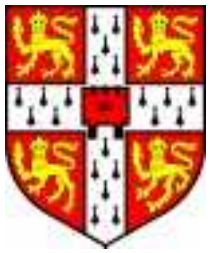
Sensor-driven Computing

Andrew Martin Robert Ward

Corpus Christi College
University of Cambridge



A dissertation submitted for the degree of
Doctor of Philosophy

August 1998

Abstract

A context-aware computing system is one that can deduce the state of its surroundings using input from sensors and can change its behaviour accordingly. Context-aware devices might personalise themselves to their current user, alter their functionality based on where they were being used, or take advantage of nearby computing and communications resources.

Location-aware systems, whose behaviour is determined by the positions of objects in the environment, represent a practical subset of the context-aware computing paradigm, and several systems of this nature have already been demonstrated. The location sensors used by those systems, however, report the positions of objects to only a room-scale granularity, limiting the extent to which devices and applications can adapt to their surroundings. Sensor technologies that can provide more detailed information about the locations of objects must therefore be investigated.

This dissertation describes a new ultrasonic location sensor, which may be deployed in indoor environments such as offices and homes. The sensor can provide fine-grain, three-dimensional position and orientation information, and its characteristics are well suited to the demands of location-aware computing—the sensor is simple, low-powered and unobtrusive. Furthermore, the location system is scalable, in both the number of objects that it can track and the volume within which they may be monitored. A thorough assessment of the sensor's performance is presented in the dissertation, so that location-aware applications can be tailored to its properties.

Subsequently, a software architecture that can efficiently distribute fine-grain location information to applications is described. The software system provides support for the types of query that will be made frequently by location-aware applications, such as those concerning the spatial relationships between objects and their proximity to one another. The dissertation concludes by examining the use of the ultrasonic location sensor and software architecture to implement a set of novel location-aware applications.

Preface

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

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

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

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

Acknowledgements

I would like to thank my supervisor, Andy Hopper, for his guidance and encouragement during my research. Thanks are also due to Alan Jones of the Olivetti and Oracle Research Laboratory, Cambridge (ORL), whose advice and willingness to participate in challenging discussion were invaluable.

I am indebted to the numerous colleagues and friends who provided inspiration and assistance during the course of this work, notably Mike Addlesee, Frazer Bennett, Dave Clarke, Andy Harter, Steve Hodges, David Leask, John Naylon, Pete Steggles, Paul Webster and Charlanne Scahill. I would also like to thank my family for their love, support and understanding.

I am grateful to Andy Harter, Steve Hodges, Alan Jones, Antony Rowstron, Ferdi Samaria, Pete Steggles and Ken Wood, who reviewed earlier drafts of this dissertation and made many useful comments.

This work was generously supported by the EPSRC and ORL, by means of a CASE award.

Contents

1 Introduction **1**

1.1 Sensor-driven Computing Paradigms 2
1.2 Research Motivation 2
1.3 Research Statement 3
1.4 Dissertation Outline 3

2 Location-aware Computing **5**

2.1 Introduction 5
2.2 Previous Work 5
2.2.1 Enabling technologies 5
2.2.2 Management of location information 7
2.2.3 Application areas 8
2.2.4 Review 9
2.3 Enhancing Location-aware Systems 10
2.3.1 Location sensor accuracy requirements 10
2.3.2 Other desirable sensor characteristics 11
2.4 Summary 12

3 Location Technologies **13**

3.1 Introduction 13
3.2 Principles 13
3.2.1 Tagged and untagged location systems 13
3.2.2 Location by containment 14
3.2.3 Location by positioning 14
3.3 Candidate Location Technologies 16
3.3.1 Optical tracking systems 17
3.3.2 Radiolocation systems 19

3.3.3 Electromagnetic tracking systems 20
3.3.4 Other tracking techniques 21
3.3.5 Review 22
3.4 Ultrasonic Positioning Technologies 22
3.4.1 Fundamentals of ultrasonic multilateration 22
3.4.2 Existing ultrasonic tracking systems 27
3.4.3 Outstanding problems 28
3.5 Summary 28

4 Indoor Ultrasonic Tracking **29**
4.1 Introduction 29
4.2 System Description 29
4.2.1 System entities 29
4.2.2 Coordination of system components 30
4.2.3 Processing of ultrasonic pulses 30
4.2.4 Multilateration considerations 31
4.2.5 Placement of receivers 31
4.2.6 Emission of successive pulses 32
4.2.7 Power management 33
4.2.8 Concluding remarks 34
4.3 Determining Object Locations 34
4.3.1 Measurement of MPD-receiver distances 34
4.3.2 Multilateration using nonlinear regression 36
4.3.3 Eliminating reflections 37
4.3.4 Effects of geometry on position accuracy 39
4.3.5 An alternative multilateration approach 40
4.4 Determining Object Orientation 42
4.4.1 Rigid-body approach 42
4.4.2 Directional beam approach 44
4.5 Summary 47

5 A Prototype Tracker **48**
5.1 Introduction 48
5.1.1 Overview 48
5.2 Hardware Implementation 48
5.2.1 Global Clock Generator 48
5.2.2 Zone Manager 49

5.2.3 Mobile Positioning Device 50
5.2.4 Receiver 52
5.2.5 Matrix Manager 54
5.3 Software Implementation 55
5.3.1 Timeslot allocation 56
5.3.2 Multilateration calculation 56
5.3.3 Determination of MPD positions 57
5.4 System Calibration 57
5.4.1 Delay calibration 57
5.4.2 Surveying receiver positions 59
5.4.3 Ceiling self-calibration 59
5.5 Tracker Evaluation 60
5.5.1 Assessment of distance measurements 60
5.5.2 Validation of regression assumptions 61
5.5.3 Assessment of position measurements 62
5.5.4 Assessment of orientation measurements 70
5.5.5 Effects of environmental noise 72
5.5.6 Power management evaluation 73
5.5.7 Safety evaluation 76
5.6 Summary 77

6 System Scalability **78**
6.1 Introduction 78
6.2 Resource Allocation 78
6.2.1 Scheduling 78
6.2.2 Group allocation 79
6.2.3 Power management 80
6.3 Dynamic Population Management 82
6.3.1 Registration 82
6.3.2 Resource reclamation 83
6.4 Multiple Ultrasonic Spaces 85
6.4.1 Tracking MPDs in multiple rooms 85
6.4.2 Increasing the system capacity 86
6.4.3 Large ultrasonic spaces 89
6.5 Multiple Radio Cells 90
6.5.1 Multiplexing schemes 91

6.5.2 Scalable system organisation 92
6.5.3 Scalable system operation 93
6.5.4 Handovers between radio cells 95
6.5.5 Group handovers 97
6.6 Summary 98

7 Software Architecture **99**
7.1 Introduction 99
7.2 Architecture 99
7.2.1 Object transformations 100
7.2.2 Object interfaces 102
7.2.3 Controlling location rates 103
7.2.4 Resource information 103
7.2.5 Sensor fusion 104
7.3 Spatial Indexing 105
7.3.1 Principles 106
7.3.2 Integration 108
7.3.3 Approximate spatial relationships 110
7.3.4 Containment queries 110
7.3.5 Related work 111
7.4 Proximity Queries 111
7.4.1 Computing shortest paths 111
7.4.2 Query implementation 112
7.4.3 Continuous queries 116
7.4.4 Related work 118
7.5 Historical Queries 118
7.6 Summary 118

8 System Realisation **119**
8.1 Introduction 119
8.2 Location System Implementation 119
8.2.1 System parameters 119
8.2.2 Synchronisation of system components 120
8.2.3 Location system management 121
8.2.4 MPD hardware 122
8.2.5 Radio message formats 123
8.2.6 MPD operation 123

8.2.7 Evaluation . . . 129
8.3 Software System Implementation . . . 129
8.3.1 Object implementations . . . 130
8.3.2 Callbacks . . . 130
8.3.3 Spatial indexing implementation . . . 131
8.4 Summary . . . 133

9 Applications **134**
9.1 Introduction . . . 134
9.2 Future Vision . . . 134
9.2.1 Location-aware application areas . . . 135
9.2.2 System limitations . . . 138
9.2.3 Acceptability of invasive technologies . . . 138
9.3 Working Prototypes . . . 139
9.3.1 Moving indoor map . . . 139
9.3.2 Workstation personalisation . . . 140
9.3.3 Construction-kit computing . . . 143
9.3.4 Automatic camera selection . . . 144
9.4 Summary . . . 146

10 Conclusion **147**
10.1 Summary . . . 147
10.2 Further Work . . . 149

A The Position Dilution Of Precision **152**

B A Simple Scheduling Algorithm **156**

Bibliography **161**

List of Figures

3.1 Multilateration in three dimensions 15
3.2 Distance measurements in the presence of reflected pulses . . 26

4.1 Operation of system using timeslots 33
4.2 Effects of varying MPD-receiver geometry 41
4.3 A directional ultrasonic signal formed by an opaque object . . 44
4.4 Determining the direction in which a person is facing 45

5.1 Overview of tracking system 49
5.2 Overview of Zone Manager 50
5.3 External and internal views of an MPD 51
5.4 MPD operating state diagram 52
5.5 Ultrasonic pulse emitted by MPDs 53
5.6 An installed receiver 54
5.7 Overview of Matrix Manager 55
5.8 Delay calibration results for five receivers 58
5.9 Accuracy of receiver distance measurements 61
5.10 Standard deviation of receiver distance measurements 62
5.11 Validation of regression assumptions 63
5.12 Region used for measurement of position accuracy 64
5.13 Accuracy of ultrasonic tracker 65
5.14 Variation of tracker accuracy with height above floor 66
5.15 Effects of averaging of readings on tracker accuracy 67
5.16 Distortion of tracker measurement volume 68
5.17 Spatial correlation of measurement space distortions 69
5.18 Temporal correlation between position measurements 70
5.19 Volume containing position measurements from one test point 71
5.20 Variation of measurement error with PDOP 71

5.21 The effects of filtering using PDOP 72
5.22 Test frame for orientation measurements 73
5.23 Accuracy of orientation measured using multiple MPDs . . . 74
5.24 Accuracy of orientation measured using a single MPD 75
5.25 Power spectrum of ambient noise detected by receivers 75

6.1 Timeslot swapping 80
6.2 Behaviour of MPDs in sleep state 81
6.3 Registration protocol 84
6.4 Resource reclamation 85
6.5 Tracker operation with several receiver matrices 86
6.6 Staggered triggering of MPDs 88
6.7 Identification of virtual ultrasonic spaces in a large room . . . 90
6.8 Scalable system overview 92
6.9 System in which only one cell colour is active 94
6.10 Active handover 96
6.11 Passive handover 97

7.1 Object model overview 101
7.2 Quadtree data structure 106
7.3 Maximal cover of a shape 107
7.4 Integration of the spatial index with the object model 109
7.5 Modelling one floor of a building as the interior of a polygon 113
7.6 Modelling a building as a set of 2D floors 113
7.7 Simplifying bounded proximity queries 114
7.8 Propagating a proximity query between floors 115
7.9 Simplifying unbounded proximity queries 116

8.1 Prototype system deployment 120
8.2 Addressing message format 124
8.3 Registration message format 125
8.4 Simplified MPD operation 125
8.5 Simulated registration performance 128
8.6 Spatial index performance 132
8.7 Spatial index performance with varying space population size 133

9.1 External and internal views of a miniaturised MPD 135
9.2 A moving indoor map 140

9.3 Spatial index entries for workstation personalisation 141
9.4 Personalising a workstation 142
9.5 Spatial events driving personalisation 142
9.6 Construction-kit computing 144
9.7 Automatic camera selection 145

List of Tables

5.1 Values of constants used for reflection elimination 57
5.2 Accuracy of surveyed receiver positions 59

6.1 Function for identifying MPD trigger times 89

8.1 Registration transmission probabilities 128

Glossary

ADC	Analogue-to-Digital Converter
AM	Area Manager
API	Application Programming Interface
AR	Augmented Reality
ATM	Asynchronous Transfer Mode
CDMA	Code-Division Multiple Access
CORBA	Common Object Request Broker Architecture
CRC	Cyclic Redundancy Check
CRT	Cathode-Ray Tube
CTI	Computer Telephony Integration
DBMS	DataBase Management System
FDM	Frequency-Division Multiplexing
FFT	Fast Fourier Transform
FM	Frequency Modulation
FPGA	Field-Programmable Gate Array
GCG	Global Clock Generator
GIS	Geographic Information System
GPS	Global Positioning System

HDOP	Horizontal Dilution Of Precision
HTML	HyperText Markup Language
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LID	Location IDentifier
LIDAR	LIght Detection And Ranging
LQoS	Location Quality of Service
LVDS	Low Voltage Differential Signalling
MM	Matrix Manager
MPD	Mobile Positioning Device
OM	Object Manager
PDA	Personal Digital Assistant
PDOP	Position Dilution Of Precision
RF	Radio Frequency
SPL	Sound Pressure Level
SPM	Shortest Path Map
TCP	Transmission Control Protocol
TDM	Time-Division Multiplexing
UHF	Ultra High Frequency
VDOP	Vertical Dilution Of Precision
VNC	Virtual Network Computing
VR	Virtual Reality
VRML	Virtual Reality Modelling Language
ZM	Zone Manager

Chapter 1

Introduction

Many indoor environments, such as homes and offices, are rich in computing and communications resources. It is likely that the density of devices in these surroundings will increase in the future, as advances in digital electronics make computers smaller, cheaper and lower powered, and networks, both wired and wireless, become more commonplace. Extrapolating these trends forward, computing devices may one day become as common, and perhaps as invisible to general awareness, as ball-point pens and paper clips, and high-bandwidth communications between those systems will be taken for granted.

In a world where computing and communications systems have been deployed to this extent, the manner in which people will interact with them will change. The configuration of such systems currently requires significant effort, specialist knowledge and the attention of the user. Although the complexity of the computing and communications fabric will increase in the future, it may be possible to hide this complexity by transferring some of the configuration burden to the devices themselves, maximising their utility whilst minimising their requirements of users. Devices will not be specialised to particular tasks, but will be capable of changing their behaviour based on how and why they are being used. The term *ubiquitous computing* [Weiser91] has been used to describe the experience of interacting with systems of this nature.

1.1 Sensor-driven Computing Paradigms

Recently, researchers have begun to investigate computers that can autonomously change their functionality based on observations of the world around them. By determining their context, using input from sensor systems distributed throughout the environment, these computing devices can personalise themselves to their current user, adapt their behaviour according to their location, or react to their surroundings. This paradigm is called *context-aware computing* [Schilit95a], and it represents a step towards the ubiquitous computing vision described above.

A key component of context-aware systems is a dynamic model of the environment, which devices and applications can use to drive changes in their behaviour. Many different qualities of objects (e.g. motion, weight, temperature) can be monitored by computer in order to build up this model, but, in general, each quality is measured by a different sensor system. As the number and diversity of sensors in an environment increases, the physical infrastructure required to support them (e.g. wiring and interfaces) becomes more complex, and the problem of managing and collating different types of sensor information arises. It would seem desirable, therefore, to investigate whether information gathered using a single type of sensor is sufficient to effect useful changes in the operation of context-aware systems.

Research into context-aware computing has shown that monitoring of the locations of objects is a feasible method of determining the state of an environment. By accurately tracking the locations of objects it is also possible to derive other useful information, such as their orientation and how they are moving. *Location-aware* systems, whose behaviour is controlled by these types of data, embody a practical subset of the context-aware computing paradigm, and several systems based on this principle have already been developed.

1.2 Research Motivation

The sensors that support existing location-aware computing systems generate relatively coarse information, with object positions reported to the granularity of rooms. This dissertation asserts that the lack of detail inherent in this data limits the range of location-aware applications that can

be developed from it. Many potential applications have remained uninvestigated, and it is therefore pertinent to consider sensor technologies that might provide more fine-grained location information about objects in indoor environments.

1.3 Research Statement

This dissertation investigates a new indoor location sensor. Unlike other technologies, the sensor generates location information with a fine spatial granularity ($\approx$ 10cm), yet has characteristics that allow it to be integrated into existing computing environments. An associated software architecture enables efficient distribution of fine-grain location information to applications, and offers support for the types of query that location-aware applications are likely to make. Interoperation of the sensor hardware and software architecture is demonstrated by the use of those components to implement a set of novel location-aware applications.

1.4 Dissertation Outline

Chapter 2 examines the current state of location-aware computing research, identifies the need for a sensor that can provide fine-grain location information, and sets out the characteristics required of that sensor.

Chapter 3 considers location technologies that might be used as a basis for the new sensor. The chapter concentrates on ultrasonic schemes, which appear to be the most promising technology in the context of location-aware systems.

Subsequently, Chapter 4 describes the principles of operation of an ultrasonic tracker designed specifically for use in location-aware computing environments, and Chapter 5 discusses the implementation and evaluation of that system.

Chapter 6 addresses several issues related to the scalability of the tracking system, which should be able to locate many objects in very large buildings. Chapter 7 then presents a software architecture, intended for use with the ultrasonic tracker, that performs data and query management functions.

Chapter 8 describes the implementation of a scalable location system that provides fine-grain position and orientation information, based on the

work presented in Chapters 4–7.

Chapter 9 considers the types of applications that might be made possible by the availability of detailed indoor location information. Several prototype applications that have been implemented using the new location system are also described.

Finally, concluding remarks and suggestions for further work are made in Chapter 10.

Chapter 2

Location-aware Computing

2.1 Introduction

The purpose of this chapter is twofold. First, a review of location-aware computing research that has been described in the literature is given. These studies highlight both the potential of location-aware computing, and also limitations on the range of applications that can be implemented using current sensor technologies. The second part of the chapter considers the properties that might be possessed by an improved location sensor for context-aware computing.

2.2 Previous Work

This section describes existing information sources for location-aware systems, and software architectures that have been designed to manage location data. Location-aware applications that make use of these technologies are also discussed.

2.2.1 Enabling technologies

Much of the previous research into location-aware computing has used tracking information provided by *Active Badges* [Want92a,b], which are small, battery-powered computing devices worn by personnel. Each badge measures 5.5cm $\times$ 5.5cm $\times$ 0.7cm, weighs 40g, and has a globally unique code, which is broadcast through an infrared interface every ten seconds. The infrared signals reflect off walls and furniture to flood the surrounding area,

and are picked up by a network of sensors placed around the building. By determining which badges are seen by which sensors it is possible to deduce the location of a badge, giving a hint to the location of the badge's owner. Information from the sensors is gathered by a location server, which client applications can contact as part of a distributed system. Badges have a bidirectional infrared interface, as well as two pushbuttons, a speaker and two visible LEDs, allowing them to act as simple signalling and paging devices. The battery lifetime of the devices is around one year. The *Active Office* [Harter94] is a development of the Active Badge technology, in which *Equipment Tags* consisting of only the transmitting components of a Badge are attached to equipment such as workstations and printers. Equipment is assumed to move less frequently than personnel, so Tags beacon only once every five minutes, extending their battery lifetime to several years.

The *ParcTab* [Adams93] is a Personal Digital Assistant (PDA) that uses an infrared-based cellular network for communication. The infrared transmissions from ParcTabs can be used to determine the locations of those devices in the same way as Active Badges are located. ParcTabs have relatively substantial input and output facilities (including a touch-sensitive 128×64 pixel display, a piezoelectric buzzer, and several push-buttons), and act as thin display clients for application agents hosted by fixed machines on a wired network.

Beadle *et al.* have described a *Smart Badge*, which combines the tracking function of an Active Badge with output and environmental sensing capabilities [Beadle97]. Like Active Badges, each Smart Badge has a unique identifier, which is periodically broadcast across an infrared interface to networked sensors placed around buildings. Information from a range of sensors on the Smart Badge (which measure temperature, light level, etc.) is encoded into the identification signal, providing further information about the state of the Badge's environment. Smart Badges also have an output port to which PDAs can be attached, and to which data can be sent from the fixed sensor network.

A tracking system for wearable computers called the *Locust Swarm* is described by Kirsch and Starner [Kirsch97]. A set of fixed beacons are placed at known points within a building, and periodically transmit an infrared message containing a unique identifier. Those signals are detected by a receiver on the wearable computer, which can use them to deduce its location.

The wearable computer may then choose to make this location information available to other devices across a wireless network, depending upon the extent to which the user wishes this data to be released to other parties.

2.2.2 Management of location information

A tracking system that operates within an office environment may generate large quantities of location information. For example, if one hundred persons within an organisation carry an Active Badge, around 300,000 position updates for those Badges will be generated each day. Suitable software architectures and mechanisms for the management of this volume of location information have been investigated. Furthermore, location-aware applications are likely to make certain types of query relatively often, and researchers have examined software architectures that can provide efficient support for those query types.

Harter and Hopper describe a distributed location system that processes Active Badge information [Harter94]. The authors also discuss the merits of different distribution models for location system architectures, comparing centralised services that have corporate responsibility for handling location data with agent approaches, which deal with location information on a per-physical-object basis. Software architectures that use agents to control access to location information about personnel are discussed by Schilit [Schilit95b], and Spreitzer and Theimer [Spreitzer94]. In [Schilit93], Schilit *et al.* describe a mechanism for application customisation in which sensor information is translated into name-value pairs (*environments*) managed by *dynamic environment servers*. Applications may receive asynchronous notification of changes in environments from servers, allowing them to adapt their behaviour accordingly.

Schilit has considered mechanisms for evaluating proximity and containment queries in location systems that deal with Active Badge and ParcTab data [Schilit95b]. Proximity queries are serviced by traversing a weighted graph that defines the lengths of possible paths between locations. Containment queries, such as "Is Person X in the same building as Person Y?", are answered by comparing *location identifiers* (LIDs) used to describe object locations. An LID indicates an object or region's position within a location hierarchy that describes domains, with buildings containing floors, which contain rooms, and so on. Textual comparisons between substrings of LIDs

can determine the depth in the hierarchy to which the locations of their associated objects or regions are identical. This information can then be used to evaluate whether an object is at or within a place, or whether it is with another object.

2.2.3 Application areas

Active Badge information has been used in a wide range of applications, including telephone call routing and security [Want92a,b], and environmental control [Elrod93]. Harter and Hopper describe a "nearest printer" service offered to users of portable computers [Harter94]. Equipment Tags placed on the computer and printers report their positions, and the computer is automatically configured to use the nearest available printer as it is moved around a building.

Another application of Active Badge information is the *Teleporting System* [Richardson94], which allows a user to redirect their X Window System environment to different computer displays. The system can find personnel wearing Active Badges, and the set of computer displays located with them can be determined using information obtained from Equipment Tags. Personnel can redirect their windowing environment to one of these displays by using the buttons on their Active Badge, which acts as a ubiquitous authentication and control device. When a user leaves the location in which their windowing environment is displayed, the Teleporting System causes it to disappear to prevent unauthorised access.

Similar ideas have been proposed by other researchers. A framework for supporting location-aware multimedia applications is described by Bacon *et al.* [Bacon97]. Mobile agents containing application code and execution state may be moved between hosts, following their users (who are tracked using Active Badges) around a building. Multimedia stream endpoint agents, such as video and audio sources, may also be transferred between devices in this way—for example, a mobile music stream could follow its user by automatically selecting a new loudspeaker as an output device when they moved into a different room. The *FLUMP* (FLexible Ubiquitous Monitor Project) system uses a set of wall mounted monitors to present information such as e-mail and diary entries to users [Finney96]. Active Badge location data controls the migration of information from one display to another, based on the positions of users.

Schilit *et al.* describe the use of the ParcTab system to implement applications involving context-triggered actions and automatic reconfiguration [Schilit95a]. An example is the *Scoreboard* application, which is displayed in a public area of a building and shows items of interest to those around it—the latest football scores, for example. The authors also suggest user interface designs for situations in which manual configuration of applications based on context information is appropriate.

The ParcTab has been used to implement a memory prosthesis, which collects information about the user's context and organises it to form a biography [Lamming94]. Later, the user, who may remember only some of the detail of an event in which they are interested, can query the biography to retrieve the details that had been forgotten. Brown *et al.* have used ParcTabs to prototype *stick-e notes* [Brown97], which are context-aware electronic equivalents of post-it notes. Stick-e notes have two components—their content, which may be text, an HTML page, multimedia content or even a program, and a context in which the content should be displayed or executed. A potential application of this technology is a tour guide for a museum, which shows stick-e notes associated with nearby exhibits. Similarly, the *Audio Aura* system described by Mynatt *et al.* [Mynatt97] presents location-dependent auditory cues to users who wear wireless headphones and Active Badges.

2.2.4 Review

Systems like the Active Badge and the ParcTab are robust, relatively cheap, and can be integrated into everyday working environments. However, they locate objects only to the granularity of rooms, which act as natural containers of the infrared signals emitted by the devices. This limits the extent to which applications can change their behaviour based on information from the system. The Teleporting application, for example, cannot know on which of the screens in a room the user intends to display their desktop—the user is required to select the correct screen by repeatedly pressing their Badge button. This behaviour would be unreasonable in a large open-plan office with hundreds of displays. If a more detailed dynamic model of the environment was available, however, the application could initially select the screen directly in front of the user. It is therefore pertinent to consider how such a model might be constructed.

2.3 Enhancing Location-aware Systems

The resolution and accuracy of an environmental model built up using location data depends upon the properties of the sensor generating that information. If more complex location-aware applications are to be implemented, it will be necessary to develop more sophisticated location sensing technologies. This section examines the properties that might be possessed by a sensor intended for use in enhanced location-aware environments.

2.3.1 Location sensor accuracy requirements

The accuracy required of the environmental model (and, hence, of the location sensor generating it) is dependent upon the demands of the applications that it must support. A representative location-aware application that cannot be implemented using current technology is a *walk-through videophone*. Suppose that many cameras, microphones, displays and loudspeakers are distributed around an office. When a user starts the videophone application, it should select input and output devices with which they can interact. For example, the application should, if possible, choose a camera that can see the user and that the user is facing. Furthermore, the callee's image should be shown on a suitable display, i.e. the display must face the user, and the user must face the display. As the user walks around the room, the application continuously monitors their location in relation to the available input and output facilities, and reconfigures itself automatically.

This example suggests that object positions must be found to within a few tens of centimetres, to enable spatial relationships such as "Person X is in the field of view of Camera Y" to be identified. It also suggests that knowledge of the orientations of objects would significantly enhance the location-aware computing experience.

As the information gathered by a location sensor becomes more fine-grained, the spatial relationships between objects can be more accurately identified, allowing a wider range of location-aware applications to be implemented. It should be noted, however, that accuracy is only one of many desirable location sensor properties (others, such as update rate and scalability are discussed below in Section 2.3.2). Location sensors with the very highest accuracy may be completely unsuitable for context-aware computing, because their performance in other areas is compromised by this focus.

For this reason, the requirements of *Virtual Reality* (VR) and *Augmented Reality* (AR) systems [Azuma97], which place a user wearing a head-mounted display in a partially or completely immersive environment, are not considered in this dissertation. These applications have perhaps the highest demands for accurate real-time location information, to prevent users from experiencing motion sickness. The head-mounted displays are typically tracked with accuracies of a few millimetres in position and one degree in orientation. It is likely that any location sensor that could support VR or AR work would have characteristics that limited its suitability for the great majority of other location-aware applications, which do not require such high accuracy.

2.3.2 Other desirable sensor characteristics

To be a useful source of information for context-aware computing, a location sensor system should additionally have many or all of the following properties:

- **High location rate:** The high spatial accuracy of location information provided by the sensor should be matched by a high update rate, so that processes of change in the environment are modelled in detail.
- **Wide-area coverage:** The sensor system should be able to find objects in all rooms of a large building, e.g. a hospital, allowing location-aware applications to be deployed throughout that environment.
- **Tracking of many objects:** To ensure that the environmental model is complete as well as accurate, the locations of many objects should be monitored. In a typical office, hundreds of personnel might be located by the system, and thousands of items of equipment might be tracked.
- **Easy integration into existing environments:** Tags (either active or passive) placed on objects must be small, lightweight, and wireless, so that motion of the objects is not hindered. Sensors or signal sources placed in the environment should be unobtrusive, and the location system should be safe to use.
- **Low power consumption:** Any components of the location system that are battery-powered (e.g. Active-Badge-like devices) must have

low power consumption, to minimise the logistical task of maintaining those components.

- **Support for applications:** The basic sensing technology used by the location system should be integrated with a software support structure that permits easy access to location data and convenient manipulation of that information.

- **Low cost:** The cost of installing and operating the sensor system must not be prohibitive.

As discussed previously, a sensor that maximises any one particular aspect of system performance may well be forced to make compromises in other areas. An improved location sensor for context-aware computing will therefore represent some reasonable compromise between all these properties.

2.4 Summary

This chapter has examined the current state-of-the-art in location-aware computing research. A number of indoor location systems have already been constructed, and several interesting applications have been developed using them. However, it has been argued that the availability of finer-grain location and orientation information would allow the implementation of novel applications, such as the walk-through videophone. A new location sensor, whose characteristics are set out in Section 2.3, is required to provide this information.

Chapter 3

Location Technologies

3.1 Introduction

This chapter investigates technologies that might be used to gather fine-grain location information about objects in indoor environments. First, a set of fundamental methods that can be used to determine the location of objects are described. Properties of existing location systems are then examined to assess their suitability as information sources for location-aware computing.

3.2 Principles

This section develops a classification of location technologies into tagged and untagged, container-based and position-based systems, and describes the characteristics of those systems.

3.2.1 Tagged and untagged location systems

Tagged location technologies locate a marker (either active or passive) associated with an object, thus allowing the object's position to be deduced. *Untagged* systems are those which determine the positions of objects directly. In the context of location-aware computing, untagged methods have the desirable property that disruption to the existing environment is minimised. However, the markers associated with objects by tagged systems make the task of finding and distinguishing objects easier, especially when the markers are coded with identification information in some way.

3.2.2 Location by containment

Containment-based location systems determine the positions of objects by identifying spatial regions that contain those objects. The size of the spatial containers governs the resolution with which the locations of objects can be found—smaller containers correspond to higher location resolution.

Active Badges are a good example of a containment-based tracking technology, with objects located to room-scale resolution. In an extension to the system called *Desk Scale Location* [Harter94], low-power radio fields, which are pulse-width modulated with unique patterns, are set up within rooms. Field strengths and antenna sizes are adjusted so that the fields form zones surroundings desks and doors. A passive tuned circuit in the Badge detects the field (if the Badge is in a field zone), and the field identifier is transmitted as part of the Badge's infra-red message. The Badge can then be located to a finer spatial granularity, with field zones acting as additional containers. *TIRIS* tags [Kaiser95], batteryless transponders that can be uniquely identified when located within a few metres of a radio interrogator unit, and cellular telephones [Samfat95] can also provide containment location information.

3.2.3 Location by positioning

Position-based location systems track objects by reporting their coordinates in a frame of reference. The actual position measurements may be made in many ways. Object (or marker) locations can be sensed directly, using measurements of fields emanated or affected by them, or by contact methods. Alternatively, locations can be deduced from measurements of other physical properties of the object. *Inertial* schemes determine the acceleration or velocity of objects, and integrate those measurements to obtain their positions relative to initial points. *Triangulation* systems use measurements of the bearings of objects from known, fixed points to find their locations. Similarly, *trilateration* (or *multilateration*) schemes find object positions by determining their distances from fixed points. Figure 3.1 shows how the position in 3D space of a transmitting device may be determined by measuring its distance from four receivers fixed at known locations. In general, to find the 3D position of an object by multilateration, distances must be measured to it from four known, non-coplanar points.

