBM Watchpad Computer. http://www.research.ibm.com/WearableComputing/photos.html.

[201] Microsoft Research. Vision SDK home page. http://research.microsoft.com/projects/VisSDK/.

[202] B. Rhodes. A brief history of wearable computing. http://wearables.www.media.mit.edu/projects/wearables/timeline.html.

[203] J. Robinson and L. Cheng. Projective transform estimation: Design criteria, performance analysis and a new method, October 2001. Internal technical Report, in submission for journal publication.

[204] J. Robinson and A.C. Robertson. The LivePaper system: Augmenting paper on an enhanced tabletop. *Computers and Graphics*, 25(5):731–743, October 2001.

[205] P. Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291–307, February 1999.

[206] D. Roy, C. Sawhney, C. Schmandt, and A. Pentland. Wearable audio computing: A survey of interaction techniques. Technical report, M.I.T. Media Laboratory Perceptual Computing Technical Report No. 434, Cambridge, Massachusetts, 1997. ftp://whitechapel.media.mit.edu/pub/tech-reports/TR-434.ps.Z.

[207] W. Rungsarityotin and T. Starner. Finding location using omnidirectional video on a wearable computing platform. In *Proceedings of the Fourth International Symposium on Wearable Computers*, pages 61–68, Atlanta, GA, October 16-17 2000.

[208] N. Ryan. ConteXML: Exchanging contextual information between a mobile client and the fieldnote server, August 6 1999. http://www.cs.ukc.ac.uk/research/infosys/mobilecomp/fnc/ConteXtML.html.

[209] N. Ryan, J. Pascoe, and D. Morse. Enhanced reality fieldwork: The context-aware archaeological assistant. In *Computer Applications in Archaeology (CAA 97)*, Oxford, UK, 1998. Archaeopress.

[210] D. Salber, A. Dey, and G. Abowd. The context toolkit: Aiding the development of context-enabled applications. In *Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI 99)*, pages 434–441, Pittsburgh, USA, May 15-20 1999.

[211] D. Schmalstieg, A. Fuhrmann, and G. Hesina. Bridging multiple user interface dimensions with augmented reality. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality 2000*, pages 20–29, Munich, Germany, October 5-6 2000.

[212] A. Schmidt, K. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde. Advanced interaction in context. In *Handheld and Ubiquitous Computing, First International Symposium (HUC '99)*, pages 89–101, Karlsruhe, Germany, September 27-29 1999.

[213] SciTech. Scitech developer products: MGL 4.0. http://www.scitechsoft.com/dp_mgl.html.

[214] T. Selker and W. Burleson. Context-aware design and interaction in computer systems. *IBM Systems Journal*, 39:1–12, 2000.

[215] T. Selker, A. Lockerd, J. Martinez, and W. Burleson. Eye-aRe, a glasses-mounted eye motion deteciton interface. In *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI 2001)*, Seattle, WA, March 31-April 4 2001.

[216] Y. Seo and K. Hong. Calibration-free augmented reality in perspective. *IEEE Transactions on Visualization and Computer Graphics*, 6(4):346–359, 2000.

[217] J. Shi and C. Tomasi. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 94)*, pages 593–600, Seattle, Washington, June 21–23 1994.

[218] A. Shirai, M. Sato, Y. Kume, and M. Kusahara. Foot interface: Fantastic phantom slipper. In *SIGGRAPH 98 Conference Abstracts and Applications*, Orlando, USA, 1998.

[219] H. Shum and R. Szeliski. Panoramic image mosaics. Technical report, Microsoft Research Technical Report MSR-TR-97-23, 1997. ftp://ftp.research.microsoft.com/pub/tr/tr-97-23.ps.

[220] D. Siewiorek. Wearable maintenance assistant, June 1995. http://web-ext2.darpa.mil/ETO/SmartMod/Factsheets/Maintenance.html.

[221] G. Simon, A. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality 2000*, pages 120–128, Munich, Germany, October 5-6 2000.

[222] B. Singletary and T. Starner. Symbiotic interfaces for wearable face recognition. In *Proceedings of HCI International 2001*, New Orleans, USA, August 5-10, 2001 2001.

[223] J. Spohrer. WorldBoard : What comes after the WWW?, 1997. http://trp.atg.apple.com/events/ISITalk062097/parts/WorldBoard/default.html.

[224] J. Spohrer. Information in places. *IBM Systems Journal*, 38(4):551–565, 1999.

[225] J. Spohrer and M. Stein. User experience in the pervasive computing age. *IEEE Multimedia*, 7(1):12–17, January - March 2000.

[226] T. Starner, D. Kirsch, and S. Assefa. The Locust Swarm: An environmentally-powered, networkless location and messaging system. In *International Symposium on Wearable Computers*, Cambridge, Massachusetts, October 13-14 1997.

[227] T. Starner, S Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, and R. Picard. Augmented reality through wearable computing. *Presence*, 6(4):386–398, August 1997.

[228] T. Starner, B. Schiele, and A. Pentland. Visual contextual awareness in wearable computing. In *Proceedings of the Second International Symposium on Wearable Computers*, pages 50–57, Pittsburgh, USA, October 19-20 1998.

[229] L. Stifelman, B. Arons, and C. Schmandt. The Audio Notebook: Paper and pen interaction with structured speech. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI 2001)*, pages 182–189. ACM Press, March 31-April 4 2001.

[230] Supercircuits. Supercircuits home page. http://www.supercircuits.com.