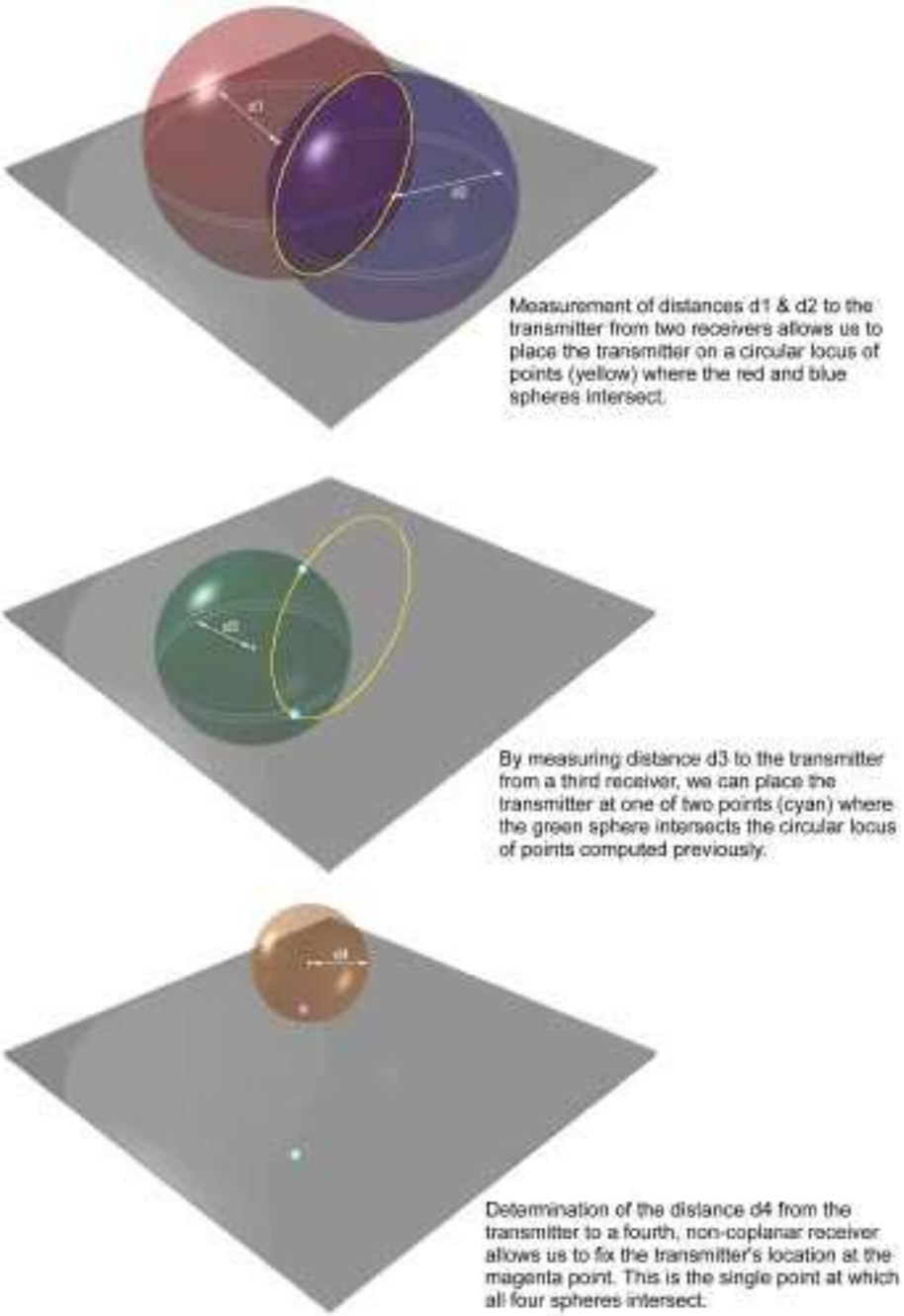


Figure 3.1: Multilateration in three dimensions

There are several ways of stating the accuracy of a 3D positioning system. It is normally impossible to guarantee that any particular position reading will lie within a specified distance of the true location, so accuracy figures are often associated with a confidence level. For example, if 75% of a system's position readings are within 10cm of the true location it would be said to have an accuracy of 10cm at 75% confidence. Commonly-used 3D position confidence levels are 50% (also known as *Spherical Error Probability*, or SEP), and 95%.

An alternative measure estimates either the one-sigma or two-sigma accuracy level of a position system, where "sigma" is the standard deviation of the position error distribution. The one-sigma and two-sigma figures take account of occasional, large position errors, unlike the SEP and (to a lesser extent) 95% confidence figures, which may be overly optimistic. If the position error distribution is approximately normal, the one-sigma level will encompass between 61% and 68% of the readings, and the two-sigma level will encompass between 95% and 99% of readings [Leick95][NIMA96]. Meaningful comparisons between positioning systems whose accuracies are specified in different ways can therefore be made.

3.3 Candidate Location Technologies

This section examines location systems that have been described in the literature, in an effort to identify techniques that might be used to gather information for use by location-aware applications. Many different location technologies have been developed, and no attempt is made to describe all of them in detail—instead, a representative sample will be considered. Other researchers have performed similar surveys in the contexts of mobile robot positioning [Feng94], human motion studies [Mulder94], military head-up displays [Ferrin91] and VR/AR [Bhatnagar93][IDA96][Azuma97].

The discussion of location-aware computing in Section 2.3.1 sets out a desire for very fine-grain location information. Containment-based location systems can only supply data with this resolution if a large number of spatial containers are physically generated and subsequently maintained. The management of a container-based location system of this kind would be unreasonably difficult and, therefore, only position-based location systems are considered as candidate location technologies for context-aware computing.

Location systems that rely fundamentally on mechanical linkage between an object and sensors (such as head-trackers based on jointed mechanical arms) cannot satisfy the requirement for unhindered object motion, and are also excluded from consideration.

3.3.1 Optical tracking systems

Many optical positioning systems have been proposed. All require a line-of-sight between a target (which may be passive or active) and a detector. This limitation may be restrictive in indoor environments that contain many opaque objects.

Outside-in videometric schemes

Outside-in videometric systems use a set of cameras placed at static points in the environment to monitor objects within that environment. Ishii *et al.* describe a sensor for tracking a robotic arm [Ishii84]. A infrared-sensitive video camera is placed at a known, fixed point, and views the scene containing the arm, to which a number of infrared-emitting LEDs are attached in a known pattern. The positions of the LEDs in the image are determined, and this information, together with knowledge of the 3-D relationship between the LEDs, allows the position and orientation of the robotic arm to be found. Average accuracy of the system is said to be within 3mm in position, and 2° in orientation, although no confidence levels for these accuracies are given. Similar systems using passive retroreflective markers [Janin94] and 2D barcodes [Rekimoto98] have also been described in the literature.

The *Pfinder* tracker [Wren96] goes one step further, dispensing with all markers. Motion in video images is used to identify the presence of a user, who is then tracked using a colour and shape model. Unfortunately, many objects of interest to location-aware applications, such as workstations and printers, are less mobile and distinctive than people. Furthermore, the task of determining *which* person is being tracked by the system is non-trivial.

Many cameras (or other expensive optical detectors) would be required to ensure comprehensive tracking coverage of an indoor environment by an outside-in videometric system, due to line-of-sight constraints. Moreover, as the number of cameras increases, there is a corresponding increase in the level of processing resources required to interpret their outputs.

Inside-out videometric schemes

Inside-out videometric systems use a set of cameras carried by an object to determine its position and orientation in relation to a set of static targets placed around the environment. A real-time 3-D tracker for use with head-mounted displays is described by Ward *et al.* [Ward92]. Three cameras, mounted on a helmet worn by the user, view an array of infrared-emitting LEDs affixed to the ceiling. A tracking controller individually illuminates the LEDs in turn—by determining which LEDs are actually seen by the cameras the helmet can be tracked with a resolution of around 2mm in position and 0.1° in orientation. However, the camera arrangement is bulky and is tethered to a control unit, and a very large number of LEDs must be accurately placed on the ceiling of the work area—the prototype head-tracker uses 960 LEDs to cover an area of 11m^2.

More unusual inside-out videometric methods include place recognition using image signatures [Engelson92], where the view from a mobile robot is compared with a large database of signatures generated by passing a measurement function over many views of the robot's environment, and the *Self-Tracker* [Bishop84], a baseball-sized cluster of outward-looking sensors. Each sensor contains a lens and an integrated circuit for imaging and image registration, and sensors distribute information regarding the views they see between each other. By comparing views, and noting how significant image features move between them, the sensors can determine how the cluster has moved from an initial reference point, and can thus deduce its position.

Optical triangulation schemes

The *Minnesota Scanner* [Sorensen89] is a system designed to track human body motion. Rotating mirrors scan three planes of laser light over the working volume (some 8m^3) of the system, and these are picked up by photodiodes placed on the body at the points to be tracked. By determining the positions of the planes when light is detected by the photodiodes, it is possible to calculate the three-dimensional position of the photodiode to within 2mm (95% confidence level). Yim and Seireg describe another laser-based system in which a tag with a special grid pattern is attached to the object to be tracked [Yim88]. A scanner picks up reflections of laser light from the tag, allowing the object's position and orientation to be determined. Un-

fortunately, these systems are unsuitable for the purposes of location-aware computing, due to line-of-sight constraints and the perceived safety hazards of lasers.

Optical trilateration schemes

Mäkynen *et al.* present results from a LIDAR (LIght Detection And Ranging) sensor designed for teaching robot paths [Mäkynen95]. A pointer on the robot is tracked to within 5mm in position and 5° in orientation by measuring the time-of-flight of a light pulse from radiators on the pointer to a fixed receiver. However, the high speed of light complicates the construction of the LIDAR rangefinder, and an optical fibre tether of known length is required to pass the distance measurement pulse to the radiators.

3.3.2 Radiolocation systems

Radio-based positioning systems are attractive because the signals they use can pass through solid objects. In indoor environments, however, signal reflections from objects are problematic. Murphy gives a good overview of the characteristics of radiolocation systems, and describes a community-wide radio surveillance technology for tracking offenders on parole [Murphy95].

Wide-area hyperbolic navigation systems, such as *Datatrak* [Banks89], the *Decca Navigator* system and *Loran* [Sonnenberg88] calculate the position of a receiver from observed phase differences between radio signals transmitted by synchronised beacons placed at known locations. Feuerstein and Pratt describe a local-area UHF radiolocation system that uses similar principles to determine three-dimensional object positions accurate to 1.3m (95% confidence level) within a $37m^3$ volume [Feuerstein89]. The authors also present results of computer simulations that estimate how the system will perform indoors in the presence of multipath effects.

The *Global Positioning System* (GPS) [Getting93] is based around 24 satellites in Earth orbit that transmit spread-spectrum radio signals, allowing a receiver anywhere on the globe to be located to within 100m horizontally and 156m vertically (95% confidence level) [NTIS94]. The system cannot be used indoors, because the frequencies at which the satellites transmit signals do not penetrate buildings. Zimmerman and Cannon describe an indoor GPS environment that uses fixed transmitters called *pseudolites* to

generate the required signals [Zimmerman94]. Centimetre-level accuracy is achieved using a differential carrier phase technique, but an auxiliary optical tracker is required to resolve possible position ambiguities.

An indoor radiolocation system called *3D-ID* [Lanzl98] measures two-way signal transit times between fixed base antennae and wireless tags placed on the objects to be tracked. Tags periodically modulate ID information onto signals received from the base antennae, and transmit the encoded signals back again. A central controller uses the round trip times of signals detected by base antennae to calculate the positions of tags to an accuracy of around 1m. The interval between transmissions from each tag can be varied according to application requirements—a tag carried by a person might transmit an encoded signal once a second, whereas one used for inventory purposes might transmit only once a minute.

Ernst describes radio direction finders for position fixing using triangulation in built-up areas [Ernst80], although both transmitters and receivers are bulky, and the system is only accurate to around 100m. Another radiolocation technique is used in a prison-guard alarm system [Christ93]. A network of receivers around a building monitor the signal amplitude of radio waves sent by portable transmitters—the authors claim that the system locates transmitters to an accuracy of 3–6m. Signals from mobile telephones can be used to locate callers requesting emergency services to better than 125m horizontally (67% confidence level) [IEEE98].

3.3.3 Electromagnetic tracking systems

Electromagnetic trackers determine the position and orientation of a receiving antenna with respect to a transmitting antenna that is fixed in space [Raab79]. Each antenna has three orthogonal field coils. The coils of the transmitter are driven using AC or pulsed DC signals, and currents thus induced in the receiver coils (which vary as a function of the relative position and orientation of the antennae) are measured and used to compute the tracking information.

The electromagnetic coupling between the transmitter and receiver antennae is weak, falling off with the cube of the transmitter-receiver distance. Strong electromagnetic fields and large transmitter coils must therefore be used if the receivers are to be compact. All electromagnetic systems are susceptible to interference from metallic objects in the environ-

ment, and are adversely affected by magnetic fields around CRTs [Nixon98]. Although compensation can be made for field distortions caused by fixed objects [Bryson92], such calibrations cannot be made for office and home environments, in which equipment and furniture are moved often.

An example state-of-the-art electromagnetic tracking system is the Polhemus *STAR*TRAK* tracker, which drives the transmitter coils with an AC signal. Up to 16 small ($3.5\text{cm} \times 3.5\text{cm} \times 1.2\text{cm}$), light (28g) sensors attached to an object are connected to a controller, which is also carried on the object. The sensors detect signals from a fixed transmitter, and the controller forwards the sensor measurements to a host computer across a wireless link. The computer can then calculate the position and orientation of the sensors. The battery lifetime of the controller unit is one hour. The system operates over an area of 57m^2, and calibrated accuracy over this area is said to be 5cm in position and 3° in orientation (one-sigma error level). A maximum of 120 measurements per sensor can be made each second. The Ascension *Motion Star Wireless* tracker, which uses pulsed DC signals, has very similar characteristics and performance.

Electromagnetic tracking systems are expensive—top of the range installations, such as those described above, can cost hundreds of thousands of dollars. Furthermore, due to the perceived safety hazards of intense electromagnetic fields, it seems unlikely that such trackers can be made to work over areas significantly larger than those demonstrated by existing systems.

3.3.4 Other tracking techniques

Inertial systems, which measure changes in the motion of an object being tracked, require no modifications to the environment in which the monitoring is taking place. Three orthogonal gyroscopes are used to determine angular rates of change of the object, and three orthogonal accelerometers measure the rate of change of its velocity. These values are integrated once and twice respectively to determine the object's orientation and position. Clearly, suitable initial orientation and position values must be determined in some other way to allow the definite integral to be calculated. Unfortunately, any errors in the rate measurements cause unbounded errors in their integrated values, resulting in large long-term drifts in tracker readings. A high-quality inertial navigation system on a commercial airliner, for example, might have a position drift of one nautical mile per hour [Feng94]. It therefore appears

that inertial trackers cannot, by themselves, be used to monitor the positions of objects in indoor environments.

An *Active Floor* has been described by Addlesee *et al.* [Addlesee97]. A matrix of load cells is laid underneath floor tiles, and by analysis of the distribution of weight across the floor the presence of moving objects can be inferred. The authors discuss the possibility of identifying people moving across the floor by Hidden Markov Model analysis of their footstep patterns. This sensor system has the attractive property that no tag or marker need be carried by people tracked by the system. However, it can provide only 2D position information, and inanimate objects cannot be identified.

3.3.5 Review

None of the types of system described above appear to be well-suited to the task of gathering information for location-aware computing. Although some systems can provide the required positioning accuracy, they cannot be extended to cover very large areas (electromagnetic schemes), suffer from severe line-of-sight constraints (outside-in optical systems), or require objects to be tagged with bulky apparatus (inside-out optical systems).

A further class of positioning systems, which are based on ultrasonic multilateration, appear to be more promising in many respects. These schemes are discussed below.

3.4 Ultrasonic Positioning Technologies

This section discusses the principles of operation of ultrasonic multilateration systems, examines existing work in this area, and identifies the problems that must be solved if an ultrasonic tracking system for location-aware computing is to be developed.

3.4.1 Fundamentals of ultrasonic multilateration

Ultrasonic multilateration systems determine the positions of objects by measuring distances between ultrasound sources and detectors. In some systems, a transmitter is mounted on the object, and several fixed receivers detect its signal. In others, a number of fixed transmitters generate signals that are picked up by a single receiver on the object to be tracked.

Ultrasonic distance measurement

Several ultrasonic distance measurement techniques have been proposed. All rely, in some sense, on measurements of the time taken for ultrasound to travel between sources and detectors. Knowledge of the speed of ultrasound in air can then be used to calculate the corresponding transmitter-receiver distance. The measurement schemes can be grouped into three categories [Lynnworth89]:

- *Broadband* methods, in which a pulse containing a range of ultrasonic frequencies is emitted by a transmitter, and receivers measure the time-of-flight of the pulse to determine the transmitter-receiver distance.

- *Narrowband*, or *phase-coherent* systems, in which received phase measurements of a continuous tone can be used to deduce the fractional part of the transmitter-receiver distance expressed as a multiple of the sound wavelength. The method cannot yield an unambiguous distance measurement, because the number of complete cycles made by the wave along the transmitter-receiver path is unknown. This phase ambiguity can be avoided by combining the narrowband measurement with a coarse, broadband measurement, at the expense of increased system complexity. Narrowband methods usually have lower signal-to-noise demands of the ultrasonic channel, because filtering can be used to select signals of only the measurement sound frequency.

- *Correlation*, or *matched filter* approaches, in which receivers identify a known pulse shape (or pulse sequence) emitted by a transmitter. These methods combine the low signal-to-noise requirements of narrowband systems with the phase-ambiguity-free characteristics of broadband systems. However, receivers that employ correlation approaches are complex.

Effects of the speed of sound

The distance measurement methods described above are based on a knowledge of the speed of ultrasound in air, which is around 340m/s at room temperature. The relatively slow speed of sound (when compared to, say,

that of light) has two main effects on the characteristics of ultrasonic multilateration systems. First, timing units with microsecond resolution are sufficient to measure distances with sub-centimetre accuracy, and so the electronic complexity of the systems is low. Second, the ultrasonic signals take an appreciable amount of time to travel from the transmitter to the receivers, resulting in a lag (typically tens of milliseconds) in position readings. Concerns about the relatively high lag of ultrasonic positioning systems have led several researchers to dismiss their suitability for VR/AR work, on the grounds that lag can cause motion sickness [Janin94], but most location-aware applications do not have such stringent timeliness requirements.

Ultrasonic propagation

Distance measurements made using ultrasound rely on the propagation of sound waves from sources to detectors. Although the wave propagation is predominantly line-of-sight, some diffraction of sound waves around objects does occur. This effect sometimes allows distance measurements from a transmitter to a receiver to be made when no line-of-sight exists between them, albeit with reduced accuracy. Furthermore, unlike, say, optical detectors, ultrasonic transmission and detection devices are simple, cheap structures [Asher97], and many such elements can be distributed around the environment to maximise line-of-sight coverage of sources by detectors. Since ultrasound does not propagate through walls, signals emitted by a transmitter in a room are contained within that room, allowing simultaneous, independent distance measurements using ultrasound in neighbouring but distinct spaces.

The range of ultrasonic frequencies that are suitable for use in positioning systems is quite narrow. The lower bound is around 20kHz, below which the sound may be audible to people with keen hearing, who may then be distracted by those emissions. The upper bound is dictated by the absorption of ultrasound in air, which increases sharply with frequency. The *Sound Pressure Level* (SPL) of a 170kHz ultrasonic signal is attenuated by up to 10dB/m in air, whereas the maximum attenuation of a 40kHz signal is around 1dB/m [Asher97]. These attenuations are, of course, in addition to the inverse-square law, which is common to all wave phenomena regardless of frequency and leads to a 6dB decrease in a signal's SPL as its distance from the source doubles. For these reasons, ultrasonic multilateration

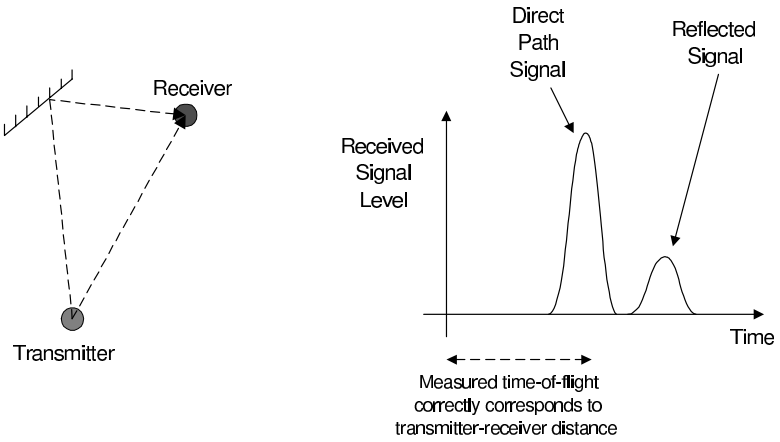
schemes often use frequencies in the range 40–75kHz, permitting accurate transmitter-receiver distance measurements at ranges of up to 10m [Feng94].

Robustness considerations

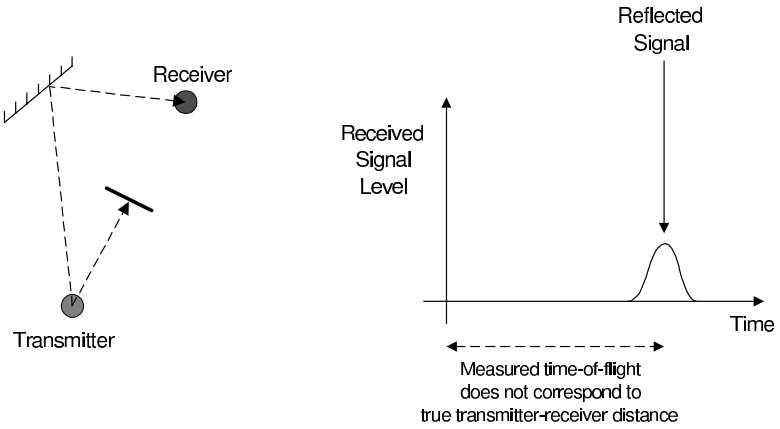
All ultrasonic multilateration systems suffer from problems caused by noise interference and reflections. Noise effects can be minimised by characterising the interference sources in the environment, selecting an operating frequency at which there is little noise, and filtering around that frequency. Reflection problems are particularly severe when the trackers are used in cluttered indoor environments. Ultrasonic signals may reflect off surfaces in such a way that they reach the receiver along routes longer than the direct transmitter-receiver distance. Examples of surfaces that may generate reflected signals with significant amplitudes include walls, hard furnishings, monitor screens, and so on. Distance measurements made using a reflected signal will be longer than those made using a direct path signal and, if used unwittingly, could introduce errors into the multilateration calculation.

When dealing with reflected signals, broadband and correlation-based systems (which use pulse waveforms) have an advantage over narrowband systems (which employ a continuous tone). Since reflected pulses travel a longer distance to a receiver than any direct pulse, they arrive at a later time, as shown in Figure 3.2(a), and the signals are distinct. Assuming that the direct pulse reaches the receiver, the first signal detected by the receiver after the pulse emission will have travelled along the direct transmitter-receiver path. It is therefore only this first signal that is used to measure the transmitter-receiver distance, and subsequent signals can be ignored. Reflections can therefore be separated from direct signals in broadband and correlation-based systems, but reflected narrowband signals cannot be separated in this way.

Occasionally, the direct path from a transmitter to a receiver is blocked, but a reflected pulse still reaches that receiver. The first pulse detected by the receiver will then have travelled along a path that is longer than the direct transmitter-receiver distance, as shown in Figure 3.2(b), and the calculated distance between the two will be incorrect. If this distance is then used by the multilateration algorithm, the reported object position will be erroneous, and so efforts must be made to reject distance measurements due to reflected signals.



(a) Direct signal reaches receiver



(b) Direct signal does not reach receiver

Figure 3.2: Distance measurements in the presence of reflected pulses

3.4.2 Existing ultrasonic tracking systems

An early ultrasonic tracker called the *Lincoln Wand* has been described by Roberts [Roberts66]. A pen-sized wand that contained a receiver was tracked around a 2.7m^3 volume, using four fixed transmitters placed around that volume. The absolute accuracy of the tracker was estimated to be around 5mm, and 25 position samples were made each second. Use of the system as a three-dimensional pointing device was envisaged.

The Logitech *6D Mouse* system [Logitech91] tracks the position and orientation of a pointing device, using a set of three transmitters attached to a fixed, triangular frame, and a corresponding set of three receivers attached to the pointing device. The manufacturers claim that the resolution of the device is 0.5mm in position and 0.5° in orientation, although accuracy figures are not given. The maximum tracking rate of the sensor is 25 position updates per second, and its operating volume is said to be around 9.7m^3. Feiner *et al.* describe an augmented reality system for maintenance assistance that uses a Logitech 6D Mouse (in conjunction with a tethered electromagnetic tracking system) to monitor the locations of a user and items of equipment around them [Feiner93].

Doussis has proposed an ultrasonic tracking system in which signals from a transmitter mounted on the object to be followed are detected by a set of static receivers [Doussis93]. A combined narrowband/broadband measurement scheme is used to determine transmitter-receiver distances, and the transmitter and receivers are triggered by a strobe signal sent along a wired network. The receiver hardware automatically decreases the gain applied to the incoming signal when the first ultrasonic pulse after the strobe is detected. This behaviour has the effect of reducing the system's sensitivity to reflections arriving after a direct-path pulse, but cannot reject a reflected signal that is the first pulse to arrive at the receiver. Figueroa and Mahajan have applied this 3D sensor to the problem of monitoring the position of a mobile robot [Figueroa94]. They describe a system deployment using five receivers that operates over a volume of 6m^3. The claimed system accuracy is around 2.5mm, and 100 position readings per second can be made.

The Intersense *IS-900CT* tracker [Intersense98] uses a combination of ultrasonic multilateration and inertial measurements to determine the position and orientation of a 30cm triangular frame with high accuracy and low lag. An infrared transmitter mounted on the frame triggers a set of address-

able ultrasonic beacons placed at known points in the environment, and the times-of-flight of the ultrasonic pulses from the beacons to three receivers mounted on the frame are determined. A signal processing unit uses these measurements and readings from an inertial sensor (also fixed to the frame) to determine the frame's position and orientation. Accuracy of the system is said to be 0.6cm in position and 0.25° in orientation (one-sigma error level). The developers foresee use of the system in AR environments, and in the television industry (for tracking cameras in virtual set applications).

3.4.3 Outstanding problems

The ultrasonic trackers described above provide sufficiently accurate location information for the purposes of context-aware computing. However, those systems follow only a single object, and are thus unsuitable for monitoring environments with dense populations of equipment and personnel. Many of the systems have a small operational volume, and only very limited solutions to the problem of ultrasonic reflections in indoor environments have been presented in the literature. The existing trackers are tethered, either for control or power purposes, and cannot meet the mobility demands of location-aware computing. Furthermore, due to the tethered nature of those systems, power management issues have yet to be investigated.

3.5 Summary

A wide range of location schemes have been described in the literature. These systems utilise several different techniques for determining the positions of objects, and they have found application in many areas. Nonetheless, it seems that no existing system has all the characteristics required of an information source for location-aware computing. Ultrasonic tracking technologies show some promise in this regard, but several outstanding technical difficulties remain to be overcome.

Chapter 4

Indoor Ultrasonic Tracking

4.1 Introduction

This chapter describes the principles of operation of a new tracking system which can determine both the positions and orientations of objects in indoor environments. The tracker, which is based on ultrasonic multilateration, has been designed specifically to support location-aware systems, and addresses several of the problems (highlighted in Section 3.4.3) that affect existing ultrasonic tracking schemes.

4.2 System Description

The components of the proposed indoor ultrasonic tracker, and interactions between them, are described in detail below.

4.2.1 System entities

The tracking system has a number of distinct entities:

- One or more *Mobile Positioning Devices*, or MPDs, which are attached to the objects to be tracked. MPDs are wireless, and have a unique ID, a radio receiver, and a transducer capable of emitting ultrasonic *positioning pulses*.

- A set of receivers, placed at a matrix of known points on the ceiling of the room in which objects are to be tracked. Receivers detect the

positioning pulses from the MPDs, and measure the times-of-flight of those pulses using on-board counters.

- A computing device called the *Matrix Manager* (MM), which gathers pulse times-of-flight from the receiver matrix, processes those measurements to obtain MPD-receiver distances, and hence determines MPD positions by multilateration.

- A device called the *Zone Manager*, or ZM, which periodically transmits an *addressing message* containing an ID across a radio interface. It is assumed that the ZM is located in the same building as the receiver matrix and MPDs.

4.2.2 Coordination of system components

The times-of-flight of positioning pulses are determined by measuring their arrival times at receivers relative to their times of emission from MPDs. Some degree of synchronisation between MPDs and receivers is therefore required. In the proposed system, MPD-receiver synchronisation is provided by the ZM's addressing messages, which are detected and decoded by all MPDs. If an MPD receives an addressing message containing its unique ID, it immediately transmits a ultrasonic positioning pulse[1]. Since the ZM's addressing messages travel at the speed of light, the time-of-flight of those messages will be small, and will certainly be insignificant compared to the ultrasonic transit time that is to be measured. Therefore, assuming that the time required for the MPD to decode the message and act upon it is fixed, the ZM can synchronise the MPDs and receivers to a sufficient level by resetting the receivers' counters as the addressing message is sent.

4.2.3 Processing of ultrasonic pulses

The proposed system uses broadband receivers to detect the ultrasonic positioning pulse, looking at the pulse envelope to determine its time of arrival. The signal arrival time may be associated with any unambiguous aspect of the envelope, for example, its peak, the mid-point between the times of the envelope's greatest positive and negative gradients, and so on. It is

[1]Ideally, MPDs would emit omnidirectional positioning pulses, so that an MPD's orientation did not affect the number of receivers that detected its signals.

assumed that the first signal reaching the receiver after pulse emission has travelled along the direct MPD-receiver path, and only this signal is used in the measurement of the MPD-receiver distance.

In order that the direct pulse is distinct from reflected pulses that reach a receiver only a short time later, a narrow positioning pulse is used. A short pulse width has the further advantage that it allows the signal arrival time to be accurately determined, because the steep pulse edges and narrow peak reduce uncertainty in identifying that time, whichever signal arrival criterion is used.

4.2.4 Multilateration considerations

By calculating MPD-receiver distances from the measured times-of-flight of the positioning pulse (using an estimate of the local speed of sound), the MM can determine the MPD's location by multilateration. The general multilateration principle considered in Section 3.2.3 requires that distances from at least four known non-coplanar points must be found to uniquely determine the 3D position of an unknown point. The receivers placed on the ceiling, however, are obviously coplanar. In this case, assuming that distances have been measured from the MPD to three or more non-collinear receivers, two solutions for the MPD's position may be found. The solutions lie on either side of the plane of the ceiling, one being a mirror image of the other in that plane. However, the physical constraint that the MPD must lie below the plane of the ceiling allows one of the solutions to be immediately discarded, and the MM can therefore uniquely determine the MPD's position using information from three or more non-collinear receivers.

4.2.5 Placement of receivers

If a direct positioning pulse is to be detected by a receiver, there must be a line-of-sight between the receiver and the MPD emitting that pulse. The placement of receivers on the ceiling of the room in which objects are to be tracked ensures that each receiver has lines-of-sight to many points within that space. Furthermore, the ceiling-mounted receivers are unlikely to interfere with normal use of that space, easing integration of the tracking system into the existing environment.

4.2.6 Emission of successive pulses

After a positioning pulse has been sent by the MPD, its reverberations will continue to arrive at the receiver from around the room for some time. Measurements of real ultrasonic positioning pulses indicate that direct-path and reflected signals have very similar pulse shapes. This result is not unexpected—most of the surfaces that generate significant reflections are relatively smooth compared to the wavelengths of ultrasound that are effective in positioning systems (see Section 3.4.1), and so the reflections are specular. Furthermore, efficient ultrasonic transducers are usually quite resonant [Asher97], implying that their bandwidths around the signal frequencies of interest are limited (perhaps only a few kHz around 40kHz). The range of frequencies that make up a positioning pulse will therefore be narrow, and it seems likely that the absorption coefficients of the reflecting surfaces are relatively constant for all the component frequencies of the pulse. For these reasons, it is difficult to directly distinguish between direct-path and reflected pulses.

This observation implies that the interval between the emission of two positioning pulses from MPDs in the tracker's operating space must not be less than the period for which reverberations from the first pulse continue. Were this not so, receivers would be unable to distinguish between the second pulse and reverberations of the first. This leads to a natural division of time into periods called *timeslots*, with a single MPD emitting a positioning pulse in each timeslot, as shown in Figure 4.1. This situation can be arranged by having the ZM transmit a single addressing message at the start of each timeslot. The length of a timeslot, which dictates the tracking rate of the system, is at least the maximum reverberation time of a positioning pulse, found to be around 15–20ms for typical office spaces. MPD IDs may be allocated to timeslots in any desired way.

The positioning pulse contains no information regarding the identity of the MPD from which it was emitted. As mentioned previously, efficient ultrasonic transducers have relatively low bandwidths, and any pulse encoding that included identity information would have to deal with the high probability of inter-symbol interference at the receiver, caused by the frequent reflections of ultrasound in the environment. Consequently, the length of the pulse might be very long, as would be the timeslots in which it was transmitted, thus limiting the maximum operating rate of the tracking sys-

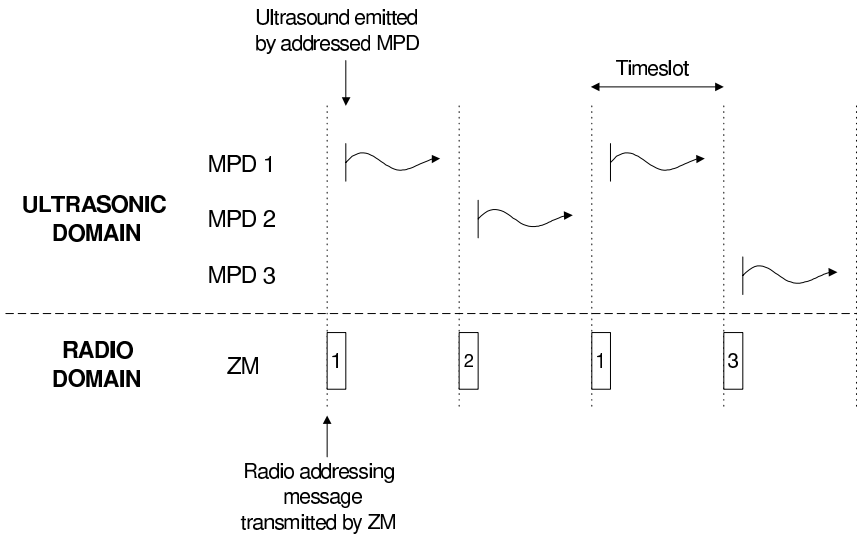


Figure 4.1: Operation of system using timeslots

tem. Instead, the system relies on coordination between the ZM and MM to generate complete tracking data consisting of MPD ID/position pairs—by combining the ZM's knowledge of which MPD was active in a timeslot with the MM's calculated position of the device active in that timeslot, the location of an MPD at a particular time can be found.