[231] J. Sutherland. A head-mounted three dimensional display. In *Proceedings of the First Joint Computer Conference*, pages 757–764, Washington DC, USA, 1968. Thompson Books.

[232] InterVision Systems. Intervision systems home page. http://www. intervision-systems.com.

[233] InterVision Systems. Typical applications. http://www.intervisionsystems.com/wearable/sys6app.html.

[234] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, pages 22–30, March 1996.

[235] R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 94)*, pages 194–201, Seattle, Washington, June 21–23 1994.

[236] TekGear. Tekgear home page. http://www.tekgear.ca.

[237] B. Thomas, B. Close, J. Donoghue, J. Squires, and P. De bondi. Arquake: An outdoor/indoor augmented reality first person application. In *Proceedings of the Fourth International Symposium on Wearable Computers*, pages 139–146, Atlanta, GA, October 16-17 2000.

[238] G.A. Thomas, J. Jin, T. Niblett, and C. Urquhart. A versatile camera position measurement system for virtual reality tv production. In *Proceedings of International Broadcasting Convention (IBC 97)*, 1997.

[239] M. Tuceryan and N. Navab. Single point active alignment method (SPAAM) for optical see-through hmd calibration for ar. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality 2000*, pages 149–158, Munich, Germany, October 5-6 2000.

[240] H. Ueda, M. Tsukamoto, and S. Nishio. W-mail: an electronic mail system for wearable computing environments. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MOBICOM 2000)*, pages 284–291, Boston, MA, USA, August 6-11 2000.

[241] M. Uenohara and T. Kanade. Vision-based object registration for real-time image overlay. In N. Ayache, editor, *Computer Vision, Virtual Reality, and Robotics in Medicine (CVRMed '95)*, number 905 in Lecture Notes in Computer Science, pages 13–22, Nice, France, April 1995. Springer-Verlag.

[242] Ricoh USA. Ricoh mini notebook PC - Magio. http://www.ricoh-usa.com/about/press/1988/1116n.asp.

[243] J.R. Vallino. Augmenting reality with minimal calibration. ftp://ftp.cs.rochester.edu/pub/u/vallino/proposal.ps.gz, 1997.

[244] A. Vardy, J. Robinson, and L. Cheng. The WristCam as input device. In *Proceedings of the Third International Symposium on Wearable Computers*, pages 199–202, San Francisco, USA, October 18-19 1999.

[245] Various. Society in the year 2005 - wearables in 2005, July 18-19 1995. http://web-ext2.darpa.mil/ETO/Displays/Wear2005/TOC.html.

[246] ViA. ACW farms chooses a high-tech solution. Technical Report 1010-01 5/00, ViA Inc., 2000. http://www.via-pc.com/product/Images/farm.pdf.

[247] ViA. Body-worn PC increases surveying efficiency. Technical Report 1014-00 6/00, ViA Inc., 2000. http://www.via-pc.com/product/Images/survey2.pdf.

[248] ViA. Northwest airlines flying high with "line busting" solution. Technical Report 1017-01 11/00, ViA Inc., 2000. http://www.via-pc.com/product/Images/customer.pdf.

[249] ViA. Northwest airlines high-tech solution speeds up maintenance. Technical Report 1012-01 5/00, ViA Inc., 2000. http://www.via-pc.com/product/Images/Maintenance.pdf.

[250] ViA. Shipbuilder trims inspection and troubleshooting time by 70%. Technical Report 1013-01 11/00, ViA Inc., 2000. http://www.via-pc.com/product/Images/Industrial.pdf.

[251] ViA. ViA s wearable PC is a cool solution for fire fighters. Technical Report 1026-00 3/01, ViA Inc., 2000. http://www.via-pc.com/product/Images/wffe.pdf.

[252] ViA. Wearable PC boosts productivity of home inspector. Technical Report 1008-01 5/00, ViA Inc., 2000. http://www.via-pc.com/product/Images/inspect.pdf.

[253] R. Wagner, F. Liu, and K. Donner. Robust motion estimation for calibrated cameras from monocular image sequences. *Computer Vision and Image Understanding*, 73(2):258–268, February 1999.

[254] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

[255] A. Webster, S. Feiner, B. MacIntyre, W. Massie, and T. Krueger. Augmented reality in architectural construction, inspection, and renovation. In *Proceedings of the ASCE Third Congress on Computing in Civil Engineering*, pages 913–919, Anaheim, California, June 17-19 1996.

[256] S. Weghorst. Augmented reality and Parkinson's disease. *Communications of the ACM*, 40(8):47–48, August 1997.

[257] M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84, July 1993.

[258] J. Yang, W. Yang, M. Denecke, and A. Waibel. Smart sight: A tourist assistant system. In *Proceedings of the Third International Symposium on Wearable Computers*, pages 73–78, San Francisco, USA, October 18-19 1999.

[259] L. Yencharis. 32-bit Active X controls: Yes, they do reduce app development time. *Advanced Imaging*, 13(2):71, February 1998.

[260] I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2D mosaic from a set of images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 97)*, pages 420–425, San Juan, USA, June 17-19 1997.

[261] A. Zomet and S. Peleg. Efficient super-resolution and applications to mosaics. In *Proceedings of the International Conference on Pattern Recognition*, Barcelona, Spain, September 3-8 2000.

[262] M. Zuke and S. Umbaugh. CVIPtools: A software package for computer imaging education. *Computer Applications in Engineering Education*, 5(3):213–220, 1997.