Although the tracking rate of the system is limited, the number of MPDs that may be tracked is dependent only on the number of distinct addresses that can be encoded into an addressing message. In principle, therefore, the system may monitor any number of MPDs by increasing the MPD address space appropriately.

4.2.7 Power management

To minimise an MPD's power requirements, the greatest possible proportion of its circuitry should be inactive for the greatest possible time. The regular nature of the addressing messages sent at the start of each timeslot allows a considerable power saving to be made by switching off the MPD's radio receiver and associated circuitry after each message has been decoded. A low-power timer (with a fixed, predetermined period) can then be used to activate those components just before the next addressing message is due, with the result that the high-power circuitry has a minimal duty cycle.

4.2.8 Concluding remarks

A similar tracking system that uses a set of static ultrasonic transmitters may be considered, with MPDs acting as receivers. One transmitter would emit a positioning pulse in each timeslot, and the MPDs would determine the times-of-flight of those pulses, reporting those measurements to the MM over a radio interface. Such a system would have the advantage that a single positioning pulse could provide a transmitter-MPD distance for many MPDs simultaneously. Moving MPDs, however, could not be accurately tracked, because the three or more distances required for multilateration would be determined at different times—the intervening movement of the MPDs would make those distances inconsistent. Furthermore, power management in MPDs would be more difficult, because they would not be able to anticipate the arrival times of the positioning pulses, forcing MPDs to keep their ultrasonic receiver circuitry active much of the time.

4.3 Determining Object Locations

To determine the location of an MPD, and hence the location of the object to which it is attached, the MM must first calculate MPD-receiver distances from the measured positioning pulse times-of-flight, and then use multilateration to find the 3D position consistent with those distances. This section details the steps taken in those calculations.

4.3.1 Measurement of MPD-receiver distances

Consider an MPD that has been triggered by an addressing message at the start of a timeslot, and a receiver that has detected the direct path pulse from the MPD. The interval t_p between the start of the timeslot and the first signal arrival time represents the sum of several individual periods:

- t_r, the radio signal transit time from the ZM to the addressed MPD.
- t_u, the ultrasound transit time from the MPD to the receiver.
- A number of fixed delays, $d_1, \ldots, d_n$, such as the time taken for the MPD to decode the radio message, and delays dependent on the signal arrival criterion used.

Then

$$t_p = t_r + t_u + \sum_{j=1}^{n} d_j \tag{4.1}$$

Also

$$t_r = \frac{l_r}{c} \tag{4.2}$$

$$t_u = \frac{l_u}{v_s} \tag{4.3}$$

where l_r is the distance from the ZM to the addressed device, c is the speed of light, l_u is the transmitter-receiver distance and v_s is the speed of sound in the room. The calculation assumes a uniform speed of sound at all points within the room.

Since the ZM and receiver matrix are roughly collocated, $l_r \sim l_u$. Also, $c \gg v_s$, so $t_u \gg t_r$, and Equation 4.1 can be rearranged as

$$l_u \approx v_s \times (t_p - \sum_{j=1}^{n} d_j) \tag{4.4}$$

By empirically determining the fixed delays $d_1, \ldots, d_n$ and making an estimate of v_s, Equation 4.4 can be used by the MM to calculate the MPD-receiver distance from the time at which the first signal arrived.

The speed of sound in air is affected by a number of environmental variables, such as the air temperature, relative humidity, air composition, and so on. Of these, changes in temperature have by far the greatest effect; raising the air temperature from 10°C to 30°C increases the speed of sound by over 3%, whilst a change in relative humidity (the next most significant variable) from 0% to 100% increases the speed of sound in air at 20°C by only 0.3% [Bohn88]. It is clear, therefore, that a good estimate of the speed of sound can be determined by measurement of the air temperature alone. Since Equation 4.4 already assumes a uniform air temperature within the room, a measurement of that temperature at a single point can be used by the MM to estimate v_s, via the following relationship [Bohn88]:

$$v_s = 331.45\sqrt{1 + \frac{T}{273.15}} \text{ m/s} \tag{4.5}$$

where T is the air temperature in degrees Celsius.

4.3.2 Multilateration using nonlinear regression

Consider a set of receivers placed at points on a horizontal ceiling. Suppose that a stationary world reference frame $\mathcal{W}$ is defined, with orthogonal coordinate axes **x**, **y** and **z**, such that all points with constant **z** coordinate are horizontal, and assign the **z**-coordinate 0 to the horizontal plane in which the receivers lie. Then, suppose that a MPD is at the coordinate (u, v, w), where $w < 0$ (indicating that the MPD is below the ceiling), and that its distance from a receiver at the coordinate $(x, y, 0)$ is l. It can be shown that

$$l = \sqrt{(x^2 + y^2) + (u^2 - 2xu) + (v^2 - 2yv) + w^2} \tag{4.6}$$

Equation 4.6 describes an idealised relationship in which all distances and positions are measured with perfect accuracy. Suppose, however, that this relationship is viewed in a setting where an estimate of the MPD's position is to be determined, based on a set of distances $l_1, \ldots, l_n$ simultaneously measured from it to a corresponding set of non-collinear receivers at positions $(x_1, y_1, 0), \ldots, (x_n, y_n, 0)$, where $n \geq 3$. The distance measurements and surveyed positions of the receivers will be subject to experimental error, and so Equation 4.6 must be extended, as below, to introduce an additional term ε that can account for these errors:

$$l_i = \sqrt{(x_i^2 + y_i^2) + (u^2 - 2x_i u) + (v^2 - 2y_i v) + w^2} + \varepsilon_i \quad (i = 1, \ldots, n) \tag{4.7}$$

This equation can be regarded as a *nonlinear model* [Myers90]. The MM can use the process of *nonlinear regression* to fit the collected values of l, x and y to the model, giving estimates $\hat{u}$, $\hat{v}$ and $\hat{w}^2$, of the parameters u, v and w^2. The estimates will minimise the sum of the squares of the *residuals* given by

$$e_i = l_i - \hat{l}_i \quad (i = 1, \ldots, n) \tag{4.8}$$

where

$$\hat{l}_i = \sqrt{(x_i^2 + y_i^2) + (\hat{u}^2 - 2x_i \hat{u}) + (\hat{v}^2 - 2y_i \hat{v}) + \hat{w}^2} \quad (i = 1, \ldots, n) \tag{4.9}$$

A estimate for the MPD location can then be determined as the coordinate $(\hat{u}, \hat{v}, -|\hat{w}|)$, taking the negative root of $\hat{w}^2$ to resolve the position ambiguity because the MPD is below the ceiling.

For $n \geq 4$, the standard deviation of the MPD-receiver distance measurements around the ideal relationship of Equation 4.6 can also be estimated, using the *standard error of the estimate*, s, where

$$s = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n-3}} \tag{4.10}$$

The standard error of the estimate provides an approximate measure of the overall predictive value of the nonlinear model, and hence indicates the extent to which the data set fits that model.

Underlying the nonlinear least squares approach are assumptions that all the error terms ε_i in Equation 4.7 are normal and independent with mean zero and common variance σ^2. Under these conditions, the least-squares parameter estimates are also *maximum likelihood* parameter estimates, i.e. those which best match the observed values of l, x and y. Furthermore, as the sample size n increases, the least-squares estimates asymptotically become *unbiased*, and become *minimum variance estimates* (i.e. the standard error of the estimate, s, becomes an unbiased estimate of σ).

4.3.3 Eliminating reflections

Section 3.4.1 highlighted the problems that can arise when receivers detect reflected pulses that are not preceded by a direct path pulse. These reflected signals could cause incorrect MPD-receiver distance measurements. Accurate determination of the MPD position by multilateration is only possible if the MPD-receiver distances used to estimate the parameters of the nonlinear model are correct, and so it is desirable to identify and reject the measurements due to reflected pulses.

As explained in Section 4.2.6, it is difficult to distinguish directly between the direct and reflected positioning pulses. The reflected path, however, is always longer than the direct path, and so distance measurements due to reflected pulses are always greater than the direct MPD-receiver distance. This observation allows data points in which the MPD-receiver distance is grossly exaggerated to be filtered out by comparing pairs of receiver measurements. Consider two receivers that have detected a pulse from an MPD, that are a distance l apart, and whose distances from the MPD have been measured as d_1 and d_2. Simple geometry shows that $l \geq |d_1 - d_2| - 2\rho$, where ρ is the maximum expected distance measurement error. It can also

be shown that if this inequality does not hold, then, under the constraint that reflections can only increase the path length, the larger of d_1 and d_2 must be an erroneous measurement, and can be discarded from the data set.

More subtly exaggerated distance measurements may be identified using statistical techniques. The circumstances in which a reflected pulse is the first to arrive at a receiver are rare, and hence the majority of distance measurements will be correct. Therefore, when the nonlinear model is fitted to the data set, the erroneous measurements, being *outliers*, are unlikely to agree well with the estimated parameters. The residuals calculated by Equation 4.8 reflect, to a certain extent, how well a particular data point fits the nonlinear model. However, a scaled residual called the *externally Studentized residual* is often more useful for detecting deviations from the model. A method of calculating externally Studentized residuals for regression diagnostics with nonlinear models is given in [Glantz90]. In the case of Equation 4.7, a data point with a large, positive externally Studentized residual is likely to be associated with an erroneous MPD-receiver distance measurement.

The following algorithm uses both the geometric and statistical tests to identify incorrect distance measurements in data gathered from receivers:

1. The geometric test is used to compare all pairs of data points, and those corresponding to distance measurements due to reflected pulses are eliminated.

2. If data points corresponding to four non-collinear receivers remain, the nonlinear model is fitted to the data set, and externally Studentized residuals are calculated for each data point. Otherwise, the algorithm terminates.

3. If the standard error of the estimate is less than a threshold τ_1, indicating that the data set fits the model well, the parameter estimates are used to determine the MPD location, and the algorithm terminates.

4. If the most positive externally Studentized residual is greater than τ_2, indicating that it deviates significantly from the fitted model, the data point associated with that residual is discarded, and step 2 is repeated. Otherwise, the algorithm terminates.

Suitable values of τ_1 and τ_2 may be determined empirically or by simulation. The algorithm may fail for a number of reasons, including non-convergence of the nonlinear regression, lack of a sufficient number of data points from non-collinear receivers, and failure of the data to fit the model well. Note that this algorithm requires at least four distance measurements to calculate the MPD's location, trading off the ability to find that location using three measurements for the ability to evaluate the extent to which the data fits the model using the standard error of the estimate.

Simulations of the above algorithm suggest that it is highly effective in eliminating distance measurements due to reflections[2]. When tested on trial data (generated for typical installations of receivers) in which 10% of readings were affected by reflections, the algorithm correctly identified and eliminated all erroneous distance measurements in over 98% of data sets. The algorithm rejected a correct distance measurement in less than 0.5% of data sets. Simulations also indicate that the statistical test can identify and eliminate all outliers detected by the geometric test. The geometric method, however, is very simple and fast, thus justifying its use as the first pass of a two-stage algorithm.

4.3.4 Effects of geometry on position accuracy

The relative geometry of an MPD and the receivers that detect its signal is important in assessing the quality of the resulting MPD location reported by the ultrasonic tracking system. Consider two receivers that detect a signal from an MPD, as shown in Figures 4.2(a) and 4.2(b). Distances determined by receivers have an associated uncertainty, and so each MPD-receiver distance measurement will lie within some range around the true distance[3]. Information from one receiver will therefore place the MPD somewhere within in a shell of points (whose thickness is equal to the range uncertainty) surrounding the receiver. The shaded areas of the two diagrams indicate those positions in the plane of the page that are consistent with the shells around both receivers, and represent the locus of possible estimates of the MPD's

[2]Section 5.5.3 assesses the effectiveness of the reflection elimination algorithms in a real system.

[3]More precisely, there will exist some probability distribution that describes the likelihood of a receiver reporting a range to an MPD with a particular accuracy. For clarity, however, this example does not illustrate the statistical interpretation of geometrical effects on position accuracy.

position in that plane. As can be seen, the relative geometry of the MPD and receivers affects the shape of the areas. In Figure 4.2(a), both the horizontal and vertical position of the MPD are well defined. In Figure 4.2(b), however, the horizontal position is more uncertain than the vertical position—a small ranging error can be magnified into a large position error.

A quantity called the *Position Dilution Of Precision*, or PDOP, is used to relate the radial range measurement accuracy to position measurement accuracy. PDOP depends solely on the relative geometry of the MPD and receivers, and thus provides a means for determining whether that geometry will result in a well-defined or more uncertain position measurement. A low value of PDOP (~ 1) indicates a good relative geometry, with higher values indicating worse geometries. Assuming that the individual range measurements have a one-sigma error of unity where the expected mean is zero and the correlation of errors between receivers is zero, PDOP is defined by

$$\text{PDOP} = \sqrt{\text{Trace}\left(\left(\mathbf{A}^{\text{T}}\mathbf{A}\right)^{-1}\right)} \tag{4.11}$$

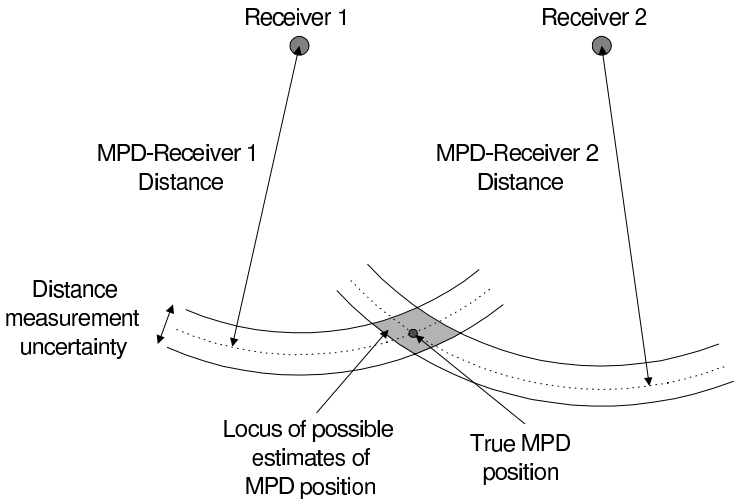
where

$$\mathbf{A} = \begin{pmatrix} \frac{\hat{u}-x_1}{\hat{l}_1} & \frac{\hat{v}-y_1}{\hat{l}_1} & \frac{\hat{w}}{\hat{l}_1} \\ \vdots & \vdots & \vdots \\ \frac{\hat{u}-x_n}{\hat{l}_n} & \frac{\hat{v}-y_n}{\hat{l}_n} & \frac{\hat{w}}{\hat{l}_n} \end{pmatrix} \tag{4.12}$$

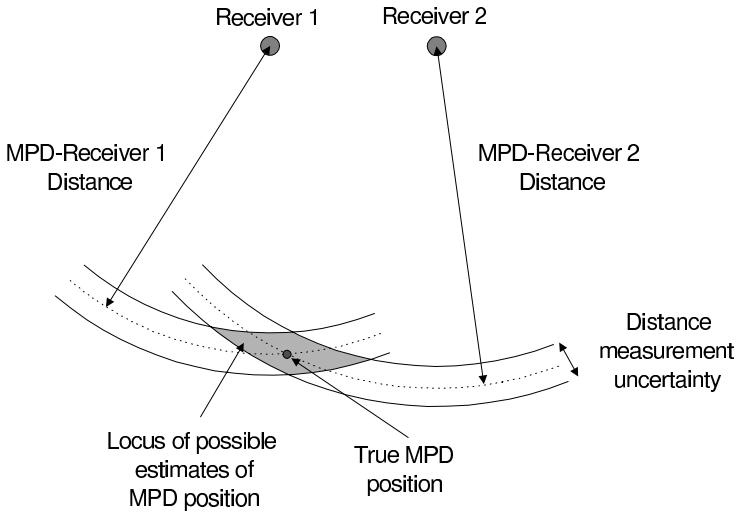
The coefficients of the matrix $\mathbf{A}$ are the direction cosines of the lines-of-sight from the MPD to the receivers as projected along the $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ axes. Related quantities called the *Horizontal and Vertical Dilutions of Precision*, HDOP and VDOP, may also be defined, indicating the effect of the relative system geometry on the horizontal and vertical components of the position measurement respectively. The interested reader may wish to consult Appendix A for the derivation of the PDOP formula and related expressions.

4.3.5 An alternative multilateration approach

Other researchers [Doussis93][Figueroa94] have proposed ultrasonic multilateration systems in which the speed of sound is not calculated from a temperature measurement, but is estimated as an additional parameter in



(a) Good geometry



(b) Poor geometry

Figure 4.2: Effects of varying MPD-receiver geometry

the regression procedure. In such systems, a model is fitted to a corpus of data comprising of signal transit times and receiver positions, rather than distance measurements and receiver positions. A suitable nonlinear model, based on Equations 4.4 and 4.7, might be

$$t_{pi} = \frac{1}{v_s}\sqrt{(x_i^2 + y_i^2) + (u^2 - 2x_i u) + (v^2 - 2y_i v) + w^2} + \sum_{j=1}^{m} d_{ji} + \varepsilon_i \quad (i = 1, \ldots, n) \tag{4.13}$$

The justification given for this approach is that the speed of sound often varies throughout a room, being affected by air currents and local temperature fluctuations, and the above model provides a better estimate of the average speed of sound within the working volume of the system than does a temperature measurement at a single point. However, this model is more unstable than that of Equation 4.7, and requires data from an additional receiver—at least four measurements are required to determine the MPD location, and at least five are needed to calculate the standard error of the estimate. This additional requirement can be restrictive, because receivers will only be able to provide this extra information if the ultrasonic signal reaches them, whereas the direct temperature measurement will always be available.

4.4 Determining Object Orientation

The discussion of context-aware computing systems in Chapter 2 identified a requirement for information about both the positions and orientations of objects in the environment. This section describes two methods by which an ultrasonic tracker can determine object orientations, and examines the situations to which each method is best suited.

4.4.1 Rigid-body approach

Suppose that a number of MPDs are placed on a rigid object, at non-collinear points $(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)$, where $n \geq 3$ and the points are measured in a reference frame, $\mathcal{A}$, which is fixed relative to that object. The corresponding positions $(x'_1, y'_1, z'_1), \ldots, (x'_n, y'_n, z'_n)$ of the points in some other

reference frame $\mathcal{B}$ can be found using the rigid-body transformation

$$\begin{pmatrix} x'_i \\ y'_i \\ z'_i \end{pmatrix} = \mathbf{R} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \mathbf{v} \quad (i = 1, \ldots, n) \tag{4.14}$$

where $\mathbf{R}$ is a rotation matrix describing the attitude of $\mathcal{A}$ relative to $\mathcal{B}$, and $\mathbf{v}$ is the position vector of the origin of $\mathcal{A}$ measured in $\mathcal{B}$.

Suppose also that the MPDs' positions in the reference frame $\mathcal{W}$ used by the tracking system are found to be $(\hat{x}_1, \hat{y}_1, \hat{z}_1), \ldots, (\hat{x}_n, \hat{y}_n, \hat{z}_n)$. If the position measurements were exact, the relationship given by Equation 4.14 could be used to find the translational ($\mathbf{v}$) and rotational ($\mathbf{R}$) mappings between $\mathcal{A}$ and $\mathcal{W}$—these mappings correspond to the position and orientation of the object in $\mathcal{W}$ respectively. Measurement errors will ensure that the precise relationship does not hold, but the mapping parameters may be estimated in a least-squares sense by finding $\hat{\mathbf{R}}$ and $\hat{\mathbf{v}}$ to minimise

$$\sum_{i=1}^{n} \left\| \begin{pmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{z}_i \end{pmatrix} - (\hat{\mathbf{R}} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \hat{\mathbf{v}}) \right\|^2$$

Challis gives a robust method for determining $\hat{\mathbf{R}}$ and $\hat{\mathbf{v}}$ given a set of corresponding points in $\mathcal{A}$ and $\mathcal{W}$ [Challis95].

In some cases, one or more of the degrees of freedom of motion of an object may be constrained, either physically or by convention. Examples might be a door, whose motion is restricted to rotation around one axis, and a chair, which is typically found upright. In such circumstances (assuming adherence to any constraints due to convention), it is possible to fully determine an object's position and orientation using data from fewer than three MPDs, because the constraints reduce the number of unknown parameters that must be estimated. For example, the locations of two points on the chair back would suffice to describe its position and rotational configuration, whereas the location of a single point (which was not collinear with the hinges) on the door could fully describe its position and orientation. No algorithms for determining the position and orientation of an object whose motion is constrained are given in this dissertation, however, because those calculations are dependent on the particular constraints of each situation.

When applied using information from multiple MPDs, this technique is best suited to stationary or slow-moving objects, because those MPDs will

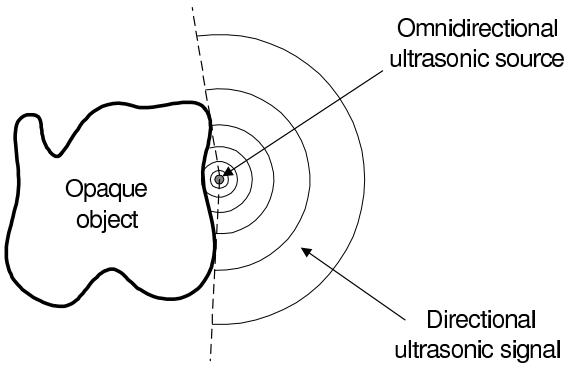


Figure 4.3: A directional ultrasonic signal formed by an opaque object

be located at different times—any intervening movement of the object will introduce errors into the calculated orientation. For the same reason, it is advantageous to locate the MPDs in consecutive timeslots, so as to minimise the period of time over which movement could take place.

4.4.2 Directional beam approach

Many objects are opaque to ultrasound, and an alternative principle may be used to obtain information about their orientation. Consider an omnidirectional MPD attached to such an object, as in Figure 4.3. The opacity of the object to ultrasonic waves ensures that sound energy can only leave its vicinity in the direction shown (this directional effect may be amplified if the sound transmitted by the MPD is itself directional to some degree). Knowledge of the object's properties, together with information about the MPD's position and orientation on the object, will therefore allow calculation of the direction in which the energy of the positioning pulse will travel, relative to the object.

Suppose a positioning pulse is transmitted by the MPD, and its position is determined by the tracking system to be (x, y, z). An estimate of the direction of travel of the positioning pulse can be found by comparing the MPD's position with the positions $(u_1, v_1, w_1) \ldots (u_n, v_n, w_n)$ of the n receivers that detected the ultrasonic signal. A further comparison between this information and the known direction in which the pulse travels relative to the object will allow an estimate of the object's orientation to be found.

To illustrate this computation, the orientation around a vertical axis of

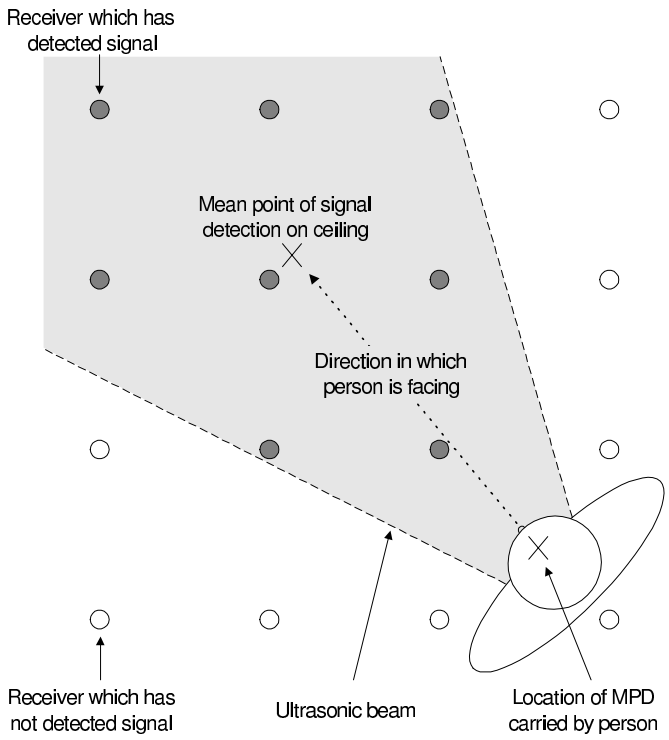


Figure 4.4: Determining the direction in which a person is facing

a person wearing an MPD (as in Figure 4.4) is determined. In this case, the sound energy is directed in front of the person, with the middle of the beam lying in the same vertical plane as the direction in which they are facing.

First, the mean point of detection of the ultrasonic signal on the ceiling, $(\hat{u}, \hat{v}, \hat{w})$, is computed, where

$$\begin{pmatrix} \hat{u} \\ \hat{v} \\ \hat{w} \end{pmatrix} = \frac{1}{n} \sum_{i=1}^{n} \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix} \tag{4.15}$$

Next, the vector (a, b, c) from the transmitter to this point is found, where

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \hat{u} - x \\ \hat{v} - y \\ \hat{w} - z \end{pmatrix} \tag{4.16}$$

The vector $(a, b, 0)$ is a good estimate of the horizontal direction of the middle of the sound beam leaving the object's vicinity, and can be used to

calculate the direction θ in which the person is orientated around a vertical axis (relative to some other horizontal direction specified by a vector $(m, n, 0)$). It can be shown that

$$sin(\theta) = \frac{\left| \begin{pmatrix} m \\ n \\ 0 \end{pmatrix} \times \begin{pmatrix} a \\ b \\ 0 \end{pmatrix} \right|}{\left| \begin{pmatrix} m \\ n \\ 0 \end{pmatrix} \right| \left| \begin{pmatrix} a \\ b \\ 0 \end{pmatrix} \right|} \tag{4.17}$$

and

$$cos(\theta) = \frac{\begin{pmatrix} m \\ n \\ 0 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ 0 \end{pmatrix}}{\left| \begin{pmatrix} m \\ n \\ 0 \end{pmatrix} \right| \left| \begin{pmatrix} a \\ b \\ 0 \end{pmatrix} \right|} \tag{4.18}$$

From $sin(\theta)$ and $cos(\theta)$, the value of θ in the range $(-\pi, \pi]$ can be uniquely determined.

This method of finding an object's orientation has three main drawbacks:

- If the directional beam is symmetrical around an axis, then no information about orientation around that axis can be deduced.
- The estimate of the pulse direction derived from the receiver positions is quite crude, limiting the accuracy of the orientation measurement.
- The calculation performed to determine orientation is dependent on the properties of the object concerned, so no algorithm provides a general implementation of this principle.

Only a single MPD need be attached to the object, however, and so this technique is still useful in situations where high accuracy and/or complete orientation information are not required, and where tagging of an object with multiple MPDs would be cumbersome.

4.5 Summary

This chapter has described a new indoor ultrasonic tracking system, which locates objects by finding the positions of Mobile Positioning Devices placed on them. The tracker is wireless, suitable for low-power operation, and its fixed components are unobtrusive. Furthermore, an unlimited number of objects may be monitored by the system.

Algorithms that can determine MPD-receiver distances from the times-of-flight of positioning pulses, and that can use those distances to find 3D MPD positions by multilateration have been presented. Methods for identifying reflections of the positioning pulse, which occur frequently in the indoor environment, have been described, as have means for assessing the quality of the positions found by multilateration.

Two techniques for determining the orientations of objects using information gathered by the system have been presented, and the situations to which each technique is best suited have also been discussed.

Chapter 5

A Prototype Tracker

5.1 Introduction

This chapter describes an indoor tracker, built according to the principles set out in Chapter 4, and evaluates its effectiveness, with particular regard to its suitability for creating context-aware computer systems.

5.1.1 Overview

The prototype tracker, an overview of which is shown in Figure 5.1, is intended for use in an office. MPDs are attached to equipment in that environment, and are carried by personnel. As described in Chapter 4, the ZM addresses MPDs and causes them to emit positioning pulses, which are detected by a set of receivers placed on the ceiling of the office. The MM uses the pulse times-of-flight measured by receivers to determine the positions of MPDs. The prototype system uses 40ms timeslots, and makes use of two previously undescribed entities, a *Global Clock Generator* (GCG) and an *Area Manager* (AM), which serve to coordinate other system components.

5.2 Hardware Implementation

This section describes the hardware used by the prototype tracking system.

5.2.1 Global Clock Generator

The GCG sends a timing signal to the ZM and MM along a 420Mb/s LVDS serial daisy-chain network. The timing signal indicates the start of each

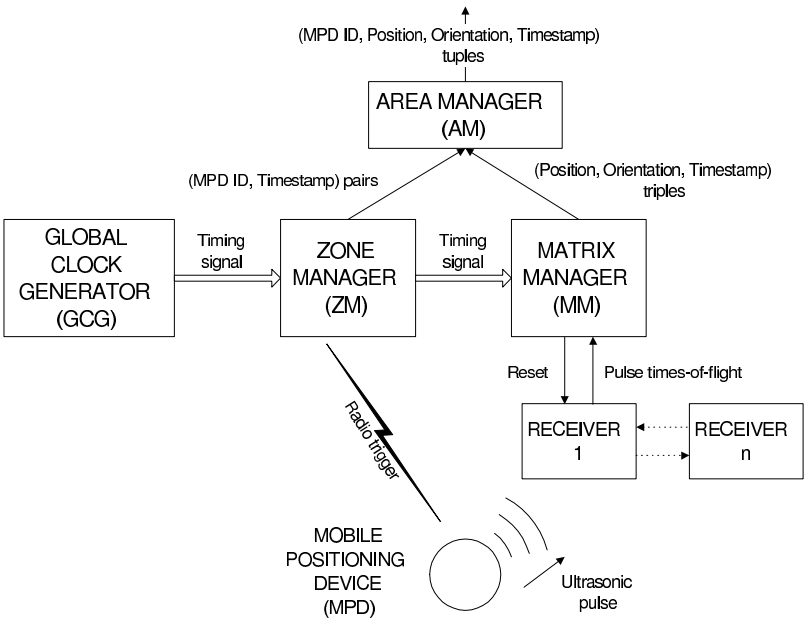


Figure 5.1: Overview of tracking system

40ms timeslot, and contains a 35-bit identifier, based on the real time, which is incremented after every timeslot. The timing signal suffers a maximum delay of 50ns along each hop of the timing chain, implying synchronisation of the ZM and MM (which are separated by a single hop) to that level.

5.2.2 Zone Manager

The ZM consists of a 200MHz Pentium PC, an interface board, and a 418MHz FM radio transceiver, as shown in Figure 5.2. The ZM is part of the daisy-chain that distributes timing signals from the GCG, and so the interface board has two LVDS network interfaces (one input, one output) for propagation of that signal.

Addressing messages are generated by the PC. Each 42-bit message includes a 16-bit field (the Active Device field) that holds an MPD ID, and an 8-bit Cyclic Redundancy Check (CRC), which permits error-checking of the message. After each message is generated, it is loaded into a store on the interface board by the PC. At the start of the next timeslot, a 135-bit preamble[1] is sent to the radio transceiver by the interface board, followed by

[1]The preamble length was empirically determined to be the minimum necessary to ensure reliable communications between the ZM and MPDs.

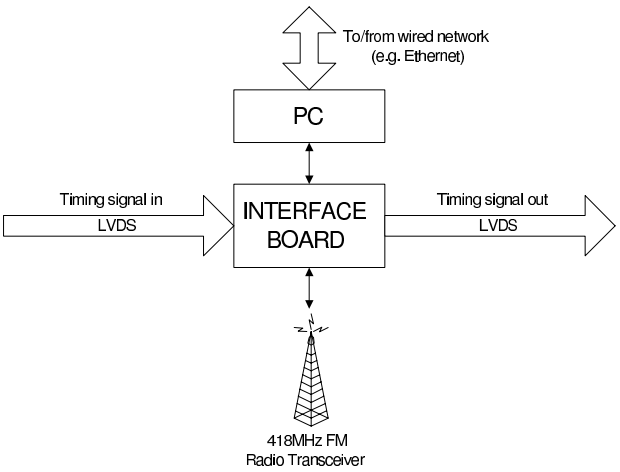


Figure 5.2: Overview of Zone Manager

a Manchester-encoded version of the stored addressing message (the encoding ensures that the message is DC-balanced). The preamble and message are transmitted at 40kb/s. The PC can interrogate the interface board to determine the current timeslot ID, which is stored at the start of the timeslot.

5.2.3 Mobile Positioning Device

MPDs are based around a Xilinx 3090L FPGA, which is connected to a 418MHz FM radio transceiver and a set of six 40kHz piezoceramic ultrasonic transducers arranged in a hemisphere. Power is supplied by two Lithium Thionyl Chloride cells connected in series, and a normal-mode helical antenna feeds the RF input to the radio transceiver. Two clocks drive segments of the FPGA circuitry; a slow 32kHz clock, which runs continuously, and a fast 8MHz clock, which can be controlled by the FPGA. MPDs measure 6cm $\times$ 4.5cm $\times$ 2cm, and weigh 67g. The MPD and both sides of its circuit board are shown in Figure 5.3.

A state diagram for the FPGA is shown in Figure 5.4. MPDs listen to the addressing messages sent by the ZM, and, if the Active Device field matches an MPD's 16-bit ID (stored as part of the FPGA configuration) and the associated CRC is valid, the addressed MPD drives its ultrasonic transducers with the signal shown in Figure 5.5(a). Figure 5.5(b) shows the form of the resulting pulse, as detected by a 40kHz ultrasonic transducer

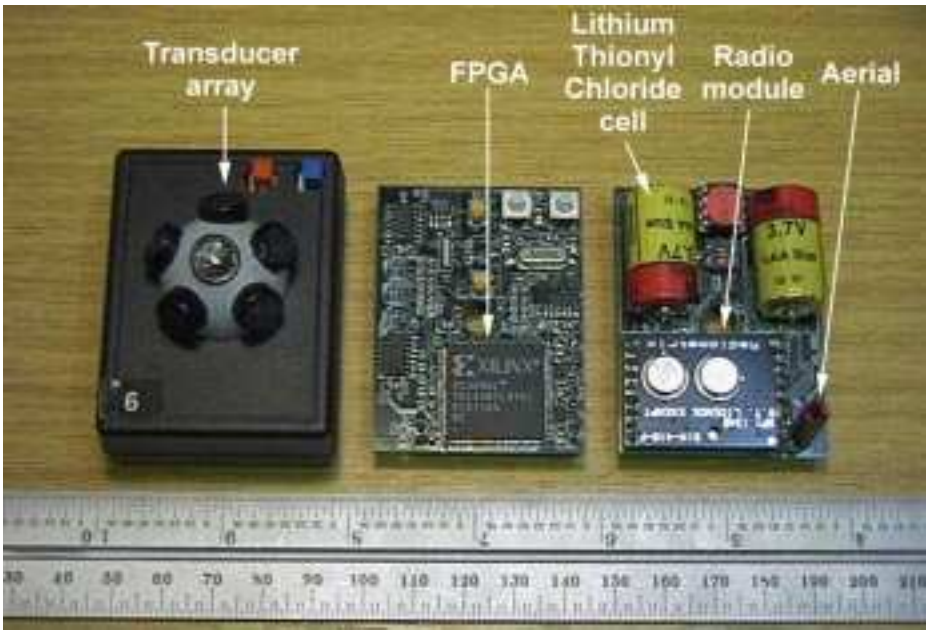


Figure 5.3: External and internal views of an MPD

placed 1m from the transmitting MPD and amplified by a factor of 470. The frequency spectrum of the pulses, generated from the averaged FFTs of fifty 5ms sample windows, is shown in Figure 5.5(c). Most of the signal energy lies in the range 35–47kHz—other peaks in the spectrum are due to the second harmonic of the fundamental (seen at around 80kHz) and a variety of noise sources in the environment, which are discussed below in Section 5.5.5.

Figure 5.4 illustrates some of the power-saving mechanisms implemented by the MPD. After the MPD has decoded an addressing message, it turns off its radio receiver and fast clock for 34ms, activating them 500μs before the next message is due (the continuous slow clock is used to time the 40ms intervals between messages). During the period between addressing messages when the fast clock and receiver are not active, the MPD draws little current from its batteries, and is said to have entered a *slumber* state.

If the MPD fails to receive an expected message, or receives a message corrupted by noise, it enters a *searching* state. The MPD continues to listen for messages for a short time (120ms) before switching off its radio receiver and fast clock for 11.88s, after which time the radio channel is sampled for 120ms, and so on, until a valid addressing message is detected. The

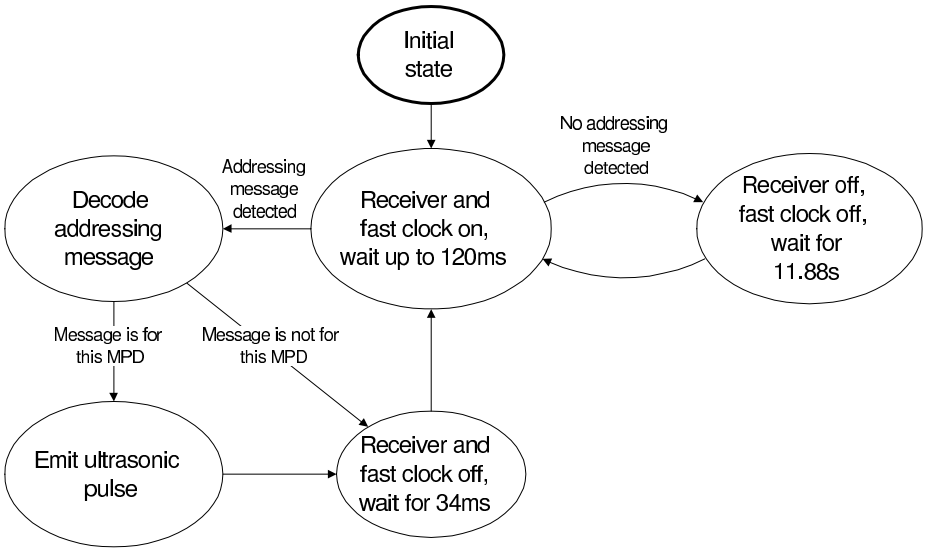


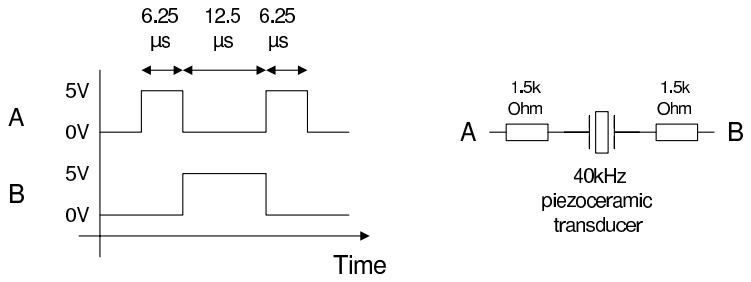
Figure 5.4: MPD operating state diagram

radio receiver and fast clock therefore have a duty cycle of only 1% in the searching state, and this behaviour ensures that power is not wasted if the ZM is switched off, or if the MPD is taken outside its range.

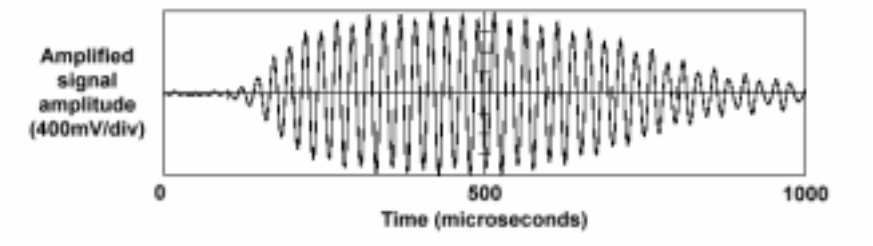
5.2.4 Receiver

Ceiling-mounted receivers detect positioning pulses emitted by MPDs. The signal is detected by a 40kHz piezoceramic ultrasonic transducer, amplified, full-wave rectified and smoothed, and the resulting envelope is digitised by an ADC. A comparator and counter (based around a Xilinx 4005E FPGA) sample the digitised signal for 20ms after the start of each timeslot. If an incoming signal rises above a threshold, set relative to the ambient noise level seen by the comparator, the time (relative to the start of the timeslot) at which that signal first peaks is stored.

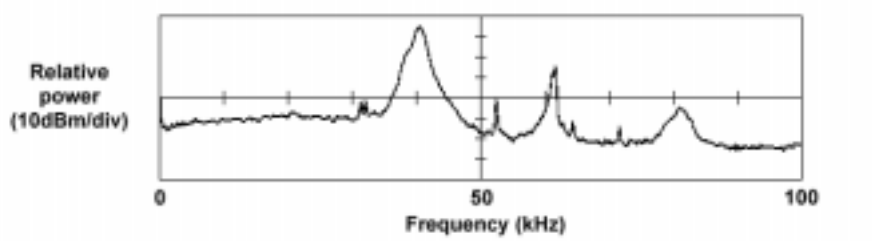
Receivers are connected by a dual 420Mb/s LVDS serial daisy-chain network, with the two daisy-chains proceeding in opposite directions. Each receiver has a unique 8-bit address in the daisy-chain, set using on-board switches. Incoming messages may indicate the start of timeslot (thus resetting the counters of the peak signal detector), set the signal detection threshold for an addressed receiver, or cause an addressed receiver to transmit its stored peak signal time (if any) on the outgoing daisy chain.



(a) Driving pulse



(b) Typical received signal



(c) Power spectrum of received signals

Figure 5.5: Ultrasonic pulse emitted by MPDs

(a) Top view (b) Bottom view

Figure 5.6: An installed receiver

Receivers are placed in a square grid on the ceiling, 1.2m apart. A relatively high receiver density was chosen to increase the likelihood that at least four receivers (the number required to calculate MPD positions, see Section 4.3.3) would detect the signals from each MPD, even when an MPD was surrounded on several sides by occluding objects. In total, 38 receivers have been installed to cover an area of around $56m^2$. The volume covered by the installed receivers is around $158m^3$. To minimise their visual impact on the office environment being instrumented, the receivers are mounted above the tiles of a suspended ceiling, as shown in Figure 5.6. The ultrasonic transducer is placed through a hole drilled in the ceiling tile, and is connected to the receiver body using a shielded cable.

5.2.5 Matrix Manager

The MM consists of a 200MHz Pentium PC, interface board and temperature sensor, as shown in Figure 5.7. Like the ZM, the MM is part of the clock distribution daisy-chain, and so the interface board has two LVDS network interfaces used for propagation of the clock signal. It has two further LVDS

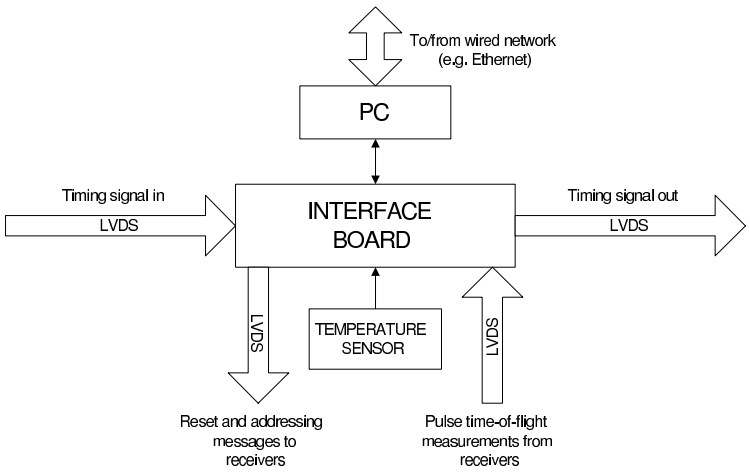


Figure 5.7: Overview of Matrix Manager

interfaces (one input, one output) that allow it to communicate with the receiver matrix.

At the start of each timeslot, the interface board resets all receivers. The reset signal suffers a 50ns hop-to-hop delay along the outgoing daisy-chain, but the 12.8μs end-to-end delay across a receiver chain of maximum length (256 receivers) could result in distance measurement errors of at most 0.5cm, and was thus considered to be insignificant in the context of the prototype system. After the receivers' sampling windows have closed, the PC can poll individual receivers through the interface board, retrieving any peak signal time that they may have measured. The MM also measures the air temperature at a single point (preferably located towards the centre of the room and away from heat-generating or air-conditioning equipment) using a National Semiconductor LM35 temperature sensor. The output of this sensor is digitised using an ADC and stored in the MM at the start of each timeslot, as is the current timeslot ID. The PC can interrogate the interface board to determine the stored values. The data sheet for the LM35 temperature sensor claims accuracy to within 0.5°C at 25°C.

5.3 Software Implementation

This section describes the software used by the prototype tracking system for control and calculation purposes.

5.3.1 Timeslot allocation

The PC component of the ZM runs a process that allocates MPD IDs to timeslots. Timeslots are allocated more frequently to the MPDs associated with objects that move often (e.g. personnel), but the priority between tracked objects is fixed. At the end of each timeslot, the timeslot ID and the allocated MPD ID are sent to the AM, via a TCP socket interface.

5.3.2 Multilateration calculation

A process running on the PC component of the MM uses the times-of-flight of the positioning pulse, together with the measured air temperature and known receiver positions, to determine the position of the active MPD in each timeslot, using the algorithms outlined in Section 4.3.

The finite size of the MPD's hemispherical transducer block must be taken into account by the multilateration algorithm. The software considers the centre of the transducer block to represent the true position of an MPD. MPD-receiver distances, however, are measured between the receiver and the face of the transmitting transducer, which is $\approx$ 1cm away from the centre of the transducer block. The directionality of the ultrasonic transducers is such that signals emitted by a transmitter are most likely to be detected by a receiver less than 60° from the axis of the transmitted signal. Simple geometry then shows that a 1cm correction added to the MPD-receiver distances calculated using the procedures of Section 4.3.1 will bring those measurements closer to the true distances from the centre of the transducer block to the receivers.

A public-domain nonlinear regression package [Gallant90] was used to implement the core of the multilateration algorithms. The chosen values of the constants used for reflection elimination (Section 4.3.3) are shown in Table 5.1. The latency between emission of ultrasound by an MPD and completion of the multilateration calculation by the MM was found to vary between 25 and 30 milliseconds, depending on the number of receivers (typically between 4 and 12) that detected a signal from the MPD.

The PC also calculates an estimate of the orientation of the MPD, using the algorithm of Section 4.4.2 and the knowledge that the positioning pulse emanates from the MPD in a roughly hemispherical pattern. At the end of each timeslot, the PC sends the timeslot ID and the calculated MPD

Constant	*Value*
Maximum expected distance measurement error, ρ	0.05m
Model fit threshold, τ_1	0.05m
Rejection threshold, τ_2	0.6

Table 5.1: Values of constants used for reflection elimination

position and orientation to the AM, via a TCP socket interface.

5.3.3 Determination of MPD positions

The AM receives information streams from the ZM and MM. The ZM stream indicates which MPD was addressed in each timeslot, and the MM stream indicates the position and orientation of the active MPD in each timeslot. Using the timeslot ID as a key, the AM correlates the streams to determine MPD positions, as below.

$$\overbrace{(\text{timeslot ID}, \text{MPD ID})}^{\text{From ZM}} + \overbrace{(\text{timeslot ID}, \text{position}, \text{orientation})}^{\text{From MM}}$$
$$\Downarrow$$
$$(\text{timeslot ID}, \text{MPD ID}, \text{position}, \text{orientation})$$

The known relationship between the timeslot ID and the real time allows the MPD's position and orientation at a particular instant to be deduced. This information is reported to client software entities across a TCP socket interface.

5.4 System Calibration

Before the tracking system can be used, it must be calibrated. This section describes the measurements taken to calibrate receivers and determine their fixed positions.

5.4.1 Delay calibration

Calibration measurements were made to determine the sum of the unknown delays, $d_1, \ldots, d_n$, in Equation 4.4 for an MPD and several receivers. Equa-

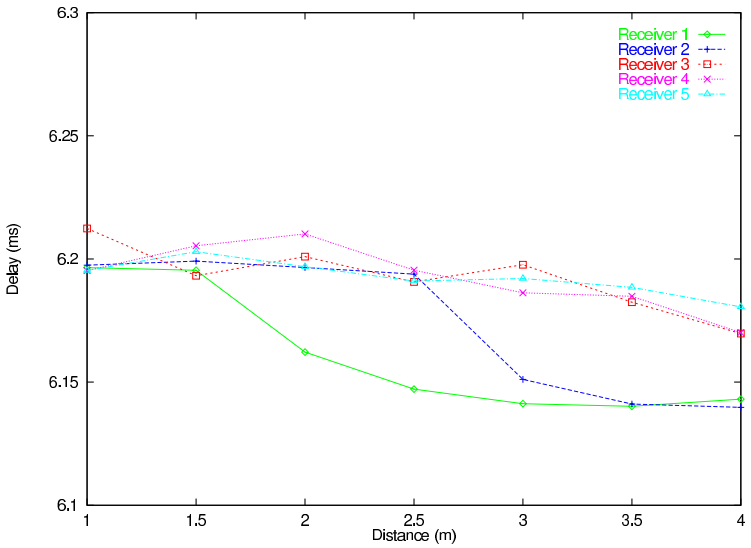


Figure 5.8: Delay calibration results for five receivers

tion 4.4 may be rearranged as

$$\sum_{j=1}^{n} d_j \approx t_p - \frac{l_u}{v_s} \tag{5.1}$$

in order to determine the sum of the delays in terms of the estimated speed of sound, v_s, a measured positioning pulse time-of-flight, t_p, and the measured MPD-receiver distance, l_u. Figure 5.8 shows values of $\sum d$ (averaged over 200 readings) for five receivers placed at varying distances from the transmitter. The average sum of delays across all readings was 6.18ms, with all measurements lying within 0.04ms of this value. A slight downward trend in $\sum d$ with increasing MPD-receiver distance is seen—this indicates that the relationship between the time-of-flight of the positioning pulse and the MPD-receiver distance is not fully modelled by Equation 4.4.

Further measurements were taken to characterise the variation in the time taken for different MPDs to decode radio messages and generate ultrasonic pulses in response. It was found that the measured values of $\sum d$ (averaged over 200 readings) for five MPDs placed at a distance of 2.500m from a receiver lay in the range 6.21 ± 0.02ms.

The delay variations identified between individual receivers and MPDs could result in distance measurement errors of at most 1.5cm, and were thus considered to be relatively insignificant in the context of the proto-

Receiver pair	*Measured separation (m)*	*Separation calculated from surveyed positions (m)*	*Error (%)*
1	6.902	6.890	-0.2
2	6.728	6.728	0.0
3	6.000	6.004	+0.1
4	5.962	5.957	-0.1

Table 5.2: Accuracy of surveyed receiver positions

type system. The value of $\sum d = 6.18$ms was therefore adopted for use in measurement of the distance from any MPD to any receiver.

5.4.2 Surveying receiver positions

The positions of the ceiling receivers were determined by construction of a *trilateration network* [Davis81] from two base points. Distances between receivers were found directly using a tape measure. The accuracy of the surveyed positions was examined by taking distance measurements between distant receiver positions that were not directly linked by the trilateration network, and comparing them with the corresponding distances calculated from the surveyed receiver positions. The results of these comparisons are shown in Table 5.2.

5.4.3 Ceiling self-calibration

Equation 4.7 allows estimates of an MPD's position to be calculated from a set of distance measurements to receivers placed at known points in the environment. However, it is possible to use the same nonlinear model to estimate the unknown positions of receivers, based on measurements of their distances from MPDs placed at known points.

This observation leads to an alternative method of ceiling calibration. An MPD is placed consecutively at n points $(u_1, v_1, w_1), \ldots, (u_n, v_n, w_n)$, where $n \geq 3$, and the corresponding distances $l_1, \ldots, l_n$ of the MPD from a receiver at an unknown position $(x, y, 0)$ are found by the method of Section 4.3.1. Nonlinear regression is then used to determine estimates $\hat{x}$ and $\hat{y}$ of the parameters x and y, leading to a least-squares estimate of the receiver position, $(\hat{x}, \hat{y}, 0)$.

Since the same set of MPD positions can be used to determine a number of receiver positions, assuming that all MPD signals can be detected by all receivers, the number of points whose positions must be accurately determined during ceiling calibration (say, by construction of a trilateration network) may potentially be reduced by the use of this technique. However, reflections must again be eliminated by methods similar to those outlined in Section 4.3.3, and geometrical limitations on the accuracy of the position estimates should also be taken into account, as described in Section 4.3.4. Furthermore, unless the distance measurements made using times-of-flight of positioning pulses have comparable accuracy to those made directly using a tape measure, the accuracy of the receiver positions determined by self-calibration is unlikely to match that shown in Table 5.2. For these reasons, the self-calibration technique was not applied to the installed receiver matrix.

5.5 Tracker Evaluation

This section investigates a number of aspects of the performance of the indoor tracking system, and validates assumptions used in developing the multilateration algorithms.

5.5.1 Assessment of distance measurements

To characterise the distance measurements made by receivers, a modified MPD with only a single transmitter was placed at varying distances from a receiver, and rotated into a number of orientations. Figures 5.9 and 5.10 show the accuracy and standard deviation of the receiver's distance measurements against the transmitter-receiver distance and the angle between the central axes of the transmitter and receiver. The effects of unmodelled relationships between the transmitter-receiver distance and angle and the positioning pulse time-of-flight can be seen in Figure 5.9 (see also Section 5.4.1). These trends are probably due to non-ideal behaviour of the receiver smoothing and peak signal detection circuits, which introduce a bias dependent on the signal strength into pulse time-of-flight measurements. As the transmitter-receiver distance and angle increase, the received signal strength decreases, leading to the observed distance measurement errors. Figure 5.10 indicates that the distance measurements become less precise as

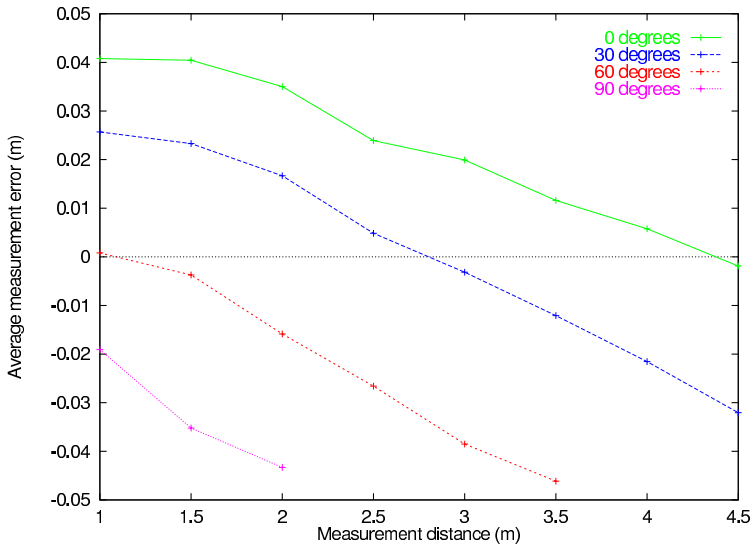


Figure 5.9: Accuracy of receiver distance measurements

the measurement distance and the transmitter-receiver angle increase.

Less than 2% of distance measurements were in error by more than 5cm, suggesting that the use of that threshold as the nominal maximum expected distance measurement error (see Sections 4.3.3 and 5.3.2) was not unreasonable[2]. The maximum range at which the signal from the MPD's transmitter could reliably be detected varied greatly with the transmitter-receiver angle. When this angle was between 0° and 50°, the maximum receiver range was 4.80m, falling to 3.20m at 60°, and then to 1.80m at 90°.

5.5.2 Validation of regression assumptions

According to the assumptions that underlie the use of nonlinear regression for multilateration, as set out in Section 4.3.2, the error terms ε_i in Equation 4.7 should be normally distributed with mean zero and common variance. These assumptions can be validated by using the residuals e_i (calculated by Equation 4.8 for each data set) as estimates of the deviations ε_i.

Multilateration was performed for a representative data set of sixteen

[2]Distance measurements whose error was greater than this value might be rejected as reflections or cause good readings to be rejected as reflections, but, due to their relative infrequency, their impact was expected to be insignificant in the context of the prototype system.

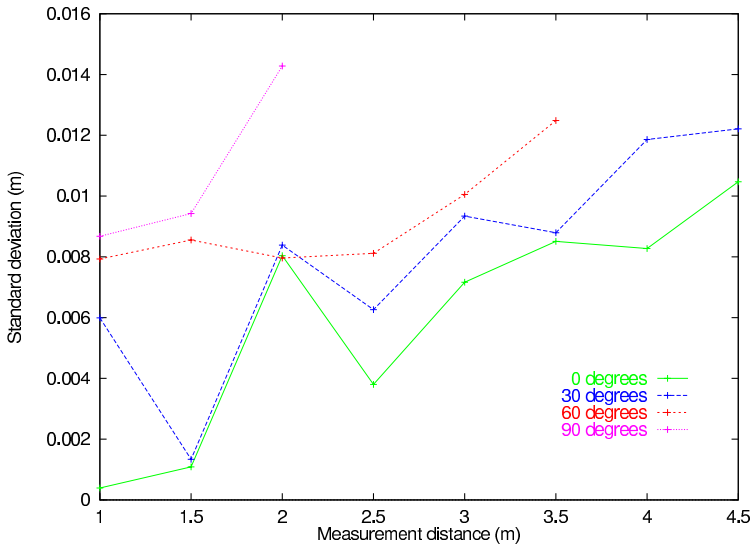
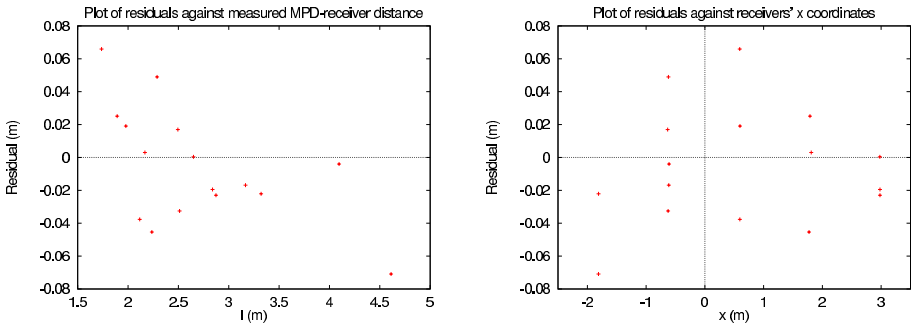


Figure 5.10: Standard deviation of receiver distance measurements

MPD-receiver distance measurements and receiver positions. The calculated residuals were plotted against the variables l, x and y of Equation 4.7, and $\hat{l}$, the predicted values of l determined using Equation 4.9, as in Figures 5.11(a)–(d). The residuals are seen to be distributed around zero, with no discernible systematic dependence of variance on the abscissa, suggesting that the constant variance and zero mean assumptions hold [Glantz90]. A further plot of the expected standardised value (supposing that the residuals *are* normally distributed) of the ith smallest residual against the actual value of that residual yields an approximately straight line passing near the origin, as shown in Figure 5.11(e), indicating that the residuals follow a roughly normal distribution [Myers90].

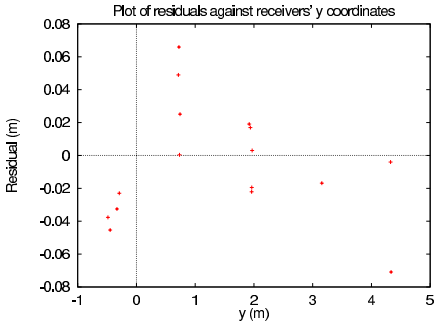
5.5.3 Assessment of position measurements

To determine the accuracy of the tracking system, measurements were made of the position of an MPD placed at known points. A 9×4 grid of points, with grid spacing of roughly 50cm, was laid out on the floor of the room in which the ultrasonic tracker was installed. The positions of the points relative to the ceiling were determined by construction of a trilateration network using a tape measure. A single MPD was consecutively placed at each point, with the transducer block facing upwards and the MPD body in a

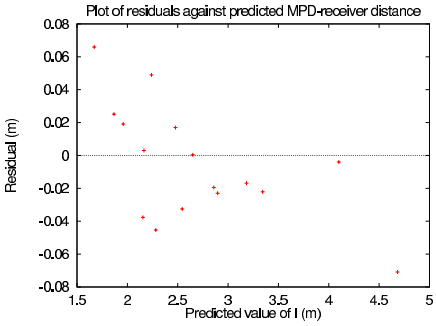


(a)

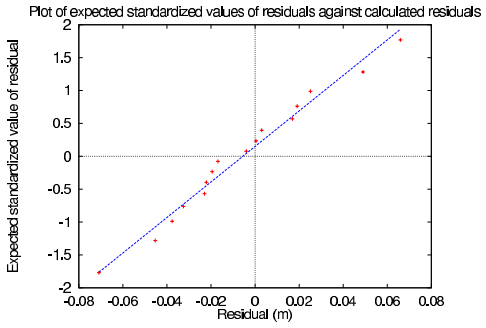
(b)



(c)



(d)



(e)

Figure 5.11: Validation of regression assumptions

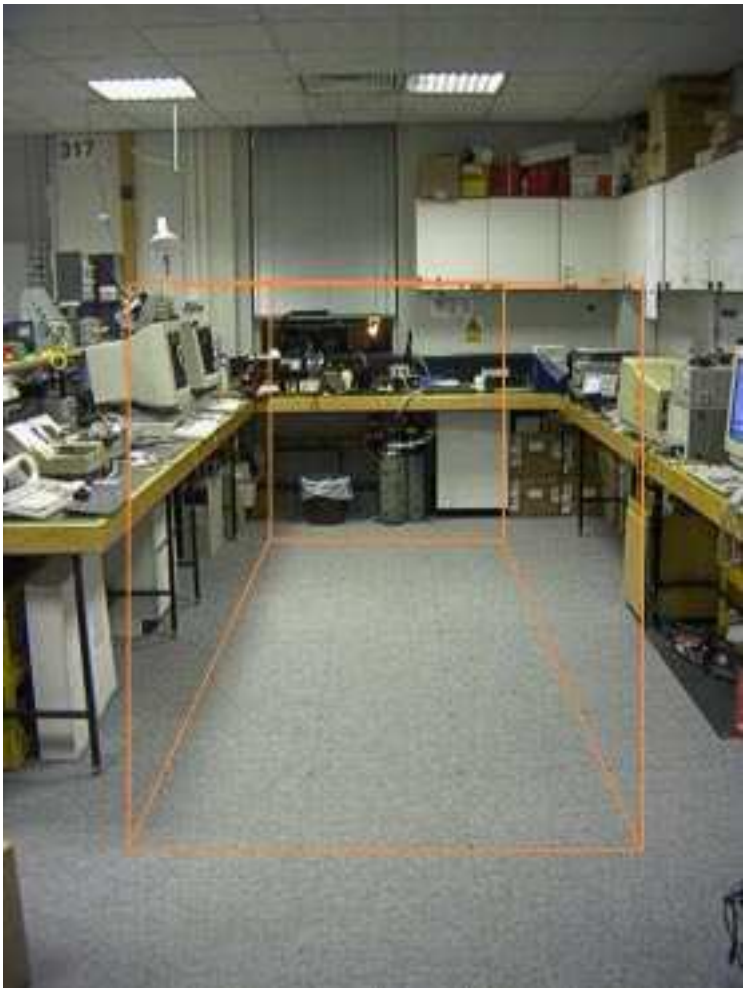


Figure 5.12: Region used for measurement of position accuracy

known horizontal attitude[3] (orientation **A**). 400 measurements of the MPD's position were made before it was rotated 180° around a vertical axis (into orientation **B**). A further 400 measurements of the MPD's position were made, and it was then moved to the next point. This process was repeated at three more levels above the floor (again separated by roughly 50cm), to give a total test volume of around $10m^3$. The use of two MPD orientations ensured that the effects of directional variations in the positioning pulse were minimised. Figure 5.12 shows the test region and its surroundings.

[3]This choice of MPD attitude is favourable, but is not unrealistic—many objects will have conventions of use that allow the MPDs tracking them to be placed such that they always face the ceiling.

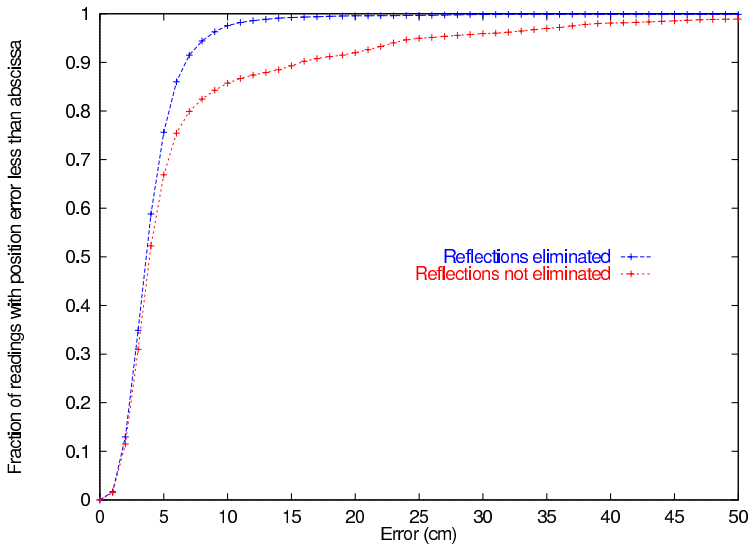


Figure 5.13: Accuracy of ultrasonic tracker

Figure 5.13 shows the cumulative error probability distribution for the raw MPD positions reported by the tracker. The 95% confidence level is just under 9cm. Also shown is the cumulative probability distribution that would be obtained if reflections were not eliminated. Clearly, the algorithms presented in Section 4.3.3 are effective. 3.4% of distance measurements obtained during the tracking assessment were found to be greater than the expected distance by more than 5cm. Given that a fraction of these unusual readings were probably due to distance measurement errors, rather than reflections, this value can be regarded as an upper limit on the average frequency of reflections in the test volume.

The accuracy of the tracking system is strongly dependent on the MPD's particular position, which will affect the number and geometry of receivers that detect its positioning pulse, and the number of error-generating reflections of that pulse. Figure 5.14 shows the system accuracy at different heights above the floor. As can be seen, the accuracy improves as the MPD is moved closer to the ceiling, because (in the test space used) more receivers then have a line-of-sight to the MPD, and there are fewer objects in close proximity that can cause reflections. No significant variation in the tracker's accuracy across the test space was seen, possibly because that space was too small for horizontal motion to result in large changes to the number or geometry of signal-detecting receivers.

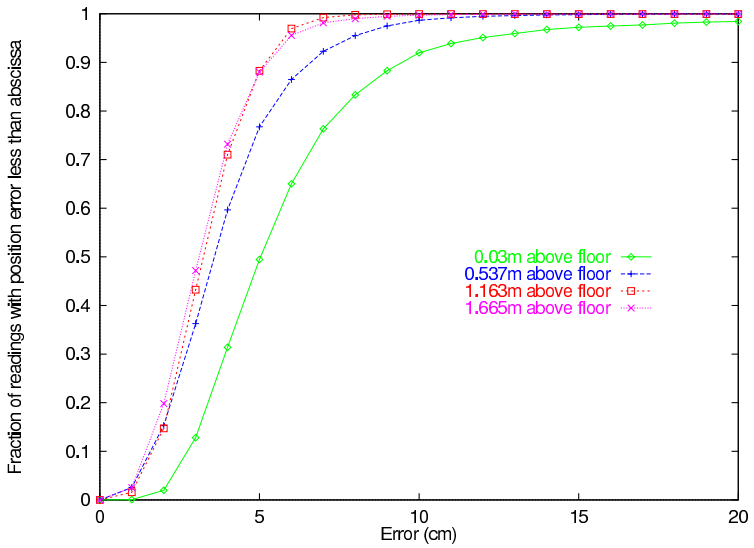


Figure 5.14: Variation of tracker accuracy with height above floor

The tracking system's measurements are affected by two types of error—random errors and systematic errors. Random errors, as their name suggests, are not correlated between readings, and their effects may be reduced by averaging several readings. Systematic errors, such as distortions of the measurement space, are correlated between readings, and cannot be averaged out. Figure 5.15 shows the effects of averaging readings from the same test point on the accuracy of the tracking system. As can be seen, the improvement in accuracy is modest, suggesting that the position measurement errors are mostly systematic.

To determine the extent of spatial distortions of the measurement volume, all measurements made in each MPD orientation at each test point were averaged, with the intention of minimising any bias due to random errors. The resulting systematic position offsets at each test point and orientation at a level 1.163m above the floor are shown in Figure 5.16—no obvious trends in the distortions are evident. A large spatial bias, with magnitude 20cm, was seen at one test point at floor level, with the MPD in orientation **A**. This gross distortion was probably caused by reflected signals that were consistently passed over by the reflection rejection algorithms.

Figures 5.17(a) and (b) show the degree to which the measurement volume distortions are spatially correlated—the magnitude of the difference vector between the distortions at pairs of points is plotted against the dis-

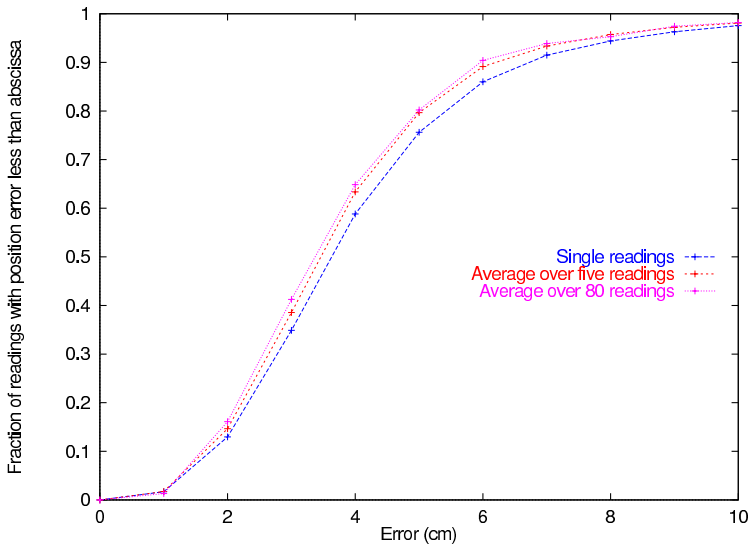


Figure 5.15: Effects of averaging of readings on tracker accuracy

tance between those points. The binned means of the difference vector magnitude (also plotted in Figures 5.17(a) and (b)) show only a slight trend towards greater spatial correlation (i.e. smaller difference vector magnitude) of the distortions at closely-spaced points in the measurement volume. This observation suggests that no simple calibration of the tracker's operating space could significantly reduce the effects of distortion of that volume.

Tests were conducted to determine the extent of temporal correlations between positions reported by the tracker. The average magnitude of the difference vector between 900 measurements of the position of a stationary MPD and further measurements taken between 40ms and 4s later was determined. The results are shown in Figure 5.18—it appears that two measurements of a stationary MPD's position are more likely to be similar if the interval between them is 0.2s or less, possibly because no great variation in atmospheric conditions across the tracker's operating volume occurs over such short timescales.

Figure 5.19 shows the volume within which 1000 measurements of a stationary MPD's position fall, together with projections of the density of points in the volume along the **x**, **y** and **z** axes. The average dilution of precision of measurements along the **x**, **y** and **z** axes was found to be 0.55, 0.59 and 0.40, respectively—the relative dimensions of the volume along each axis approximate those values, illustrating the way in which random

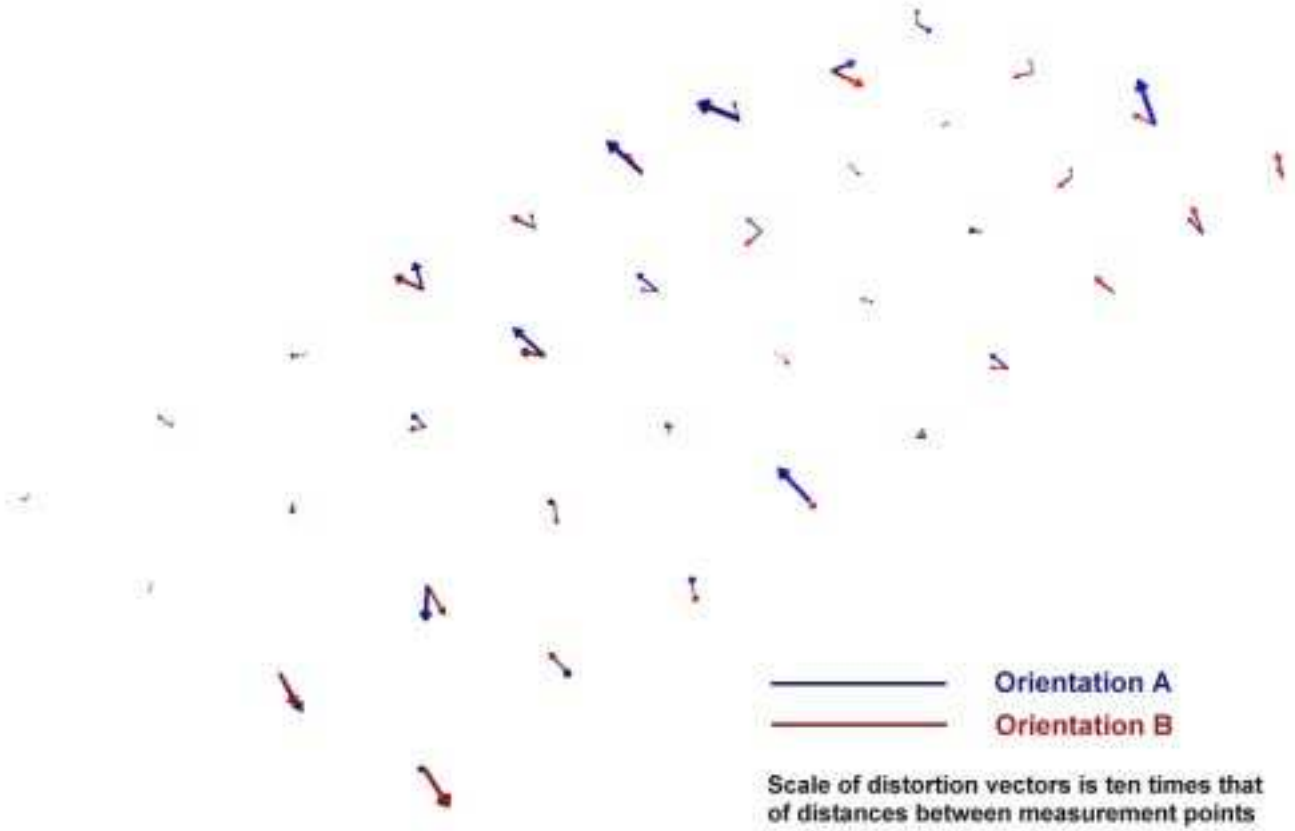
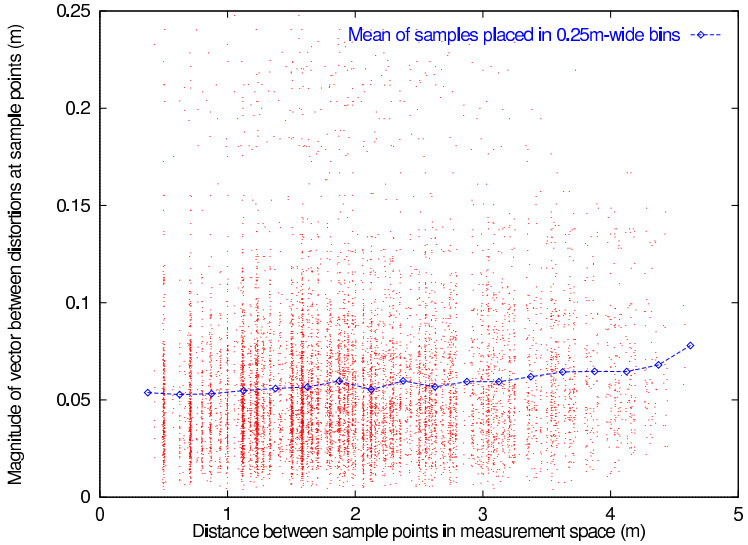
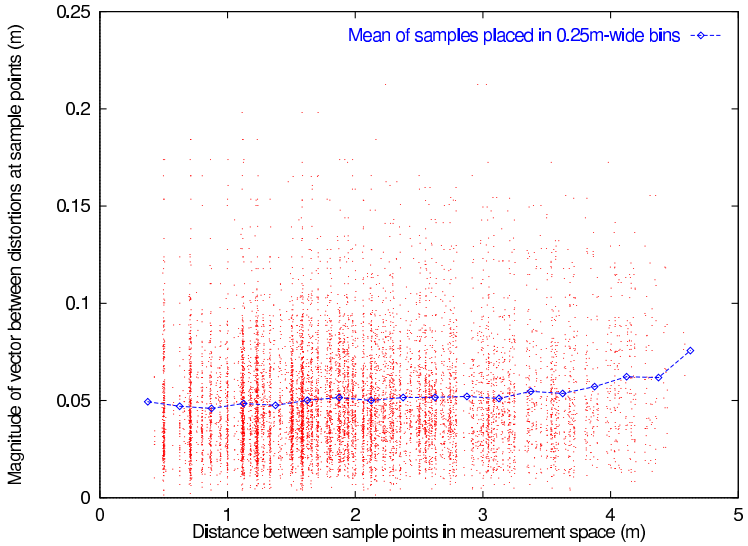


Figure 5.16: Distortion of tracker measurement volume (1.163m above floor)



(a) Orientation **A**



(b) Orientation **B**

Figure 5.17: Spatial correlation of measurement space distortions

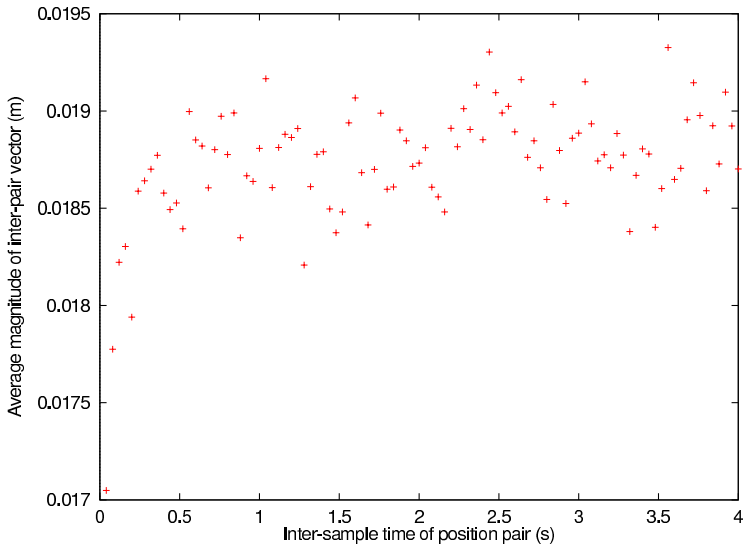


Figure 5.18: Temporal correlation between position measurements

errors have been magnified by the measurement geometry.

Figure 5.20 shows the relationship between the PDOP calculated for each measurement taken in the test volume, and the position error of those measurements. In general, measurements with high values of PDOP are associated with large position errors. This relationship can be used to identify and reject measurements that might have a large error. Figure 5.21 shows the increased tracker accuracy obtained by rejecting readings with a PDOP greater than or equal to 2 (approximately 10% of all readings taken). When filtering measurements in this way, it was found that no good positions were computed for the MPD at ten of the 288 test points, regardless of the MPD's orientation. These points were located in regions of the test space with limited visibility of the ceiling.

5.5.4 Assessment of orientation measurements

To evaluate the system's ability to find the orientations of objects using multiple MPDs, a test frame with four MPDs (shown in Figure 5.22) was placed at various combinations of roll, pitch and yaw angles between 0° and 60°. Two sets of three MPDs (also shown in Figure 5.22), were selected from the four, and their measured positions at each orientation were used to estimate the attitude of the test frame. Figures 5.23(a)–(c) show the accuracy of

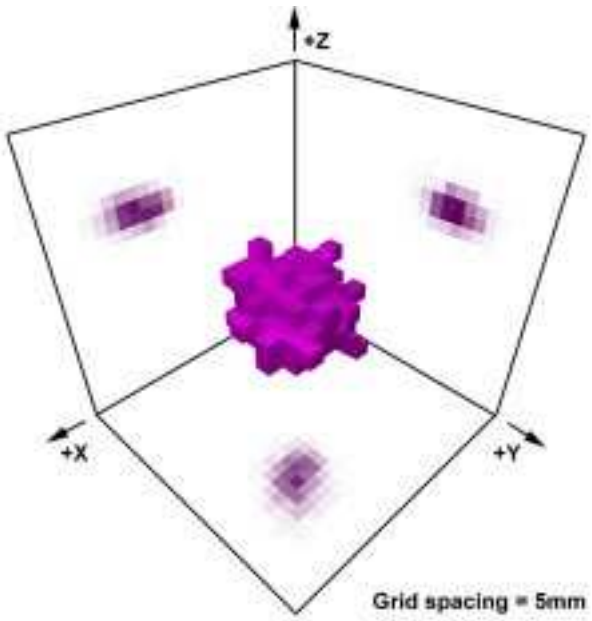


Figure 5.19: Volume containing position measurements from one test point

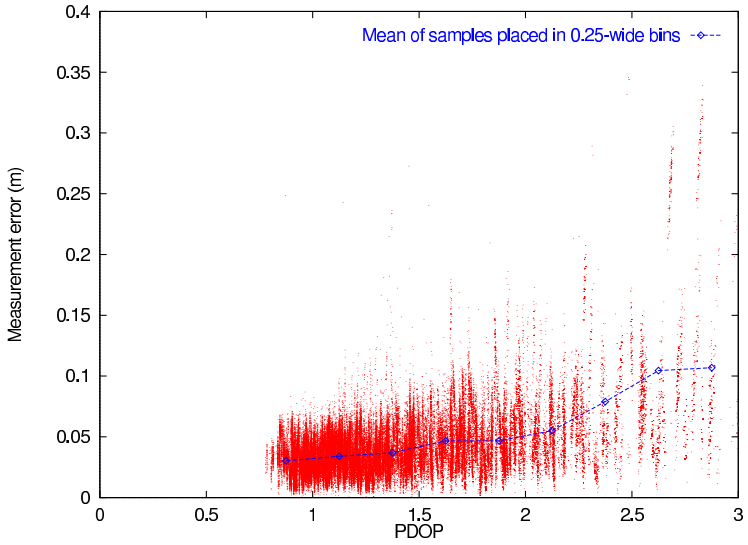


Figure 5.20: Variation of measurement error with PDOP

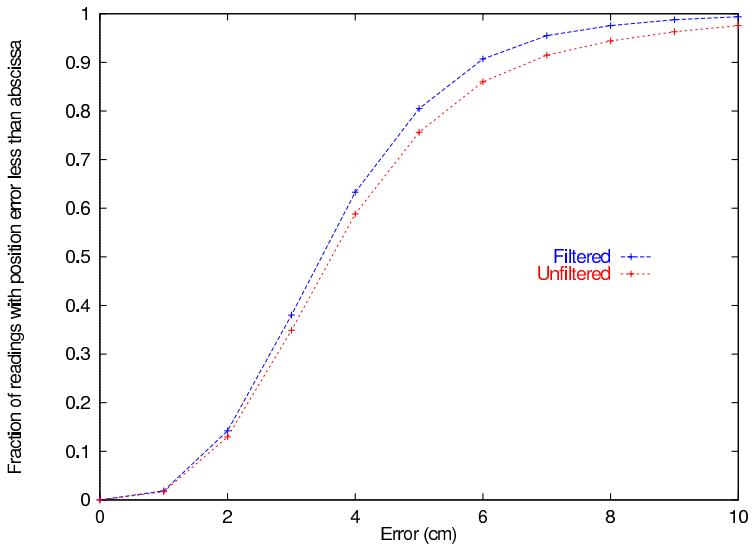


Figure 5.21: The effects of filtering using PDOP

the roll, pitch and yaw estimates made using each MPD set. The triangle formed by MPD set 1 is larger than that formed by MPD set 2 around the roll and yaw axes, and should therefore have allowed more accurate orientation measurements around those axes. These observations are borne out by Figures 5.23(a) and (c).

The method of finding object orientations using a single MPD (Section 4.4.2) was also assessed. A person wearing an MPD on a chain around their neck stood in the centre of the test volume described above. Estimates were made of the direction in which the person was facing as they turned through a complete circle in 22.5° increments. Figure 5.24 shows the accuracy of the heading estimates, which, although not as accurate as the measurements made using several MPDs, are sufficiently good to identify, say, the set of objects in front of a person.

5.5.5 Effects of environmental noise

A number of devices in the office environment (for example, monitors and PC cooling fans) can generate ultrasound. Tests were therefore conducted to determine the nature of the ultrasonic component of this ambient noise, and the sensitivity of the system to ultrasound generated by sources other than MPDs.

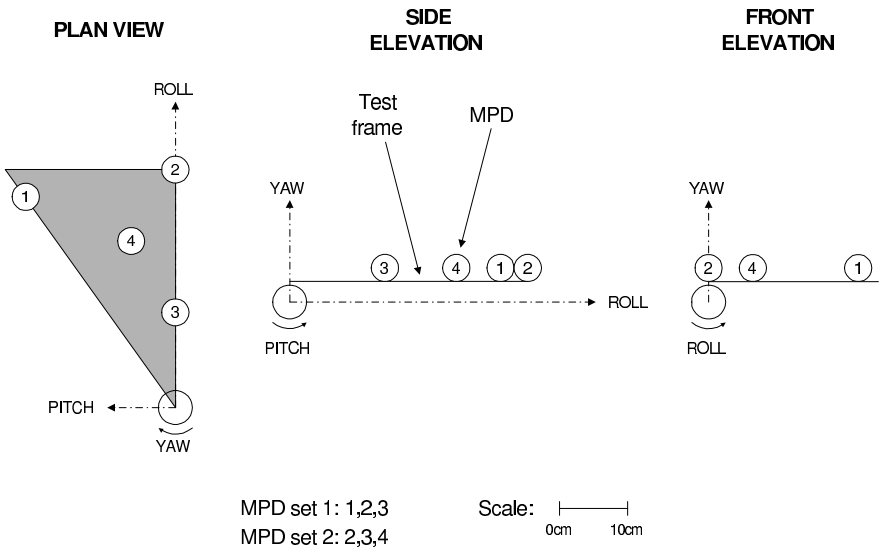


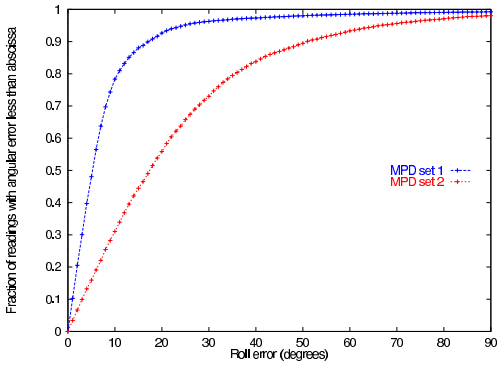
Figure 5.22: Test frame for orientation measurements

Figure 5.25 shows the power spectrum of the ambient noise detected by receivers. The power spectrum was generated from the averaged FFTs of fifty 5ms sample windows. It was found that highest noise peaks (caused by emissions from CRTs around the test space) were above 50kHz, and were not sufficiently intense to affect the operation of the receivers.

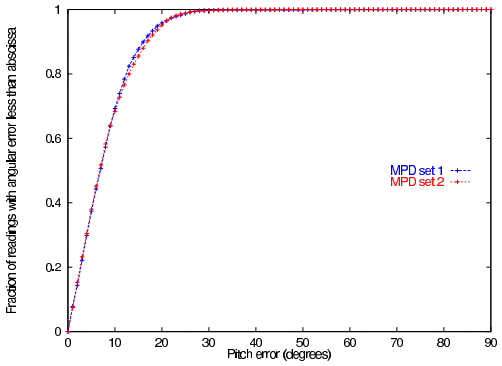
The typical frequency with which receivers were triggered while the ZM was switched off (and hence when no MPDs were emitting ultrasound) was measured. It was found that a false signal might be reported by one of the 38 receivers in roughly 0.5% of all timeslots. The effects of such false readings on the positions reported by the system depend on whether or not they are rejected as reflections, and on how many good signals were reported.

5.5.6 Power management evaluation

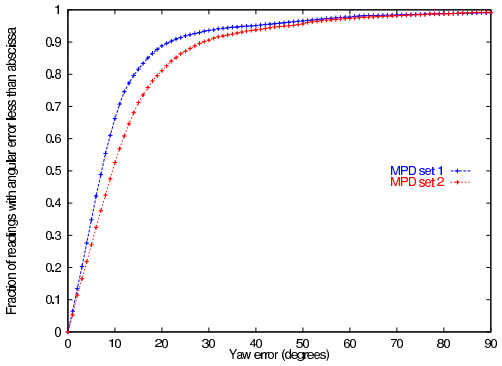
Estimates for the MPD battery lifetime can be obtained using measurements of the current drawn by the MPD in its various operating modes and the known capacity of the batteries. When an MPD is in the slumber state, only the slow clock and FPGA are active, and the current drawn from the MPD's battery is 50μA. When the fast clock and receiver are also active, the MPD current drain is 15.46mA. The Lithium Thionyl Chloride cells have a capacity of 1Ah.



(a) Roll



(b) Pitch



(c) Yaw

Figure 5.23: Accuracy of orientation measured using multiple MPDs

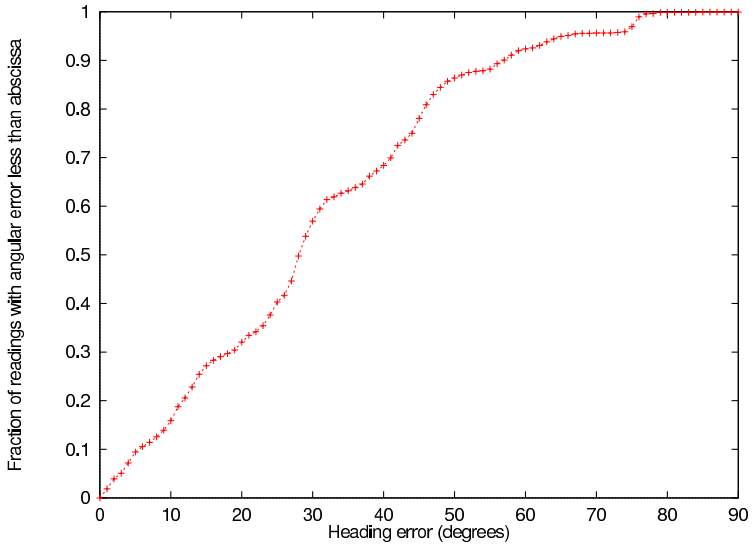


Figure 5.24: Accuracy of orientation measured using a single MPD

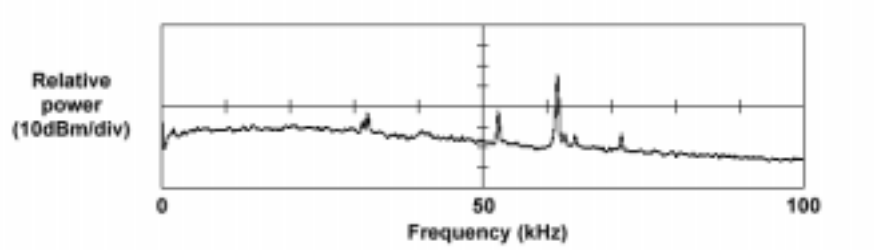


Figure 5.25: Power spectrum of ambient noise detected by receivers

Consider an MPD that checks every addressing message from the ZM. When decoding a message, the fast clock and receiver are active for a total of around 6ms. Since there are 25 addressing messages each second, the average current consumption of the MPD is 2.36mA, indicating an estimated battery lifetime of 18 days. The small amount of current drawn by an MPD's ultrasonic transducers (< 2mA) is ignored in this analysis, because they are driven only when that particular MPD is addressed, and then only for a very short time. A corresponding computation for an MPD that cannot detect signals from a ZM indicates an average current consumption of 200μA, and an expected battery lifetime of around 210 days.

Similar calculations were made for an interim system that was used over a period of some time. Taking into account the typical patterns of use of the system, MPD battery lifetimes of around 140 days were predicted. In practice, battery lifetimes of around 100 days were observed, suggesting that the above estimates are not unreasonable.

5.5.7 Safety evaluation

Any device that is intended for everyday use must meet generally accepted safety standards. Use of the ultrasonic tracking system will expose persons in the immediate environment to RF signals and ultrasonic signals, and their possible effects on human health must be investigated.

Several studies have raised concern about the safety of human exposure to the RF signals emitted by personal communications systems such as mobile phones (an overview of this subject is given by Lin [Lin97]). However, the radio signals used to synchronise the MPDs with the static elements of the system are very weak (the maximum emitted power is 0.25mW), and lie well within suggested exposure limits for 418MHz RF energy [FCC96].

A more serious concern is that ultrasonic energy generated by MPDs could have detrimental effects. In order that positioning pulses are detected by as large a number of receivers as possible, the MPDs emit relatively strong ultrasonic signals. Some data indicate the possibility of hearing loss from laboratory exposure to very intense ultrasound, although no well defined threshold for this effect has been determined [IRPA84]. Furthermore, industrial workers exposed to more moderate airborne ultrasound have complained of subjective symptoms such as nausea, tinnitus, and headaches [Acton74]—such symptoms are considered to be more sensi-

tive indicators of potential harm than hearing threshold shifts. A number of investigators have therefore recommended that the maximum SPL of 40kHz airborne ultrasound (measured at the ear) to which humans should be routinely exposed is 110dB [Acton74] [IRPA84].

Accurate peak SPL measurements of the MPD's positioning pulse are complicated by its very short duration. However, calculations made using the data sheets for the MPD's transducers [ProWave89] indicate that the SPL of the positioning pulse should certainly be less than 110dB at distances of 2.4cm or more from the hemispherical transmitter block. It is very unlikely that personnel will approach an MPD to within this distance other than by deliberate action. To discourage misuse it would seem prudent to label MPDs with warnings similar to those placed on laser pointers, which could also potentially cause harm if used improperly. No subjective symptoms have been reported by those personnel who have worked with the prototype system.

5.6 Summary

This chapter has described the implementation of a wireless, low-power 3D tracker that can be used in indoor spaces, and has evaluated that system. Position measurements made by the tracker were found to be accurate to within 9cm in position (95% confidence level), and two methods of finding the orientations of objects were assessed. The reflection elimination algorithms presented in Chapter 4 have been shown to be highly effective, and a discussion of the types of error affecting the tracker's performance has been made. Furthermore, it has been shown that the tracker's safety characteristics do not preclude its use in the office environment for which it was intended.

Chapter 6

System Scalability

6.1 Introduction

Chapters 4 and 5 describe an indoor tracking technology that has a number of characteristics which make it suitable for the implementation of context-aware systems. However, the tracker, as described, also has a number of limitations. The set of MPDs that are tracked is static, as are the rates at which each MPD is tracked. Furthermore, the area over which objects may be tracked by the basic system is limited. This chapter addresses these limitations and related issues.

6.2 Resource Allocation

Since the tracking system can only locate a certain number of MPDs in any time period, efficient distribution of timeslots to the MPDs that must be tracked is a priority. This section describes methods by which timeslots may be allocated to MPDs.

6.2.1 Scheduling

The process of timeslot distribution is called *scheduling*, and is carried out by an algorithm called the *scheduler*. The allocation of timeslots generated by the scheduling algorithm is called the *schedule*. A priority level called the *Location Quality of Service* (LQoS) is assigned to each MPD. Although no guarantee can be made of the rate at which an MPD will be located (an MPD in a closed box will never be located, regardless of the resources devoted to

it), the LQoS is expressed as a real number representing the desired period of location. For example, an LQoS of 25 requests one position every 25 timeslots.

The scheduling environment is highly dynamic, and the LQoSs requested by users or applications for particular MPDs may change dramatically over a period of a few seconds. For example, the tracker might normally monitor MPDs carried by people (who move often) a few times a second, and it might monitor those attached to workstations (which move infrequently) once a minute. If a person was seen to walk up to a workstation, however, the tracker might then monitor the workstation's position once a second, because it would then be more likely to be moved. For this reason, the scheduling algorithm should be equally dynamic, allocating an MPD ID to a timeslot just before the start of that timeslot, based on the current location priorities and the scheduling history.

A simple, dynamic scheduling algorithm has been developed for ultrasonic tracking, and details are presented in Appendix B. Each MPD managed by a ZM is allocated an entry in a *scheduling table* stored by that ZM. The entry contains the LQoS requested for that MPD, and a value called the *score*, which is a measure of the scheduling history for that MPD. At the start of each timeslot, the scores of all MPDs in the scheduling table are examined, and that timeslot is allocated to the MPD with the highest score. The scores are then updated, based on the current LQoS of each MPD. Changes to the LQoS of any MPD may be made by altering the contents of the scheduling table appropriately. The scheduler can deal with service demands both greater and less than the system capacity, by scaling down the excessive requests fairly in the former case, and padding with dummy timeslot assignments in the second.

6.2.2 Group allocation

One constraint on the allocation of timeslots is the desire to locate members of certain groups of MPDs in consecutive timeslots. This behaviour is advantageous when the positions of a group of MPDs are used to determine both the location and orientation of an object (see Section 4.4.1). In order to determine which MPDs should be linked in this way, each MPD must be a member of one, and only one, *MPD group*, identified by a unique integer. To simplify the scheduling of MPD groups, a restriction that all MPDs in

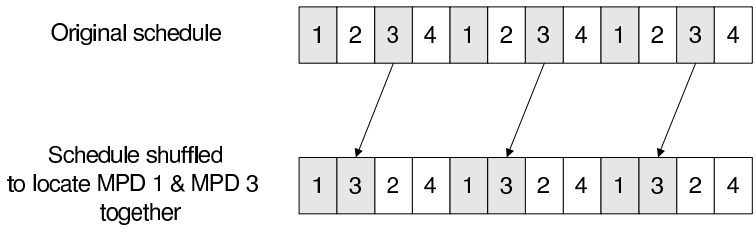


Figure 6.1: Timeslot swapping

the same group must have the same LQoS is imposed. A request to change the LQoS of one MPD in a group therefore simultaneously changes the LQoS of all members of that group.

The scheduling algorithm described in Appendix B allocates timeslots to MPD groups using a principle called *timeslot swapping*. A schedule that takes no account of MPD groups is shuffled so that MPDs in the same group are located in adjacent timeslots, as shown in Figure 6.1. Since the frequencies with which particular MPDs are located are not changed (when viewed over a long period of time), if the original schedule matches the LQoS demand for each MPD, so will the reordered schedule.

6.2.3 Power management

Some MPDs, e.g. those which track objects that move infrequently, will be addressed by the ZM only once every few minutes, or even only once every hour. In such cases, it would seem wasteful to require that the MPD check every incoming addressing message, since only a very small fraction of those messages will be relevant to that MPD. Instead, when a timeslot is allocated to an MPD, the scheduling algorithm presented in Appendix B can use the LQoS and score information in the scheduling table to calculate the earliest time at which that MPD will next be addressed. Until that time, the MPD can enter a long-term, low-power *sleep* state in which the MPD's receiver and fast clock are turned off, resulting in a considerable power saving.

Addressing messages are extended to include a Sleep Time field, which informs the addressed MPD of a conservative estimate of the number of timeslots that will elapse before it is addressed again. If the LQoS demand of any of the MPDs managed by the ZM changes, however, the schedule will change, and the estimate may cease to be conservative. For this reason, there must be some mechanism by which MPDs in the sleep state can be

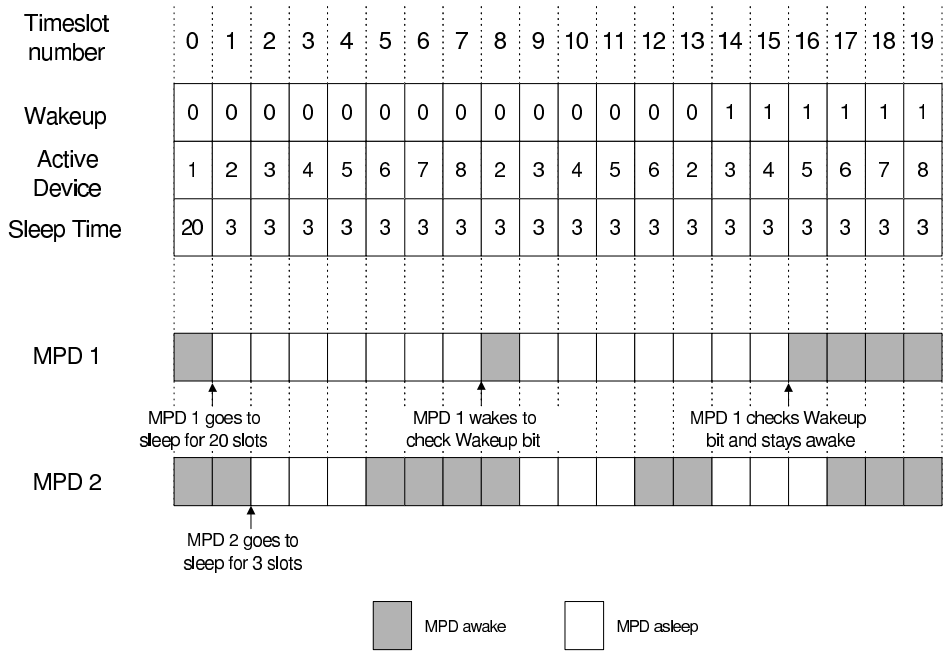


Figure 6.2: Behaviour of MPDs in sleep state

commanded to revert to checking each addressing message. The addressing messages are therefore further extended to include a Wakeup bit at the start of each message. MPDs in the sleep state wake periodically to check the Wakeup bit of every eighth addressing message[1]. If an MPD detects that the Wakeup bit in an addressing message is set, it will check every incoming addressing message until it receives an indication to return to the sleep state.

The ZM will usually clear the Wakeup bit in every addressing message it sends out. If a request to change the LQoS of an MPD is received by a ZM, however, it sets the Wakeup bit in the next eight timeslots. After that time, any MPD that was in the sleep state will have begun to check each incoming addressing message, and all MPDs will be able to respond instantly to the new LQoS demands, as shown in Figure 6.2. Only at this stage are changes made to the scheduling table, resulting in a latency of eight timeslots before a request to change an MPD's LQoS takes effect.

It can be seen that MPDs will spend little time in the sleep state when changes are being made frequently to the LQoS demands in the scheduling

[1]Whilst checking the Wakeup bit, MPDs can resynchronise their local clocks with that of the ZM, ensuring that clock drift does not cause an MPD to sleep for too long.

table. However, during stable periods (for example, overnight), many MPDs will check only one in eight of the addressing messages that they would otherwise monitor. It can also be seen that a compromise must be made between the frequency with which MPDs in the sleep state check addressing messages and the latency with which changes can safely be made to the scheduling table.

6.3 Dynamic Population Management

The tracking system described in Chapter 5 determines the positions of a fixed set of MPDs. When objects that may move in and out of the operating volume of the tracker are monitored, this approach is undesirable—it may become necessary to locate new MPDs as they enter the operating volume, and a fixed allocation of tracking resources may result in waste of those resources on untrackable objects that have left the volume. Mechanisms are therefore required for *registration* (addition of MPDs to the list managed by a ZM) and *resource reclamation* (deletion of MPDs from that list).

6.3.1 Registration

Section 5.2.3 describes how an untracked MPD searches periodically for signals from a ZM. When an MPD detects a nearby ZM, it must assume that the ZM has no knowledge of its presence, and that no timeslots will be allocated to it. The MPD must therefore identify itself to the ZM, allowing that ZM to determine an appropriate LQoS with which to track the MPD. Once a successful registration has occurred, the MPD must be told that further registration attempts are unnecessary.

Registration in this manner requires a bidirectional communication channel between the ZM and the MPD. The radio link used to transmit addressing messages to the MPD provides one half of the channel. It is suggested that this radio link should be made bidirectional, so as to also provide the necessary back-channel. In principle, the new MPD could transmit its ID to the static elements of the tracker using ultrasonic signals. However, use of the ultrasonic channel for transmission of registration information would restrict use of that channel for measurement of MPD-receiver distances with positioning pulses, due to constraints on the consecutive transmission of ultrasound (see Section 4.2.6). This would be undesirable, because the track-

ing capacity of the system is determined by the rate at which positioning pulses are sent.

In the proposed registration system, MPDs have the opportunity to transmit an *announcement message* to the ZM after receipt of each addressing message, as shown in Figure 6.3. After transmitting the outgoing addressing message, the ZM listens for an incoming announcement message from an MPD. Each announcement message contains at least the unique ID of the MPD that transmits it. Since more than one MPD may wish to transmit an announcement message in any one timeslot, some method of resolving contention between MPDs must be implemented.

If a ZM receives a valid announcement message, it updates its scheduling table by adding a request for the new MPD. The LQoS associated with that request may be a default rate for incoming MPDs, or may be a specific rate previously assigned to the new MPD by client applications. To provide MPDs with an indication that registration has been successful, addressing messages are extended to include a field called the Registration Acknowledgement. The ZM encodes the ID of the newly-registered MPD in this field of the next addressing message, as shown in Figure 6.3. If an MPD receives an addressing message with either an Active Device or Registration Acknowledgement field matching its ID, it can be certain that the ZM is aware of its presence, and can cease registration.

6.3.2 Resource reclamation

When an MPD moves out of range of a ZM, it would be advantageous to reclaim any location resources allocated to it. There are two mechanisms by which a ZM can determine that an MPD is still potentially trackable. First, the detection by receivers of a positioning pulse from an MPD indicates that the MPD is still within range of the ZM. Second, the ZM can request that a particular MPD transmits a radio acknowledgement, receipt of which also indicates that the MPD is within range of the ZM. Unfortunately, if an MPD's positioning pulses are temporarily shielded (by objects moving around it), use of the first method may incorrectly suggest that its allocated resources can be reclaimed. The second method is potentially a more reliable indicator, but it uses more power. A combination of the two methods would therefore seem to be appropriate.

Figure 6.4 illustrates the proposed approach to resource reclamation.

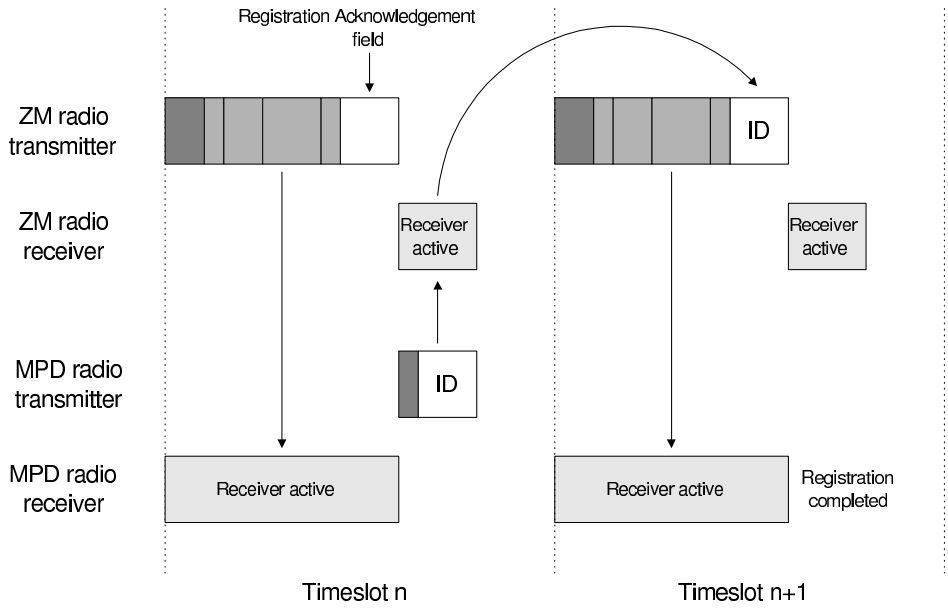


Figure 6.3: Registration protocol

The AM informs the ZM when an MM successfully finds the position and orientation of an MPD in a timeslot. If the ZM addresses an MPD a number of times without being informed that its location has been determined, the MPD may have moved out of range of the ZM. To ensure that the pulses are not blocked, addressing messages are extended to include a Transmit ID bit, which is set by the ZM when the MPD is next addressed. When an MPD receives an addressing message with an Active Device field matching its ID, and the Transmit ID bit set, it must send an announcement message in that timeslot. Other MPDs receiving that addressing message may not attempt to send an announcement message in that timeslot, to avoid contention.

If the ZM fails to receive the desired announcement message in response to its request, it assumes that the MPD has moved out of range. Addressing messages are also extended to include a Drop bit. When the MPD is next addressed, the ZM sets the Drop bit to indicate that the MPD's location resources have been reclaimed, and then deletes the entry in its scheduling table associated with that MPD. Any MPD receiving an addressing message with an Active Device field matching its ID, and the Drop bit set must reregister with the ZM—this situation may occur if, for example, the MPD's response to a Transmit ID request is corrupted by noise.

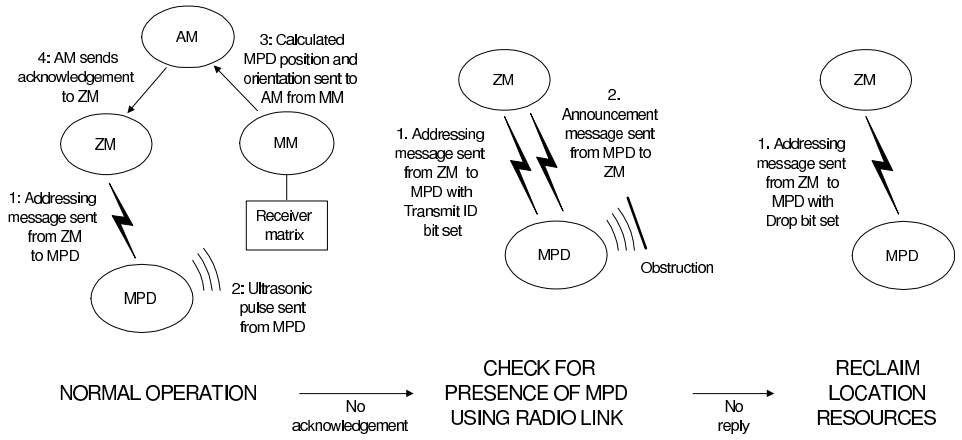


Figure 6.4: Resource reclamation

MPDs must ensure that they do not miss indications that resources allocated to them have been reclaimed. For this reason, addressing messages must contain error-detecting information, allowing MPDs to identify when messages have been corrupted. If a corrupted addressing message is received by an MPD, it must conservatively assume that the message was destined for it (i.e. the Active Device field matched its address), and also that the message contained a set Drop bit. In such cases, the MPD should attempt to reregister with the ZM.

6.4 Multiple Ultrasonic Spaces

The system described in Chapters 4 and 5 tracks MPDs within a single room, using one receiver matrix. This section investigates ways in which the system can be adapted to cover larger areas, which may be either open-plan or comprised of several rooms. Methods are also described by which the system's tracking rate can be improved to handle the larger number of objects in that greater area.

6.4.1 Tracking MPDs in multiple rooms

The system described in Section 4.2 can be extended to cover multiple rooms by equipping each room with a receiver matrix and MM, as shown in Figure 6.5. In order that MPDs may be triggered in each room, those rooms must lie within the radio coverage of the ZM's transmitter. All MMs are

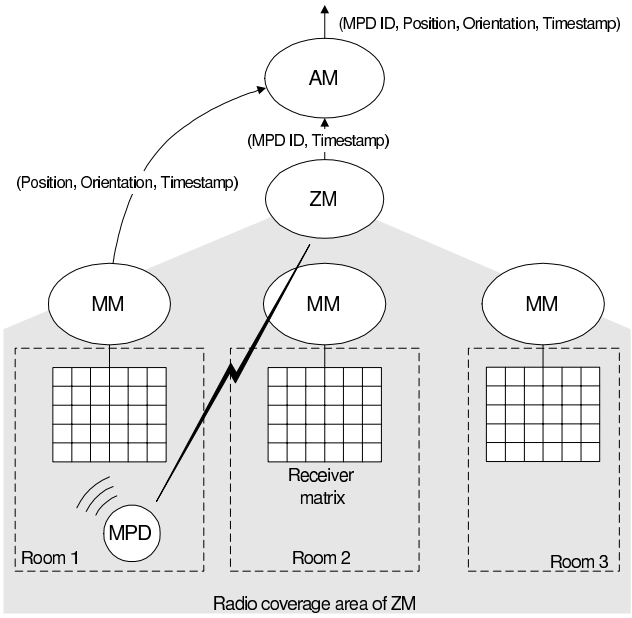


Figure 6.5: Tracker operation with several receiver matrices

synchronised, so that they can identify the start of timeslots. At the end of each timeslot, MMs send the MPD position and orientation calculated from any signals seen by receivers in that timeslot to the AM. Since only one MPD is triggered by the ZM in each timeslot, only one MM should send MPD information to the AM in that timeslot. Receipt by the AM of more than one result would indicate a noise event in one or more rooms. The AM would then have to use heuristics (based on historical location information) to attempt to identify the genuine result, or would have to discard all results for that timeslot.

6.4.2 Increasing the system capacity

Section 4.2.6 sets out the requirement that the interval between emission of positioning pulses from MPDs in the same room must be greater than the signal reverberation time, so that those pulses can be clearly distinguished. This places an upper limit on the rate at which the tracker can address MPDs. However, if several MPDs are located in separate, ultrasonically isolated rooms, then the intervals over which their positioning pulses are in flight may be safely overlapped, without the possibility of signal interference. This observation suggests that a ZM could address two or more

carefully chosen MPDs simultaneously, potentially allowing a higher aggregate location rate.

Consider a number of MPDs, thought to be in separate rooms, that are triggered simultaneously. If the number of rooms in which signals were detected is less than the number of MPDs triggered, all signals must be discarded—several MPDs could, in fact, have moved into the same room, and their positioning pulses may have interfered[2]. Otherwise, if the number of rooms in which signals were detected is equal to the number of MPDs, no signal interference can have taken place. However, the MPDs' positioning pulses contain no addressing information, and a set of signals detected in a room could have been generated by any of the addressed MPDs. It would be impossible to know with certainty which MPD was in which room—past location information for MPDs might provide some guidance in this matter, but would not rule out the possibility that several of the MPDs had been transposed.

In order to distinguish between positioning pulses from MPDs, the times at which they emit those pulses are staggered by small intervals. The iterative nonlinear regression calculation used to determine the MPD positions and reject reflections (see Section 4.3.3) should only report a solution if it is given a set of consistent MPD-receiver distances and receiver positions. If a set of signals detected by receivers in a room are ascribed to the wrong MPD, the positioning pulse times-of-flight calculated from those signals will be incorrect, because the wrong trigger time will have been assumed. The corresponding MPD-receiver distances will then be incorrect, and so the iterative regression computation will not succeed. Furthermore, a positioning pulse that is detected before a particular MPD has been triggered cannot have been generated by that MPD, providing extra information with which to ascribe signals to MPDs in certain circumstances.

To enable near-simultaneous triggering of MPDs, addressing messages are extended to include a number of Active Device fields, each associated with a different trigger time after the start of the timeslot, as shown in Figure 6.6. MPDs receiving addressing messages check their ID against all Active Device fields—if one matches, the MPD emits a positioning pulse at the corresponding trigger time. Timeslots are extended so that the interval between the last trigger time and the end of the timeslot is equal to the

[2]Alternatively, an MPD may have been obscured, but the worst case must be assumed.

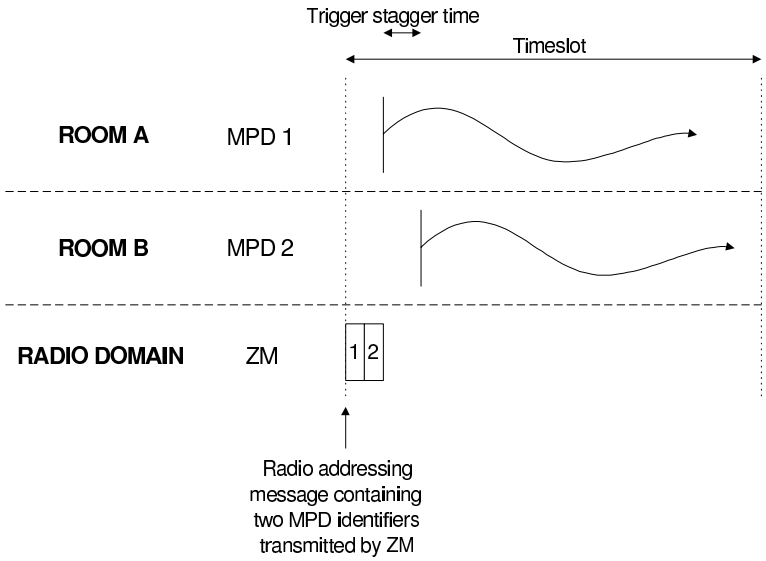


Figure 6.6: Staggered triggering of MPDs

maximum ultrasonic reverberation time in any of the rooms covered by the system.

Clearly, the effectiveness of the staggered triggering method depends on the ability of the scheduler to select sets of MPDs that may be located simultaneously. The scheduler could use the last known MPD locations, and knowledge of whether two points are in the same ultrasonic space, to facilitate this selection. If the scheduler is unable to find a set of MPDs that are unlikely to interfere if triggered together in a timeslot, it must address only a single MPD in that timeslot.

Simulations were conducted to determine the effectiveness of the staggered triggering method. The simulations envisaged two rooms, A and B, with an MPD placed at a random point in each, a reverberation time of 20ms, and the MPDs triggered with a stagger time of 6.6ms. 10% of the simulated MPD-receiver distance measurements were affected by reflections. Two iterative regression calculations, one for each of the trigger times (identified as 1 and 2), were computed for signals from each room. The function presented in Table 6.4.2 was used to identify the trigger times (and hence identity) of the MPD in each room, based on the success or failure of the four iterative regression calculations (A1, A2, B1 and B2). 1000 trials were made, and it was found that signals detected by the receivers in each room

Iterative regression successful?				*Deduced MPD trigger time*	
A1	A2	B1	B2	Room A	Room B
×	×	×	×	?	?
×	×	×	✓	1	2
×	×	✓	×	2	1
×	×	✓	✓	?	?
×	✓	×	×	2	1
×	✓	×	✓	?	?
×	✓	✓	×	2	1
×	✓	✓	✓	2	1
✓	×	×	×	1	2
✓	×	×	✓	1	2
✓	×	✓	×	?	?
✓	×	✓	✓	1	2
✓	✓	×	×	?	?
✓	✓	×	✓	1	2
✓	✓	✓	×	2	1
✓	✓	✓	✓	?	?

Table 6.1: Function for identifying MPD trigger times

could be ascribed to the correct MPD in 90% of cases. No signals were incorrectly ascribed—this is highly desirable, given that MPDs would otherwise be placed in the wrong room. In practice, the last known MPD locations might be used to filter out any spurious readings caused by incorrect signal identification. With these system parameters, two MPDs may be located in a period of 26.6ms, therefore increasing the potential aggregate location rate from 50Hz to 75Hz.

6.4.3 Large ultrasonic spaces

Some applications of the tracking system may require its installation in very large rooms, such as open-plan offices or warehouses, which may also contain many MPDs, implying a requirement for a fast tracking rate. Since the space in the room is continuous, covered using a single MM and receiver matrix, the method of increasing system capacity described in Section 6.4.2 cannot

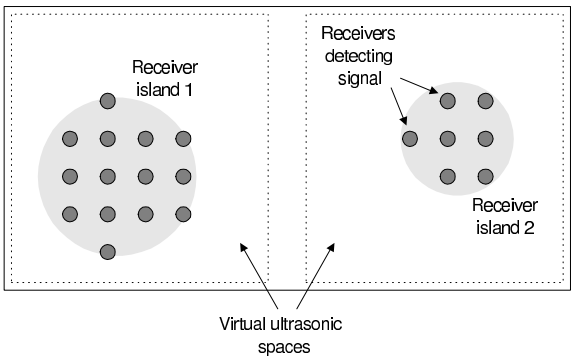


Figure 6.7: Identification of virtual ultrasonic spaces in a large room

be applied directly. However, if two MPDs are triggered almost simultaneously, but those MPDs are sufficiently far apart that their positioning pulses do not overlap in space, the sets of signals detected by receivers can be separated as if isolated "virtual" ultrasonic spaces existed within the room, as shown in Figure 6.7. The methods described in Section 6.6 can then be used to ascribe sets of signals to particular MPDs. When selecting sets of MPDs to be simultaneously tracked in this way, the scheduler must use historical location information to find MPDs that are likely to be some distance apart. Again, if no such set of MPDs can be determined, the scheduler must address only a single MPD in that timeslot.

If MPDs are triggered almost simultaneously in a large room of this kind, checks must be made to ensure that their positioning pulses have not overlapped in space—the number of distinct *islands* of receivers that have detected signals (see Figure 6.7) must be equal to the number of MPDs that were triggered. It can be shown that two receivers separated by a distance of more than twice their range cannot detect signals from the same MPD. This observation allows islands to be identified by partitioning the receivers that detected signals in a timeslot, based on their positions and the known receiver range (see Section 5.5.1).

6.5 Multiple Radio Cells

It is clear that the systems described previously, which use a single ZM, cannot be extended to cover an arbitrarily large space, because the ZM's radio transmitter would then be required to have an arbitrarily large range.

In practice, regulatory concerns would limit its power output at a particular frequency, and hence would limit its range. Instead, a system with a number of independent ZMs, each of which has its own radio transmitter and is controlled by its own scheduler, must be considered.

The area around a ZM's radio transmitter in which its transmissions can reliably be detected is called a *cell*. ZMs are placed in the environment in such a way that any point in the arbitrarily large space discussed above is covered by at least one cell. This allows a very large area to be covered using multiple ZMs. Furthermore, the aggregate rate at which several ZMs can send out addressing messages is much greater than the rate at which a single ZM can send out messages, suggesting that a combination of small cells could have a higher tracking capacity than a single large cell.

This section describes the principles behind a scalable tracking system that uses multiple radio cells to trigger MPDs across a very large area.

6.5.1 Multiplexing schemes

Radio cells must be orthogonal in some sense, so that transmissions in each cell do not disrupt transmissions in neighbouring cells. This might be achieved in a number of ways, including time-division multiplexing (TDM), frequency-division multiplexing (FDM) and code-division multiple-access (CDMA). Each cell has a *colour*, which is an identifier used to distinguish between orthogonal channels in the multiplexing scheme—for example, an FDM system might use four frequencies, and each cell would have a colour indicating which of the four frequencies it uses. When colours are assigned to cells, orthogonality is arranged by ensuring that no two cells with the same colour overlap at any point in space. Each cell also has a Cell ID, transmitted with addressing messages, which allows MPDs to differentiate cells of the same colour.

The process of cell colouring in indoor radio networks is highly problematic [Steele92]. The propagation of radio waves through buildings can be unpredictable, giving rise to cells whose extents can only be determined reliably by direct measurement. Furthermore, the cell extents can change as doors within the building are opened and closed. For these reasons, it is likely that many cells will be required (to ensure full coverage of the building if cell extents decrease), and that opportunities to reuse colours will be limited (so that cells of the same colour do not overlap if cell extents increase).

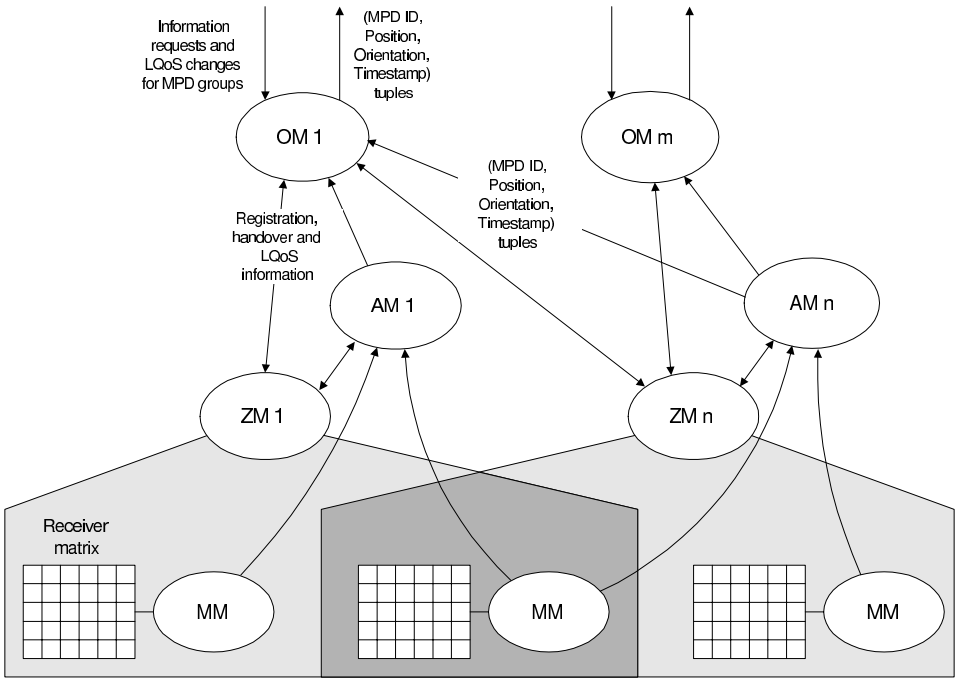


Figure 6.8: Scalable system overview

Consequently, many colours will be required to configure the radio network. Ideally, the choice of multiplexing scheme should simultaneously reflect both the demand for a large number of colours and the requirements that MPDs must be small, low-powered and relatively cheap.

6.5.2 Scalable system organisation

Each ZM in the scalable system has an associated AM, as shown in Figure 6.8. Every AM is connected to the set of MMs that control receiver matrices in rooms which (partially or completely) lie within range of their respective ZM, and receive from those MMs messages containing MPD locations. ZMs inform their AM of the scheduling decision they make in each timeslot, and AMs inform their ZM when MMs determine the location of MPDs within range of that ZM (see Section 6.3.2).

ZMs are coordinated by one or more *Object Managers*, or OMs. Each MPD group is associated with one, and only one OM. The OMs act as a point of contact for users and applications, storing the desired LQoS for MPD groups. During the period in which a ZM manages a particular MPD,

both the ZM and its associated AM are in contact with the appropriate OM. The OM informs the ZM of the desired LQoS for that MPD, and of changes to that LQoS. The AM passes MPD locations to the OM, which can process them and distribute them to users and applications. An entity location service allows ZMs to be found using their associated Cell IDs, and allows OMs to be found using the IDs of associated MPD groups, or members of those groups.

6.5.3 Scalable system operation

This dissertation considers a mode of operation of the scalable system in which cells of only one particular colour may be active in a particular timeslot. In this case, the system may be regarded in each timeslot as a number of smaller, independent tracking systems, as shown in Figure 6.9. The active cells do not overlap, so at most one MPD can transmit a positioning pulse in the rooms covered by an active cell[3]. The task of the AM associated with each active cell, to match a position and orientation calculated by a MM to the MPD scheduled for that cell, is then quite simple. The methods for near-simultaneous triggering of several MPDs within a cell (Section 6.4.2) and handling of large rooms (Section 6.4.3) may also be applied directly. However, if a large number of colours are required, the duty cycle of individual cells will be low, and therefore the maximum rate at which MPDs can be addressed may be limited.

Higher tracking rates might be possible if cells of many colours were active in each timeslot, so that MPDs handled by cells of different colours were triggered simultaneously. In each timeslot, however, several MPDs could then be triggered in rooms covered by one particular cell, and the AM's task of determining which set of ultrasonic signals was generated by which MPD would be more complex. Detailed knowledge of cell extents and staggered triggering of MPDs in different cells might allow these ambiguities to be resolved, but such schemes cannot be examined in more detail in this dissertation.

[3]To ensure that this property holds, an additional constraint, that two cells of the same colour may not cover parts of the same ultrasonic space, is applied to the colouring scheme.

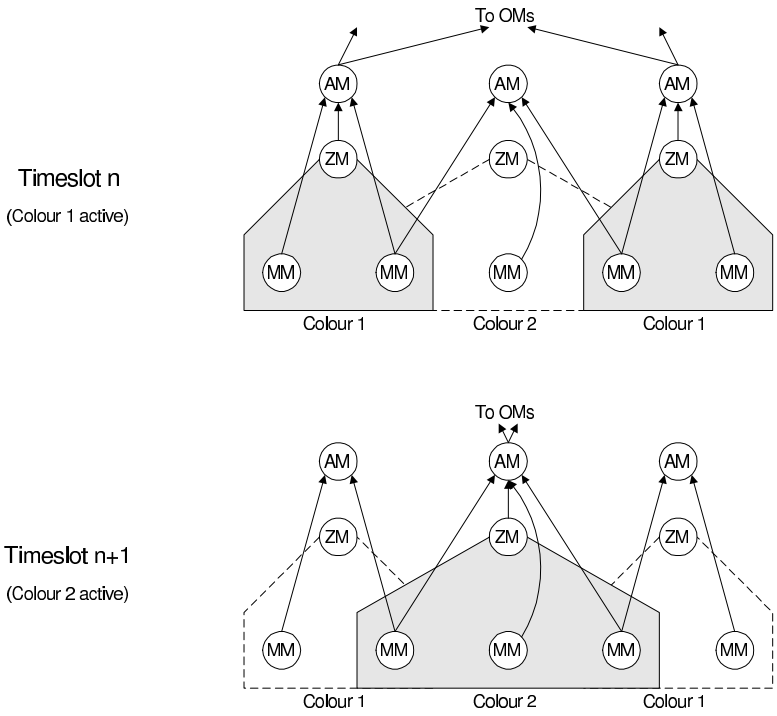


Figure 6.9: System in which only one cell colour is active

6.5.4 Handovers between radio cells

In some situations, an MPD may move from one radio cell to another. When this happens, the MPD must perform a *handover*, so that control of the MPD passes from one ZM and its scheduler to another, and so that the MPD begins to receive messages via another cell. The handover between cells may be active or passive.

Active handover

In an active handover, the ZM, OM, MPD or other monitoring service decides that a transfer of control is appropriate. The transfer may be initiated because of poor radio link quality, the physical position of the MPD, or a desire to share load between cells. Knowledge of cell extents gathered during the configuration of the cellular radio network may be used to help select the cell to which the MPD will be directed. Hysteresis should be incorporated into the handover decision criteria, so that oscillation between cells does not occur.

This dissertation considers an active handover mechanism that is coordinated by the OM associated with the MPD that is to hand over, as shown in Figure 6.10. When a requirement to transfer control is established, the OM selects a new cell to which the MPD should hand over, and presents that information to the ZM which is currently addressing the MPD. In turn, the current ZM indicates to the MPD that a handover should take place, transmitting the colour and Cell ID of the new cell in an extended addressing message. At the same time, the OM sends a message to the new, provisional ZM, instructing it to start addressing the incoming MPD at the appropriate rate. Both ZMs then address the MPD until the provisional ZM determines that the incoming MPD has picked up its addressing messages. The handover is then complete, and a message is sent back to the original ZM (via the OM), instructing it to stop addressing the MPD[4]. If the handover requirement changes during this time, the provisional ZM is informed (and stops addressing the MPD), and, if a handover is still desirable, a new provisional ZM is assigned. Provisional ZMs do not attempt to recover location resources in the manner described in Section 6.3.2 if no response is seen from

[4]When ZMs stop addressing an MPD, they set the Drop bit of the last addressing message sent to that MPD, to ensure that the MPD is aware of the change—see Section 6.3.2

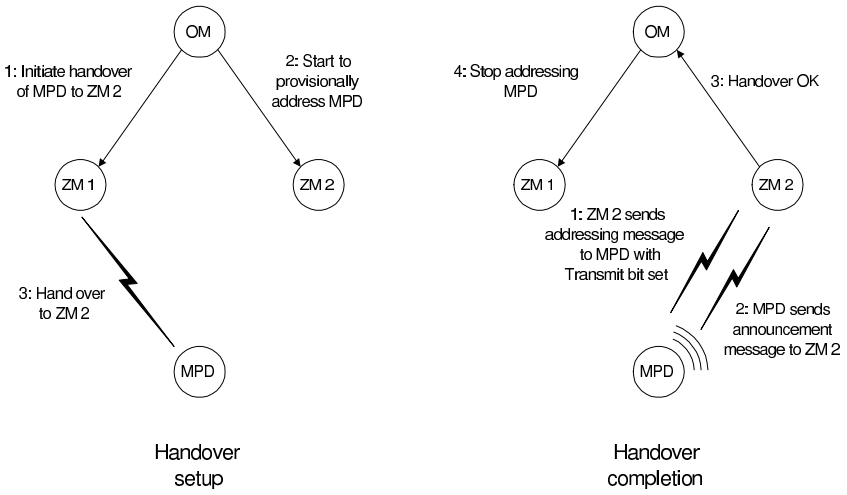


Figure 6.10: Active handover

an MPD.

To complete an active handover, the provisional ZM must be satisfied that the MPD is receiving its messages. For this reason, MPDs perform the registration process when they lock on to a new cell, and the Transmit ID bit is set in addressing messages destined for the MPD that is handing over. When the MPD receives such a message, it immediately transmits its address to the ZM; this process is known as *triggered registration*. There are therefore three ways in which the MPD can indicate to the provisional ZM that its messages are being received; standard registration, triggered registration, and transmission of a positioning pulse in response to an addressing message for that MPD.

Passive handover

A passive handover, shown in Figure 6.11, occurs when an MPD determines that it has started to receive messages from a new ZM. Examples of situations in which this might occur are:

1. An MPD moves out of a cell quickly, and there is no time to perform an active handover. Instead, the MPD fails to receive an addressing message when one was expected, starts to search for another cell (looking on all possible multiplexing channels), and finds one.

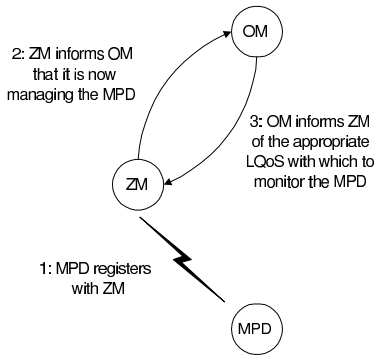


Figure 6.11: Passive handover

2. An MPD moves quickly from one cell to another that uses the same multiplexing channel. In this case, the MPD does not miss an addressing message, but detects that it has moved into a new cell, because the Cell ID in the incoming messages has changed.

3. An MPD that has failed to receive an addressing message and is in the searching state (as described in Section 5.2.3) wakes up, searches to find a radio cell, and finds one.

During the passive handover, the MPD will register with the ZM associated with the new cell, and the ZM indicates to the OM that it is now addressing that MPD. The OM then sends a message to any ZM that was previously addressing the MPD, instructing it to stop doing so. At this point, the passive handover is complete.

6.5.5 Group handovers

Section 6.2.2 describes a mechanism by which a ZM addresses MPDs in the same group consecutively to increase the accuracy of orientation measurements. To simplify the coordination of schedulers, it is proposed that all members of an MPD group should be controlled by the same ZM, whenever possible. Otherwise, the schedulers of the ZMs concerned would be required to communicate frequently to ensure that the MPDs were addressed together.

A *leader* is selected from each MPD group, and when the leader hands over to new cell, the rest of the group follows suite. This convergence is orchestrated by the OM, which is always aware of the cellular locations

of the group members and can give instructions to the appropriate ZMs. Typically, when the leader hands over to a new cell, the other members of the group will also be able to hand over, because the MPDs are attached to a single object, and will thus be in close proximity. The same procedure can be used at start-of-day to arrange that members of a group are controlled by a single ZM. During the period in which the group members are handing over to the same cell, it is unlikely that they will be addressed consecutively. Applications using orientation information calculated from their positions should therefore be made aware that this data may not be accurate.

6.6 Summary

This chapter has presented methods by which the relatively inflexible tracking system of Chapters 4 and 5 can be extended into a scalable location system. The system can properly share location resources between objects based on a variable priority value associated with each MPD, and this resource allocation information may also be used to aid MPD power management. By detecting when MPDs enter and leave the tracking space, the location system can monitor a changing set of objects, and a multi-cellular radio architecture enables operation of the system over very large areas. The capacity of the basic location system may be increased by near-simultaneous triggering of MPDs in distinct ultrasonic spaces, and an adaptation of this technique to allow efficient operation of the system in open-plan environments has been described.

Chapter 7

Software Architecture

7.1 Introduction

The location system described in the previous chapters is able to accurately determine the positions and orientations of objects in indoor environments. Once that information has been collected, it must be managed in such a way that location-aware applications can use it effectively. Furthermore, it would be advantageous to hide details of the underlying location technology from applications, to minimise the effects of future changes in that technology.

This chapter discusses a new software architecture, suitable for use in a single administrative domain, in which applications are presented with a view of the environment that is abstracted away from the ultrasonic location system. Methods that can be used to answer the types of queries which might be made frequently by location-aware applications are also described. Certain issues, such as the control of access to sensitive location information and interoperation of location systems in different administrative domains, are beyond the scope of this dissertation, and have been excluded from consideration.

7.2 Architecture

The proposed software architecture is shown in Figure 7.1. Each physical object in the environment is represented by a software object. Software objects act as a translation mechanism, converting data specific to the location system (such as MPD positions) into forms suitable for use by location-aware

applications (such as physical object positions). Since the concepts of physical and software objects are strongly linked, this chapter will use the term *object* to describe both together.

Location-aware applications determine the state of the environment by communicating with sets of objects. For example, a call-forwarding application might receive information from all telephone and personnel objects in a domain. Some applications may act as *services*, performing particular, specialised types of query or filtering on behalf of other applications. Others may act as *compound object services*, providing higher-level applications with a single point of contact from which they can receive up-to-date state information about a set of objects (e.g. all telephone objects). Compound object services should not only forward state updates for particular object sets to applications, but should also monitor and handle changes to those sets. The use of compound object services therefore reduces the complexity of applications such as the call-forwarding system by minimising the number of connections that they must maintain.

7.2.1 Object transformations

The transformation required to convert MPD positions into an object state will differ for each type of object considered. In principle, a general approach to managing the large range of objects to be tracked might be taken. This would involve describing each object type in a modelling language that expressed the object's freedom of motion and relationship with its MPDs in a standard way, possibly using hierarchical decomposition of complex objects [Foley90]. A single software object implementation could then parse the relevant model in each case, using it to determine the necessary MPD-object translation. However, the description language required by this approach would be very complex, as would be the object implementation's task of processing it.

Instead, it is proposed that each object type is modelled using a different software implementation, capable of interpreting the MPD sightings *for that particular type*. The techniques outlined in Section 4.4.1 are used by each object to derive their position and orientation from MPD locations, taking into account the known freedom of motion and conventions of use of the object. Similar objects, such as monitors with different screen sizes, may be represented by software objects with the same implementation but

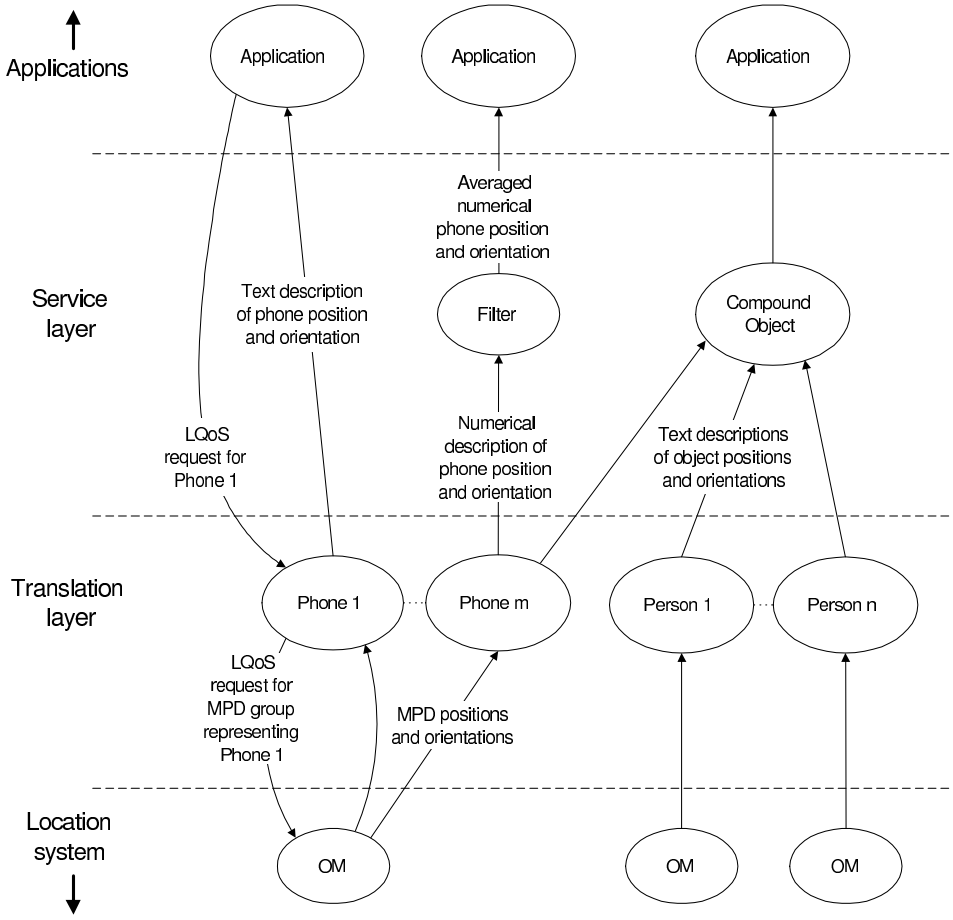


Figure 7.1: Object model overview

with different configuration parameters. The object position and orientation assume the timestamp of the last MPD reading used in their calculation.

7.2.2 Object interfaces

Once an object has determined its position and orientation, that information must be made available to location-aware applications. It is likely that the difficulties of dealing with multiple object types (discussed in the previous section in the context of handling MPD positions) will also affect applications attempting to interpret object states. It is therefore suggested that presentation functions are also performed on a per-object basis, wherever possible.

Objects have a number of interfaces, one for each presentation function that they perform. For example, objects might have a "Text" interface, via which applications can receive textual descriptions of their position, such as "Monitor X: First floor, main lab" or "Room Y: Door open". A resource-tracking application could then be implemented simply by collating descriptions from all objects, without requiring knowledge of how to interpret positions and orientations for each object type.

In certain circumstances, applications must perform their own interpretation of object state—for example, an application may wish to display information that is not provided by a standard interface, such as "Room Y: Door open 30°". Each object should therefore have an additional interface, across which applications can receive a per-object data stream that fully describes the object's knowledge of its state. Clearly, this approach requires that those applications are able to decode the data stream from each object. The availability of a per-object data stream also allows new presentation functions to be prototyped in test applications before they are incorporated into the object itself.

From an application's point of view, updates to the state of objects may occur at any time. It would be undesirable to require that all client applications polled objects frequently to determine when their state had changed. Instead, a method by which objects can asynchronously notify applications of such changes is required. Client applications can *register* interest in the object information provided by a particular interface. Until the application *deregisters* with that interface, it will receive *callbacks* when new information is available. Each callback will contain the latest information, thus keeping

the client supplied with the most recent data. When a callback is made to an application, the object state's timestamp is also passed to the client, allowing that application to determine the extent to which the reported object state is current.

7.2.3 Controlling location rates

Each object has an interface over which applications can request that it be located with a particular LQoS, or indicate that a prior LQoS request should be ignored. The object maintains a list of the requested LQoSs, and ensures that its OM always knows the highest current LQoS request.

No inter-object arbitration is performed on the LQoS requests made by applications, and objects take no account of the current load placed on the location system when passing the highest LQoS request to their OM. Clearly, a simple system of this nature is vulnerable to rogue applications that make excessive LQoS requests, wasting resources that should be assigned to genuine requests from other applications.

For this reason, it is suggested that more sophisticated system implementations would utilise a per-domain rate arbitration service, to which applications would pass LQoS requests. The arbitration service would forward appropriately scaled LQoS requests to objects, with the scaling factor being determined by the number of applications interested in each object, their requested rates, and the location system load. The design of such a service is beyond the scope of this dissertation.

7.2.4 Resource information

A number of items of resource information, such as the object's physical properties and the relative positions of MPDs mounted on it, are required to fully determine the object's state from the data provided by the location system. This resource information must be managed, so that it can be readily altered, accessed, and backed up when necessary. It is therefore suggested that some part of the software object should be *persistent*, i.e. stored in a permanent manner using a external database management system (DBMS), or similar system, to avoid duplication of data management functionality in objects.

The same database may be used to store information about the types of

objects, so that applications can determine the set of objects with which they are concerned. An external naming service then allows the applications to contact those objects. Applications that contact sets of objects in this way will require notification of updates to the database, to ensure that the sets are current. Database updates will be relatively infrequent, and applications may therefore receive them via either a polling or callback mechanism. The use of compound object services, outlined in Section 7.2, frees applications from the task of monitoring database updates, because those services will transparently handle any relevant changes to the resource database.

7.2.5 Sensor fusion

The term *sensor fusion* is used to describe the correlation of information from several different sensor sources to determine the state of an object or environment. Sensor fusion is a difficult task, because the range of sensor types that might supply data is large, and different sensors may supply contradictory information.

No general technique can reconcile sensor data from diverse sources in all situations. If the domain of the sensor fusion problem is limited, however, knowledge of the characteristics of the domain and the sensors can be used to develop a suitable correlation mechanism. Such methods are easily integrated into the per-object-type approaches to handling MPD data described above, which already use different algorithms to deal with location and orientation information in each case. It is proposed that objects may have several interfaces across which they can receive sensor data, the interface with the OM being only one. The implementation for each object type must combine the sensor data in some meaningful way, but need only do so for that particular type.

Consider, for example, a telephone with three MPDs on the base unit, and three MPDs on the handset. A *Computer Telephony Integration* (CTI) service [Walters97] also provides information to the telephone object about the up/down state of the handset. The telephone object regards the CTI service as the most accurate sensor regarding handset up/down events, because when the handset is down, random errors in the MPD positions might suggest that the handset is a few centimetres above the base unit. Therefore, if the CTI service indicates that the handset is up, the telephone object uses the three MPDs on the handset and base unit to determine the position and

orientation of each component. If the CTI service indicates that the handset is down, the telephone object uses all six MPD locations to determine an accurate position and orientation for the base unit/handset pair. This example represents a simple priority-based form of sensor fusion, in which information from one sensor overrides that from another, less reliable sensor.

7.3 Spatial Indexing

The behaviour of many location-aware applications will be determined by the spatial relationships between objects, such as "X is in sight of A", "X is next to B" or "X can touch C". It would therefore seem appropriate to investigate ways in which queries about spatial relationships can be managed.

One approach to this problem is to examine interactions between regions around objects that are generated using their positions and orientations. For example, if a region corresponding to the points that a person could touch was to intersect a region corresponding to the volume of a mouse, an application could conclude that the person could use the mouse. Naïve calculation of all such possible interactions every time an object (and its related regions) moved would clearly be expensive. However, *spatial indexing* techniques, which sort data with respect to the space it occupies, can be used to prune the set of possible interactions to those in the area surrounding the moving object.

Spatial indexing is a standard procedure in spatial DBMSs, which find use in applications such as environmental modelling, resource management and Geographic Information Systems (GISs) [Güting94]. Spatial data is typically held by those systems in a persistent store—queries are fast, and the power of the database query language can be fully exploited, but the cost of placing the data in the store may be high. This model is not appropriate for the management of the real-time information provided by the indoor location system, because location-aware applications are often unconcerned with persistent storage of the current environmental state, and require low latency. Instead, a dynamic system that provides a spatial management service for applications, and that can make spatial relationship queries on a snapshot of current object positions and orientations must be considered.

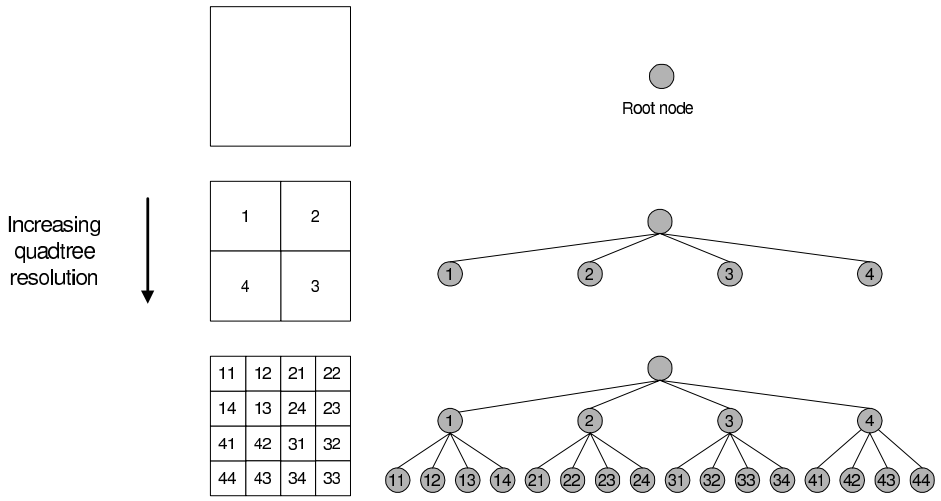


Figure 7.2: Quadtree data structure

7.3.1 Principles

Steggles [Steggles96] has developed a suitable dynamic spatial indexing system, which is described below. Areas around objects that are associated with spatial interactions are modelled as 2D shapes. Each shape has a type, which identifies the meaning of that shape (e.g. "In front of the monitor") and can be used by applications as a key to create spatial queries (e.g. "Tell me when an area around a person overlaps the area in front of a monitor"). Shapes also have a unique identifier; mappings may link the shape to a particular object using this identifier, or indicate that the shape represents some arbitrary area (e.g. "First floor landing").

Shapes are managed by a system called the *containment index*, which allows spaces to be inserted, moved or deleted in a quadtree-based store held in memory. The quadtree is generated by breaking down the plane containing the shapes into sub-quadrants, where each quadrant cell is represented by a node in a tree [Foley90]. The root node represents the entire indexed region of the plane, and the four sub-quadrant nodes are children of the root node. In turn, each of those four cells and nodes has four sub-quadrants and children respectively, and so on, to form the hierarchical quadtree, as shown in Figure 7.2.

When a shape is placed into the containment index, the *maximal cover* of the space is calculated. The maximal cover is the smallest set of quadtree

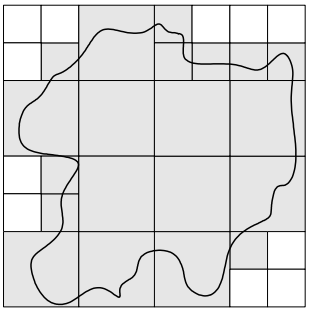


Figure 7.3: Maximal cover of a shape

cells required to cover the space at a particular *resolution*. The resolution dictates the minimum size of quadtree cell that is considered during the indexing operation—higher indexing resolutions consider smaller quadtree cells, and so result in a better approximation to the shape in the index. An example maximal cover is shown in Figure 7.3. Maximal covers are stored by labelling the quadtree nodes corresponding to the cells in the maximal cover of each space with that space's identifier. The cost of placing a shape in the index depends on its size (the cost is roughly proportional to the space's perimeter), the indexing resolution, and the type of space—simple spaces such as circles and rectangles are easy to index, concave polygons are more costly to index.

The maximal covers stored in the containment index allow cheap identification of containment and overlapping relationships between spaces. By traversing the quadtree structure and counting the number of nodes labelled with the identifiers of particular spaces, the following theorems can be used to determine those associations:

- **The containment indexing theorem:** A space s is contained is a space t iff, for each cell x in the maximal cover of s, there exists exactly one cell in the maximal cover of t that is an *ancestor* of x (i.e. it contains x) or is equal to x.

- **The overlap indexing theorem:** A space s overlaps a space t iff there exist two cells x and y in the maximal covers of s and t respectively such that x is equal to y, x is an ancestor of y, or y is an ancestor of x.

When an update is made to the containment index, the relationships between

the space undergoing the change and other spaces stored in the index are re-evaluated. Four kinds of events are then generated:

1. Positive containment events are generated for all pairs of spaces s and s' where s now contains s', but did not before the update.

2. Negative containment events are generated for all pairs of spaces s and s' where s no longer contains s', but did before the update.

3. Positive overlapping events are generated for all pairs of spaces s and s' where s and s' now overlap, but did not before the update.

4. Negative overlapping events are generated for all pairs of spaces s and s' where s and s' no longer overlap, but did before the update.

Using the callback paradigm discussed in Section 7.2.2, client applications can register interest in particular types of events between specified space types. When such an event occurs, a callback containing a description of the events is made to the client application.

7.3.2 Integration

Figure 7.4 shows the integration of the spatial indexing system with the object model described in Section 7.2. A 2D spatial index is created for each floor of the building under consideration, and a naming service (not shown) allows objects to find the spatial index managing the floor on which they are located. Objects have an interface across which they may register one or more spaces with the relevant spatial index. The extents of the spaces are calculated by each object using its position, its orientation and knowledge of any regions around it that might be relevant to spatial queries. For example, a monitor object might register two spaces—one representing the region where information on the screen is visible, and the other representing the centre of the monitor, perhaps for inventory purposes. When spaces are indexed by objects, they are tagged with appropriate types, identities and timestamps. Objects might use sensor information other than positions and orientations to refine the spaces to be indexed. If a sensor on the monitor indicated that it was switched off, the "screen visible" space of the monitor might be removed from the spatial index. Spaces that are not associated with any physical object may be placed directly into the index by applications.

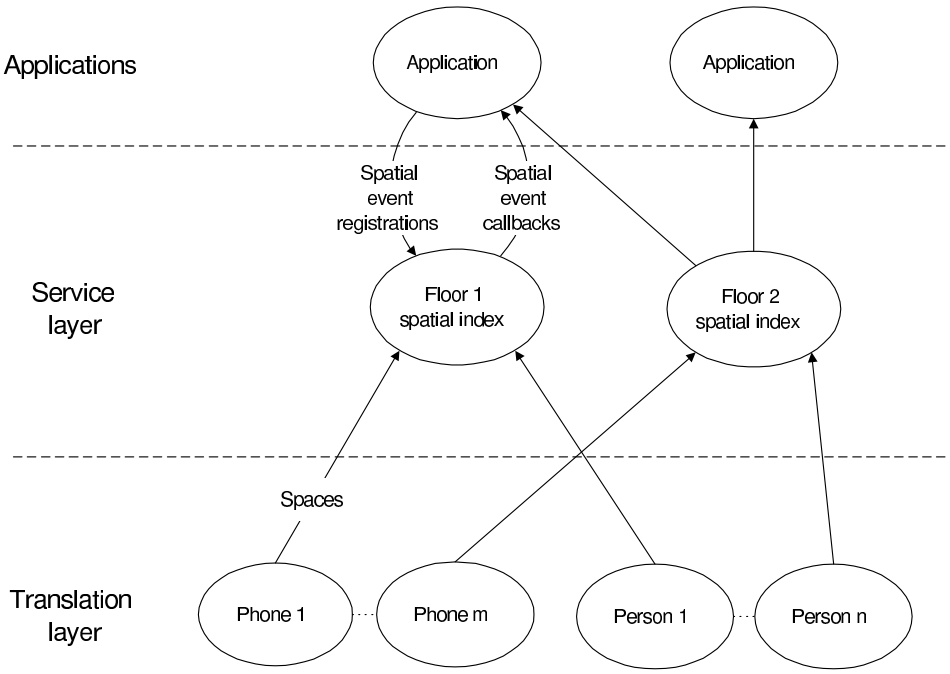


Figure 7.4: Integration of the spatial index with the object model

When an object moves, it recalculates and reindexes its spaces. Clearly, a certain amount of responsibility must be exercised by objects when entering spaces into the index. Objects whose locations will be updated frequently (e.g. people) must only register small, simple spaces at relatively low resolution, so as to minimise the indexing work required.

Client applications may register interest in particular spatial events with an index or several indices. When a spatial index determines that an event matching a request has taken place, the appropriate client applications receive a callback containing the type of event and the identities of the spaces involved in it. If the spaces are associated with objects, their identities can then be found using the space identifiers. The timestamps of the interacting spaces are also passed to clients, to enable them to synchronise spatial events with events obtained directly from objects or from other monitoring services. For example, an application might wish to present information on a monitor only when the intended recipient is in the field of view of a monitor, and is facing it—orientation data would, in this case, be obtained from the recipient's object. However, the event streams might suffer variable delays, requiring buffering of events and reordering using their timestamps

to achieve a consistent global view. Hayton has considered the problems of handling events in distributed systems, and presents a grammar that allows composite events to be specified and subsequently evaluated [Hayton96].

7.3.3 Approximate spatial relationships

Although only approximations to spaces around objects are stored in the containment index, client applications may determine the spatial relationship between objects with high accuracy, but at relatively low cost, by using callbacks from the spatial index as hints. Only when an "approximate" event is received from the spatial index should applications perform the necessary exact geometrical calculations using the positions and orientations of the interacting objects. This technique minimises the number of expensive computations that must be made. In many cases, however, "approximate" events will be sufficiently accurate to allow applications to change their behaviour without need for further calculations.

A similar approach allows applications to monitor 3D volumes using a 2D spatial index. 2D projections of the volumes are indexed, and interactions between them trigger further geometric calculations in client applications to determine the true 3D relationship between the objects. An alternative, a 3D spatial index based around an octree data structure [Foley90], could be envisaged, with the cost of indexing a volume being roughly proportional to its surface area. However, it is unclear whether the ability to make direct 3D queries would justify the increased complexity of such a system, given that objects move most often along the floors of buildings.

7.3.4 Containment queries

Spatial indexing may be used to answer containment queries in an efficient manner. As described in Section 2.2.2, domains are typically organised into a hierarchy of relatively static containers, such as buildings and floors. Areas corresponding to the shape and position of each container are placed in the spatial index. If small regions corresponding to each object under consideration are then entered into the index, containment events will be generated in response. Those events can be processed by applications or services to maintain the containment state of each object, allowing containment queries to be answered.

7.3.5 Related work

Nelson has described an alternative spatial indexing system for location-aware computing [Nelson98]. He proposes a 3D index based on an *R-tree* data structure [Güting94], with objects modelled as simple cuboids that are aligned with the coordinate axes. The author describes the use of this index to generate events when an object enters or leaves a spatial area, and shows how simple orientation information (indicating which faces of the cuboid object are "active" in some sense, e.g. the screen face of a monitor) can be used to prune the number of events generated in response to queries. However, spatial regions with non-cuboid volumes cannot be modelled by the system, so it is not possible to use the index as an efficient mechanism for identifying interactions between regions around objects using "approximate" events. Furthermore, the detailed orientation information available from the ultrasonic tracking system would not be used effectively by a spatial index of this kind.

7.4 Proximity Queries

Applications may wish to make proximity queries regarding objects in the environment. Examples of this type of query might be "Which telephones are within 10m of Person X?", or "Which is the closest printer to me?". Proximity queries might be evaluated by services offering specific query interfaces, or, more generally, by applications themselves.

This section considers the problems associated with the evaluation of two types of proximity query. The first, the *bounded* proximity query, allows applications to find all objects of particular types (the *targets*) within some distance (the *extent*) of a point in space (the *source*). The second, the *unbounded* proximity query, returns, if possible, a specified number of objects of particular types that are the nearest such targets to some source point. All proximity queries studied in this section treat objects as points, and are based upon the Euclidean distance metric.

7.4.1 Computing shortest paths

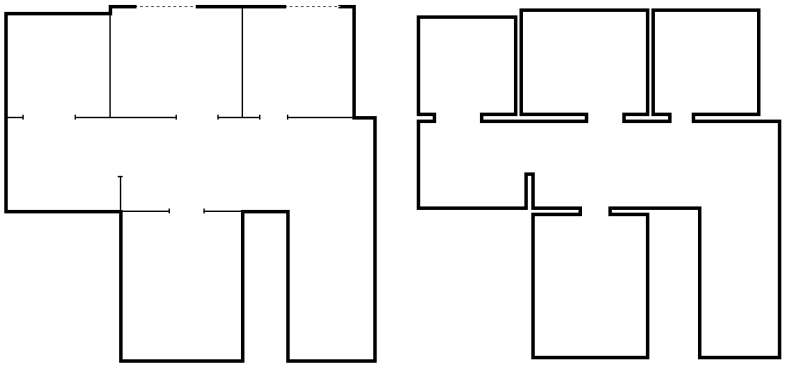
To evaluate proximity queries, distances between points in the environment must be determined. The calculation of direct path lengths between points

in two and three dimensions is trivial. However, proximity queries may take account of the topology of the space in which objects are located, and there may then be more than one possible path between points in the environment. Since both the bounded and unbounded query types determine, in some sense, the set of objects *within* some distance of another, only the shortest path between objects need be considered when evaluating those queries. The task of finding the shortest path between two points is well known in the field of computational geometry, and Mitchell presents a good overview of the state-of-the-art in this area [Mitchell98].

It is likely that many proximity queries made by applications will be anthropocentric, i.e. they will determine the distance that personnel will have to move to reach objects or resources. Only feasible routes, avoiding walls, desks and other obstacles will be considered appropriate by such queries. Finding the shortest feasible path between two points on a floor of a building modelled as the interior of a polygon (see Figures 7.5(a) and (b)) is computationally tractable, having complexity $O(n)$, where n is the number of vertices of the polygon. The related case in which the model contains a number of disconnected polygonal obstructions, perhaps representing pillars and the like, is also tractable, having complexity $O(n \log n)$, where n is the total number of polygon vertices. The corresponding general problems in buildings modelled as three-dimensional spaces are known to be NP-hard. It is therefore suggested that for the purposes of proximity queries, buildings are represented as discrete two-dimensional floors, with appropriately weighted links joining floors at stairwells, as in Figure 7.6. Distances between points on different floors may be calculated by adding the distances to the link points on each floor to the weight of the link.

7.4.2 Query implementation

The bounded and unbounded proximity queries may be evaluated in a naïve way by determining the set of all known objects that match the search criteria, finding their distances from the source by the methods outlined above, and sorting that list to obtain suitable query results. Clearly, this approach will be inefficient if the set of objects matching the search criteria is large. It is therefore suggested that the spatial indexing techniques discussed in Section 7.3 are used to optimise the queries.



(a) Plan view (b) Floor modelled as a polygon

Figure 7.5: Modelling one floor of a building as the interior of a polygon

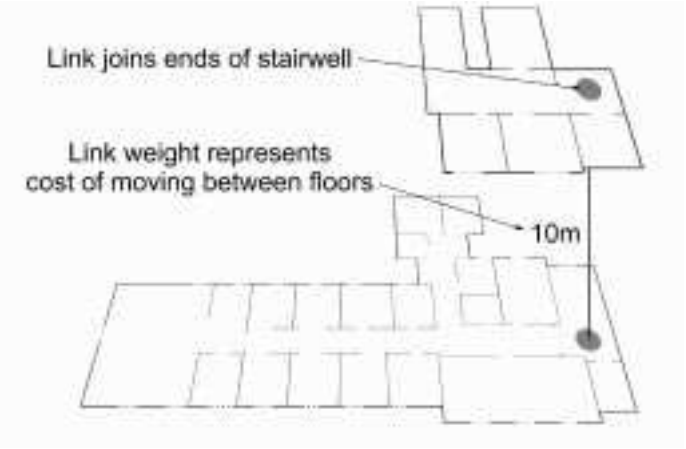


Figure 7.6: Modelling a building as a set of 2D floors

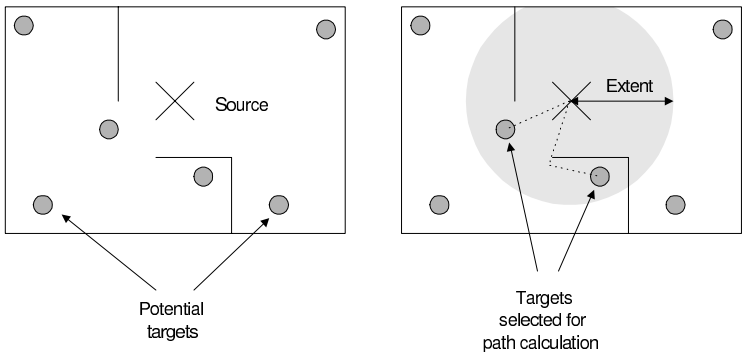


Figure 7.7: Simplifying bounded proximity queries using spatial indexing

Bounded proximity query

First, it is noted that the distance between points in an environment in which movement may be restricted is greater than or equal to the corresponding straight-line distance. Therefore, a circle around the source with radius equal to the extent will contain all points that may be reached along a path with length less than or equal to the extent.

A query processing strategy called *filter-and-refine* [Esperança97] is then used to reduce the cost of the evaluation task. Suppose a circular *test area*, with radius equal to the extent, is placed in the spatial index around the source. The representation of the area in the index is conservative, in that the maximal cover contains all of that area. Suppose also that all potential targets are routinely included in the spatial index, by entering a small space for each as their positions are found by the location system. It can be seen that all potential targets that may be reached by a path with length less than or equal to the extent will overlap the circular area to some degree, as in Figure 7.7, and the spatial index will generate overlapping events identifying them. Therefore, exact path length calculations need only be made for that subset of the potential targets. Since it is likely that the cardinality of the subset will be much smaller than the total number of potential targets, the query evaluation is simplified.

When evaluating proximity queries regarding objects in buildings with several floors, a modification to the above algorithm is required. If the test area overlaps a link point on one floor, a second circular test area must be placed in the index of the adjoining floor. The radius of the second area is

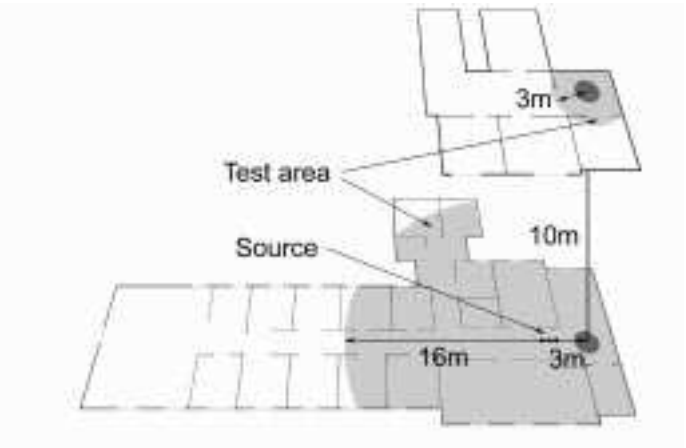


Figure 7.8: Propagating a proximity query between floors

equal to the extent less the weight of the inter-floor link and the distance from the source to the link point on the first floor. This process is shown in Figure 7.8.

Unbounded proximity query

Evaluation of unbounded proximity queries may also be simplified using the spatial index, as shown in Figure 7.9. A small circular test area (whose exact size may be dependent on the estimated density of potential targets) is indexed around the source, and distances are calculated to all potential targets falling within that circle. The test area is repeatedly grown (and distances from the source to newly-found targets calculated) until the following criteria are met:

1. At least the desired number of targets, with distances from the source less than the extent, lie within the test area.

2. The radius of the test area is greater than or equal to the distance of the n^{th} target from the source, where n is the desired number of targets.

If all potential targets lie within the test area, but the above conditions are not met, the query cannot be fully satisfied.

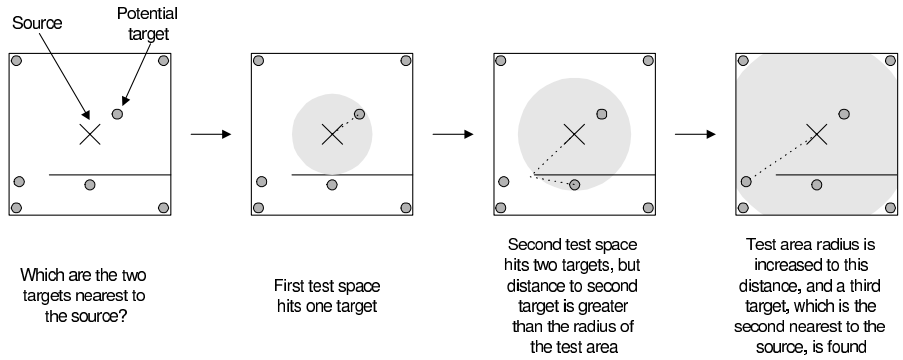


Figure 7.9: Simplifying unbounded proximity queries using spatial indexing

7.4.3 Continuous queries

It is likely that proximity queries will be made by continuous monitoring applications. In such cases, the algorithms presented above for "one-shot" queries are unsatisfactory, because the algorithms must be evaluated whenever relevant objects are seen to move, and the work done during each evaluation is subsequently discarded. It would therefore seem sensible to investigate ways in which "continuous" queries could be optimised.

Bounded proximity query

It can be seen that the distance from a source to a target is the same regardless of the direction in which it is measured. It can also be seen that when a source moves, a relatively large area must be placed in the spatial index, and distances to all potential targets in that space must be recomputed. When a potential target moves, however, only a small space need be indexed, and only its distances from relevant sources must be determined.

Suppose that lists of the type **B** objects lying within some distance of each of a set of type **A** objects are required, and that those lists should be continuously updated. By calculating the product of the population and update rate for each object type, and then selecting the object type associated with the smallest product as the source of the proximity query, the work involved in evaluating the query over time will be minimised. For example, if the rates of movement of objects of types **A** and **B** are similar, the least numerous objects should be made the sources of the proximity query, because this choice will minimise the total amount of space re-indexing re-

quired. Note that if objects of type **B** were chosen as the query sources, the results of those proximity queries would have to be collated to provide the desired answers the original problem.

Consider, then, a call-forwarding service that determines the set of telephones within a few metres of all personnel in an office. Telephones move infrequently, and the rate at which their positions are updated by the location service will be low. Personnel, however, are highly mobile, and their positions will be updated much more often. Furthermore, offices typically contain roughly equal numbers of telephones and people. Bearing in mind the above observations, the monitoring service should select the telephones as the sources of proximity queries and the personnel as the targets. From the information thus collected, which indicates which people are near each telephone, the set of telephones near each person can be found.

Further optimisations may be made by computing a *shortest path map* (SPM) from each of the (relatively stationary) sources [Mitchell98]. This process takes $O(n)$ time if the building is modelled as a simple polygon, and $O(n \log n)$ time if the building model includes polygonal obstacles. Each map may be used to determine the distances from a source to targets in $O(\log n)$ time, until that source moves, at which point the SPM must be recalculated. Lazy evaluation may be used to postpone the computation of the SPM until such time as the presence of a potential target within the extent distance of the source is identified by the spatial index.

Unbounded proximity query

Repeated unbounded proximity queries are more difficult to optimise, because movement of either the source or the potential targets may result in several indexing operations on the test area and calculation of many source-target distances. For this reason, it is suggested that continuous monitoring services do not make unbounded proximity queries. In practice, this restriction may not be serious, because many queries will be anthropocentric, and the maximum distance over which people are prepared to interact with resources is limited. The unbounded query then becomes a case of the bounded proximity query.

7.4.4 Related work

Nelson describes how an R-tree-based spatial index may be used to evaluate one-off bounded proximity queries, using a building model comprised of cuboid containers linked by junction nodes [Nelson98]. However, this modelling approach is less flexible than the geometric scheme proposed above, and neither unbounded nor continuous proximity queries were investigated.

7.5 Historical Queries

Some location-aware applications may wish to make queries about the past state of the environment, such as "Who last had the multimeter?". Other applications may wish to determine why location-aware services modified their behaviour, by tracing the set of location events that triggered those changes. Clearly, these types of query must be based upon organised, long-term storage of sensor data. Furthermore, it must be possible to efficiently perform both spatial and temporal queries on data in that store. *Spatio-temporal databases* have been investigated by GIS researchers in an effort to implement systems with these characteristics [Varma90]. However, existing spatio-temporal databases are built around DBMSs, and would not work well as a core component of the location system for the reasons set out in Section 7.3. Further investigation of problems associated with historical queries is beyond the scope of this dissertation.

7.6 Summary

This chapter has outlined a mechanism by which the position information generated by the ultrasonic location system can be converted into object state information and distributed to a range of applications. Methods of dealing with the wide range of object types that might be tracked by the location system have been discussed, as have ways in which a wide range of sensor data types might be combined on a per-object-type basis.

The chapter has also explained how three types of query that might be made by location-aware applications could be handled. Spatial relationship queries are handled by a dynamic spatial indexing system, which also supports containment queries. Proximity queries are evaluated in an efficient manner by using this spatial indexing system as a filter mechanism.

Chapter 8

System Realisation

8.1 Introduction

This chapter describes a prototype location system that is based upon the tracking methods discussed in Chapter 4, and that takes account of the scalability principles set out in Chapter 6. Existing system components, such as MPDs, ZMs, and MMs are modified and used to demonstrate registration, radio cell handover and low-power MPD operation. A software architecture, modelled on that presented in Chapter 7, has been implemented to handle the data generated by the location system.

8.2 Location System Implementation

This section describes the parameters of the location system, and the behaviour of its components.

8.2.1 System parameters

The prototype location system separates radio cells using the simple time-division multiplexing strategy described in Section 6.5.3. Timeslots are 40ms long, and the multiplexing scheme may accommodate up to 31 channels. The use of a TDM approach has the advantage of simplicity of implementation and operation, but the aggregate system capacity is limited to 25 position updates per second if TDM channels are not reused. This limitation was not thought to be important in the context of the prototype system. Each radio cell has an 8-bit Cell ID, permitting system operation with up to 255 cells.

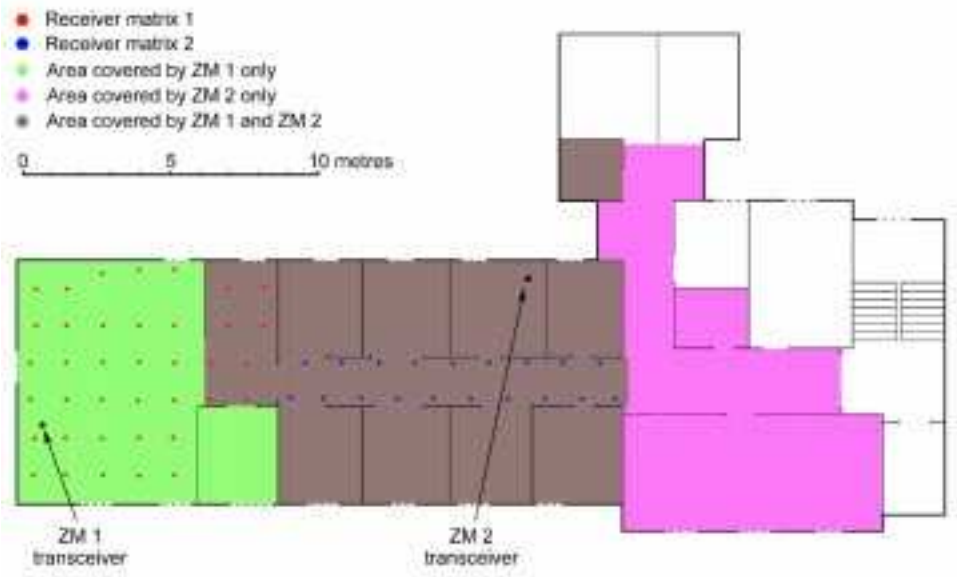


Figure 8.1: Prototype system deployment

16-bit addresses are used to identify both MPD and MPD groups. Address 0 is reserved in all address spaces.

The system has been deployed in a configuration that has two radio cells and two ultrasonic spaces (each equipped with a receiver matrix and MM), resulting in a total tracking area of some $76m^2$. The total tracking volume of the deployed system is around $215m^3$. The extents of the ultrasonic receiver matrices are shown in Figure 8.1, as are the approximate extents of the radio cells. Although the number of ZMs and receiver matrices that have been deployed is limited, both the software and hardware components of the prototype system may be used in larger deployments without change.

8.2.2 Synchronisation of system components

The ZMs and MMs are synchronised by extending the daisy-chain timing network described in Section 5.2.1 to include all those components. The timing messages generated by the GCG are also extended to include the identity of the TDM channel active in each timeslot. The modified ZMs use the channel information to ensure that they only transmit in timeslots that correspond to the colour of their associated radio cell.

8.2.3 Location system management

The prototype location system uses a single OM to orchestrate the behaviour of the ZMs. Each ZM implements its scheduling decisions using the methods outlined in Section 6.2, and has an associated AM that is connected to both MMs. TCP socket links and *CORBA* (Common Object Request Broker Architecture) [OMG91] interfaces allow the system components to communicate with each other, as detailed in Figure 6.8.

MPD handovers between the radio cells are initiated by the OM on the basis of MPD positions and measurements of the extents of radio cells. Hysteresis in handover decisions is arranged by ensuring that an MPD must move some distance into a cell before that cell becomes a candidate handover destination. The latency of requests to change the LQoS of a particular object, neglecting network delays, is 0.64s (see Section 6.2.3), and ZMs begin attempts to reclaim location resources (Section 6.3.2) after six addressing messages evoke no ultrasonic response from an MPD.

Four types of message passed between the system components govern the behaviour of the location system. The following *management control messages* are sent by ZMs to the OM, informing it of the current registration states of MPDs. The timestamps in each message protect the OM against the possibility of message re-ordering caused by variable delays on network links—stale messages are discarded upon receipt by the OM.

Registered(mid,cid,timestamp):
: Indicates that the MPD mid was registered with the ZM of cell cid at time timestamp, and has now been entered into that ZM's scheduling table.

Dropped(mid,cid,timestamp):
: Indicates that the ZM of cell cid reclaimed location resources allocated to MPD mid at time timestamp.

Rate control messages are passed to the OM by the software objects representing physical objects, and indicate the desired LQoS for the group of MPDs locating that object. *Position messages* are sent by AMs to the OM, and give the position and orientation of MPDs at particular times. These updates are first used by the OM to determine whether MPDs should hand over between radio cells, and are then forwarded to the relevant software objects.

In response to the above messages, the OM informs ZMs of the current location and handover requirements using *ZM control messages*, which have the following form:

ZM-Control(mode,mid,gid,lqos,hcid,offset):

> If lqos is non-zero, the message instructs the ZM to locate members of the MPD group gid at the rate specified by lqos. It also indicates that the MPD mid is a member of that group, and that it should be added into the scheduling table. The Mode flag indicates whether the resource allocation is provisional (i.e. whether the MPD is handing over to that ZM). If hcid is non-zero, the message indicates that the members of the MPD group gid should hand over to the cell hcid, whose offset in the TDM scheme is offset.
>
> If lqos is zero, the message indicates that the MPD mid should be removed from the scheduling table.

8.2.4 MPD hardware

The MPD design described in Section 5.2.3 is modified so that the transmitting elements of the radio transceiver may be controlled by the FPGA, allowing data to be both received from and sent to the ZMs.

The revised MPD design also includes two push buttons, and an 8-bit wide data port with strobe line, both of which are connected to the FPGA. These features take advantage of the bidirectional RF link, and give the MPD limited input/output facilities. A button-down event causes an MPD to send an announcement message in the next timeslot, and the states of the MPD push-buttons are encoded in that message. Data destined for the output port is sent by ZMs as part of the addressing messages. Buttons might be used as a ubiquitous control interface, or could be connected to simple sensors, perhaps detecting the presence of someone sitting on a chair. Data placed on the output port may be used to control objects in the environment; a robot tracked using one or more MPDs could be directed around a building by a service linked to the location system, for example.

8.2.5 Radio message formats

Two types of messages are sent across the RF link connecting the ZMs and MPDs. Addressing messages, whose format is shown in Figure 8.2, are sent from the ZMs to MPDs, and are divided into four *message segments*, each of which has a CRC, allowing error-detection to be performed independently on each segment. Addressing messages, like those described in Section 5.2.2, are transmitted using Manchester encoding, and follow a 135-bit preamble.

Announcement messages are sent from the MPDs to the ZMs, and their format is shown in Figure 8.3. Each message contains the ID and button states of the MPD that sent it, together with a CRC for that ID. Announcement messages are transmitted using Manchester encoding, and follow a 167-bit preamble. This preamble length was empirically determined to be the minimum required to enable reliable communications between MPDs and ZMs.

8.2.6 MPD operation

A simplified operation diagram for the MPD is given in Figure 8.4. In the core MPD state, every addressing message transmitted in the current cell is checked by the MPD, which enters the slumber mode in the intervening periods by switching off its receiver circuitry and fast clock. The MPD uses a counter called the *channel counter* to identify the timeslots in which the current radio cell is active. The channel counter is incremented every 40ms, and is compared against the value contained in the Colour Space field to ensure that the MPD leaves the slumber state 500μs before the next relevant addressing message is due. The channel counter is reset when the MPD leaves the slumber state.

Power-saving techniques

MPDs keep the duty-cycle of their receiver and fast clock as low as possible by receiving the minimum amount of addressing message data. For example, an MPD that has determined the Colour Space value, that has registered with the current radio cell, and whose ID does not match the Active Device field, will not receive message segments 3 and 4, because those segments will not contain relevant information.

When an MPD is in the sleep state (Section 6.2.3), it wakes up every

Description	*Size (bits)*
Wakeup bit	1
Cell ID	8
Segment 1 CRC	8
Drop bit	1
Active Device	16
Segment 2 CRC	8
Transmit ID bit	1
Handover Channel Offset	5
Handover Cell ID	8
Sleep Time	24
Segment 3 CRC	8
Colour Space	5
Data Channel Value	8
Registration Acknowledgement	16
Segment 4 CRC	8

Message segment 1: Contains the Cell ID field, which identifies the radio cell on which the message was transmitted (Section 6.5.1), and the Wakeup bit used to control low-power behaviour of MPDs (Section 6.2.3).

Message segment 2: Identifies the MPD being addressed by the ZM in the Active Device field, and contains the Drop bit, used by ZMs to reclaim location resources (Section 6.3.2).

Message segment 3: Contains commands destined for the active MPD. The Transmit ID bit is used by ZMs to force the addressed MPD to register in that timeslot (Section 6.3.2), and the Sleep Time field indicates how long an addressed MPD can remain in a low-power mode after receiving the message (Section 6.2.3). The Handover Channel Offset and Handover Cell ID fields inform the addressed MPD of the TDM channel location and identity of the cell to which it should hand over, if handover is appropriate (Section 6.5.4).

Message segment 4: Contains the Registration Acknowledgement field, used to indicate that an MPD's registration attempts have been successful (Section 6.3.1). Also contains the Colour Space field, which indicates the number of TDM channels being used in the current system configuration, and a Data Channel Value, which is placed on the data port of the addressed MPD.

Figure 8.2: Addressing message format

Description	*Size (bits)*
ID	16
CRC	8
Button 1 Status	1
Button 2 Status	1

Figure 8.3: Registration message format

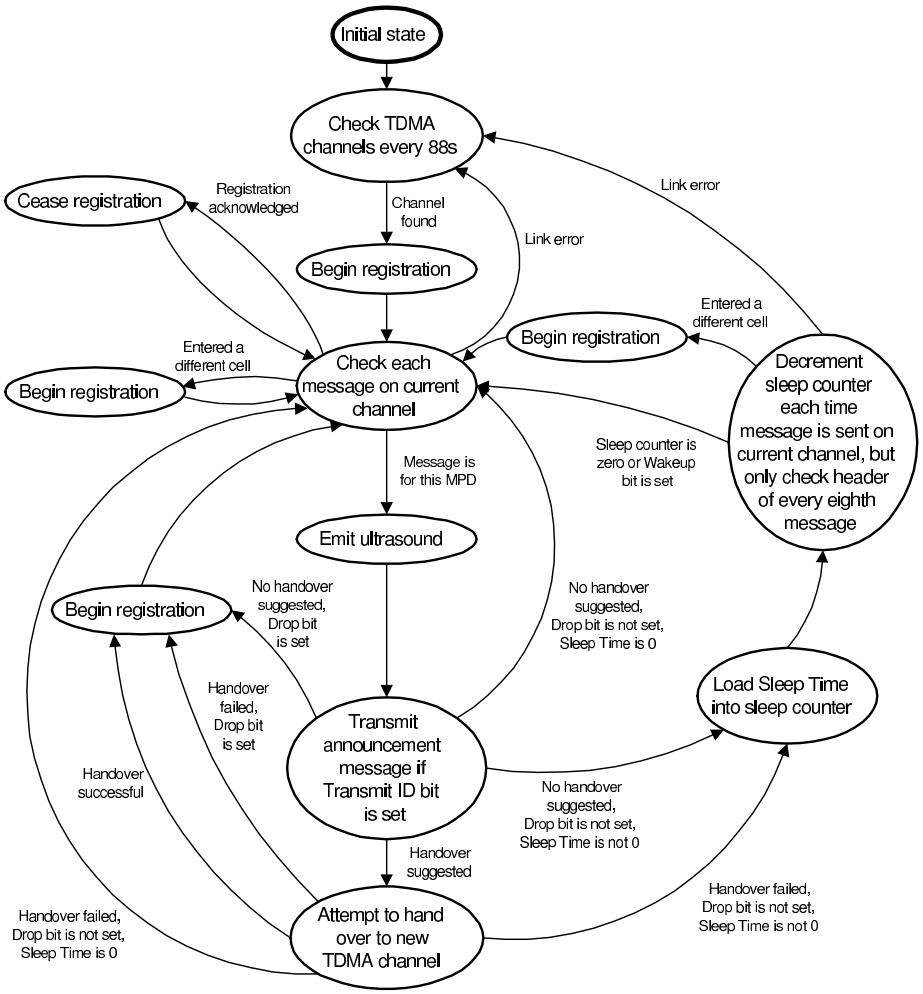


Figure 8.4: Simplified MPD operation

eight timeslots to check the Wakeup bit transmitted by the current ZM. It also checks that the Cell ID in incoming messages has not changed, to ensure that the MPD has not been moved between cells. To complete these tests, only message segment 1 must be received, minimising power usage in such circumstances.

Link errors

The MPD uses the CRC in each addressing message segment to detect link errors that have occurred during transmission of the segment. If a link error is detected, the MPD assumes that the entire message has been corrupted, and begins to search for another radio cell. The MPD leaves its receiver circuitry and fast clock active for 1.375s to ensure that it checks all possible TDM channels (a process that takes at least 31×40ms). If a valid preamble is detected during this time, the MPD takes note of the new Cell ID in the incoming message, resets its channel counter, and starts registration attempts with the new cell (see Section 6.5.4). Otherwise, it switches off its receiver and fast clock for 86.625s before attempting to find a new cell again.

Handover management

If an MPD receives a message in which the Active Device field matches its ID, and the Handover Cell ID field is not zero, it should attempt to hand over to another radio cell. The Handover Channel Offset field indicates the location in the multiplexing scheme of the channel to which the MPD should hand over, relative to its current channel. The MPD uses this value, together with the known timeslot size, to switch on its receiver circuitry and fast clock at the start of the next timeslot in which the destination cell is active.

During a handover attempt, the MPD's receiver remains active for 3.9ms. If, during this time, a valid addressing message preamble is detected, and the Cell ID field of that message matches the Handover Cell ID field of the handover-initiating command, then the destination cell has been positively identified, and the handover has succeeded. If the Cell ID does not match that field, the MPD has detected another radio cell on the same TDM channel, and does not hand over.

When a successful handover has taken place, the MPD takes note of the

new Cell ID, resets its channel counter, and starts registration attempts with the new cell.

Contention resolution during registration

Normally, the rate at which MPDs will enter the tracking environment will be low, as will be the corresponding rate at which they will try to register with ZMs. However, at start-of-day, large numbers of MPDs will register with the location system. To prevent contention between MPDs, a *slotted-ALOHA* protocol [Roberts75] is used to control their registration transmissions.

During registration, MPDs transmit announcement messages in timeslots with some probability. If two or more nearby MPDs happen to send an announcement message in the same timeslot, their transmissions will collide, and the registration attempts will fail. However, the transmission probability is decreased as the registration process goes on. Eventually, it will be so low that regardless of the number of MPDs attempting to register with the ZM, the probability of two of them transmitting in the same timeslot will be small, and the registrations will succeed.

In the prototype system, the initial transmission probability is 1/32, falling to 1/16384 after 16 transmission attempts, as shown in Figure 8.1. The transmission probability is then fixed at this value for future attempts. Simulations have been performed to determine the effectiveness of this protocol, and Figure 8.5 shows the average time (over 100 trials) required to successfully register a population of MPDs against the population size. This graph demonstrates that relatively large numbers of MPDs (in the context of the prototype system) can be registered with a ZM in a reasonable amount of time—1000 MPDs may be registered in around 20 minutes ($\approx$ 15000 timeslots, at 12.5 timeslots/second).

More efficient registration protocols could be devised. For example, suppose that an MPD which was attempting to register with a ZM, but which was not intending to transmit in a timeslot, listened for announcement messages from other MPDs in that timeslot. If several timeslots were monitored, and no announcement messages were detected, the MPD could conclude that the registration traffic was light, and could raise its transmission probability in response. This would hasten the registration process, but the MPD would use more power during the extra listening phase.

Transmission attempt	1	2	3	4	5	6	7	8
Transmission probability	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{128}$	$\frac{1}{128}$	$\frac{1}{256}$	$\frac{1}{256}$
Transmission attempt	9	10	11	12	13	14	15	16+
Transmission probability	$\frac{1}{512}$	$\frac{1}{512}$	$\frac{1}{1024}$	$\frac{1}{1024}$	$\frac{1}{2048}$	$\frac{1}{4096}$	$\frac{1}{8192}$	$\frac{1}{16384}$

Table 8.1: Registration transmission probabilities

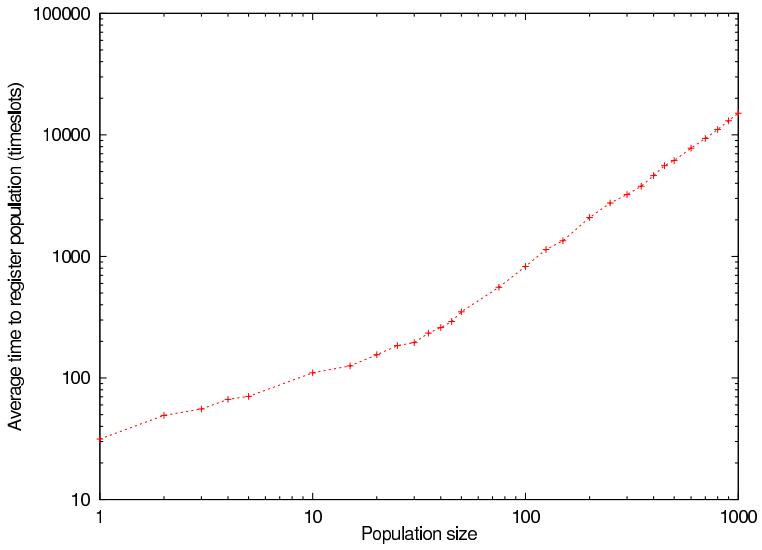


Figure 8.5: Simulated registration performance

8.2.7 Evaluation

Sixteen MPDs have been deployed in the prototype installation, and the maximum rate at which a particular object can be tracked is 12.5Hz. The location system management software performed as expected, as did the procedure by which MPDs hand over between radio cells. The maximum RF power emitted by MPDs is 0.25mW, and so the system conforms with the safety evaluation made in Section 5.5.7.

The battery lifetimes of MPDs are highly dependent upon such factors as the time spent registering with ZMs, the frequency with which handovers are made, and the extent to which the MPDs remain in the sleep and searching states. Since these factors are themselves dependent on the pattern of use of the MPDs, only rough estimates of the battery lifetime can be made.

Consider an MPD that has registered with a ZM, and that receives every addressing message from the ZM. Most addressing messages will not be destined for the MPD, so it will typically receive only message segments 1 and 2. Together with the preamble, and the 500μs pre-message activation period (see Section 8.2.6), this implies that the MPD's receiver circuitry and fast clock will be active for around 6ms, 12.5 times each second. Coupled with the 1Ah battery capacity of the MPD, and the current consumption levels presented in Section 5.5.6, this implies an approximate lower bound on the battery lifetime of around 35 days. The expected battery lifetime of an MPD that remains in the searching state is around 145 days, and an MPD that spends most of its time in the sleep state, checking only the first segment of every eighth message, should have a battery lifetime of around 255 days. This range was considered to be acceptable in the context of the prototype system.

8.3 Software System Implementation

Chapter 7 describes a software architecture that is suitable for the management and dissemination of fine-grain location information. This section describes an implementation of that architecture, built using distributed objects that communicate across CORBA interfaces, and that store configuration information in a relational database.

8.3.1 Object implementations

The software counterparts of physical objects have been implemented as persistent distributed objects using CORBA and an Oracle 7 database. A package called *Ouija* [Steggles98], which provides an object-oriented data modelling language and CORBA mapping for PL/SQL (a proprietary Oracle procedural extension to SQL), was used to build an object-oriented layer on top of the relational Oracle database. Objects are stored in the database as rows of data and associated PL/SQL operations. They are accessed via a proxy that receives CORBA calls and forwards them to the appropriate PL/SQL operation, via Oracle's proprietary API.

The prototype object implementations take advantage of the work of the *SPIRIT* project [Steggles98]. A data model that describes objects in an office environment (such as people and computers), and proxies that provide access to stored information about those objects have already been written as part of the SPIRIT effort. For the purposes of the location system, these proxies are extended with CORBA interfaces that receive position callbacks from OMs, and that pass the object state on to client applications and services, as described in Section 7.2. Proxy implementations also contain the per-object-type code used to transform the location system data into object state information (see Section 7.2.1), and may make use of the object information stored in the database to effect this transformation.

It is possible that thousands of objects might be tracked by the location system. Furthermore, many objects (e.g. those representing printers) will receive only infrequent updates from the location system, and their proxy implementations will be inactive much of the time. Caching is therefore used to reduce the total level of resources required by object proxies. A proxy server creates a proxy for an object in the database when, and if, it is first accessed. The proxy server can manage the number of active proxies by unloading object implementations on a least-recently-used basis. Subsequent calls to unloaded proxies cause them to be reloaded.

8.3.2 Callbacks

The callback mechanism described in Section 7.2.2 is implemented on the server side by providing a CORBA registration interface that accepts a reference to a CORBA callback interface on the client object. The client is

notified via this interface when new information becomes available from the server. A server-side CORBA deregistration interface allows clients to indicate that they are no longer interested in receiving callbacks.

In some cases, a callback request may automatically reserve location system resources. If the client object were to terminate abnormally without deregistering, those resources might be tied up without purpose for some time. Furthermore, objects receiving callbacks must ensure that their requests are reiterated if the server terminates and is restarted. For these reasons, clients are required to periodically reregister with the callback server to keep their requests and associated resource demands live.

Persistent objects, by their very nature, never terminate. However, persistent client objects must still be alerted if associated non-persistent servers fail and forget their requests. Persistent server objects must also be alerted if non-persistent clients fail, so that allocated resources can be reclaimed. Since persistent objects may not always be loaded into proxy servers, they cannot monitor the state of non-persistent objects involved in callback transactions themselves. A further, non-persistent software entity called the *session manager* therefore performs this task, managing reregistration on behalf of persistent objects.

8.3.3 Spatial indexing implementation

A 2D spatial index covering the area in which the location system is installed has been implemented. The service provides CORBA interfaces for indexing simple polygons and circular regions at specified resolutions. Each indexed region is tagged with the CORBA object reference (if appropriate) and textual description of its owner, allowing the physical object or area associated with the space to be identified. Regions also have a textual type that determines the meaning of the space, and are timestamped in the manner described in Sections 7.2.2 and 7.3.2.

A further CORBA interface allows client applications to specify the types of spatial event in which they are interested. Applications can define sets of space types, and can express interest in both overlapping and containment events between members of pairs of type sets. The use of type sets allows applications to make complex queries (such as "Tell me when a person walks past a monitor or loudspeaker") without managing large numbers of independent callback requests.

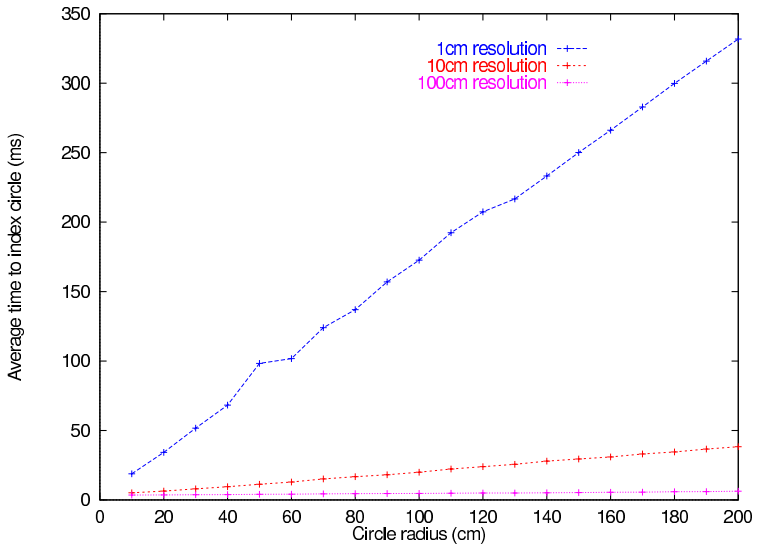


Figure 8.6: Spatial index performance

Figure 8.6 shows the time taken to index a circular space of varying radius at different resolutions on a 110MHz Sun SPARCstation-4. It can be seen that the indexing time is roughly proportional to the radius (and hence perimeter) of the space, and is inversely proportional to the resolution. It can also be seen that with reasonable indexing resolutions ($\geq$ 10cm) and space sizes, the current, unoptimised spatial index is fast enough to handle the maximum data rate from the location system.

Figure 8.7 demonstrates the spatial sorting properties of the index. A population of circular spaces was placed into the index, as was a circular test space of radius 1m at 10cm resolution. When the test space was placed such that it overlapped the other regions, the time required to index the test space was roughly linear to the size of that population, because all relationships between interacting spaces that might generate events were tested. However, when the test space was placed far away from the population, no such tests were made, and the time required to index the test space was independent of the population size. Assuming a relatively uniform density of spaces within typical buildings, indexing times will therefore be independent of the size of those buildings.

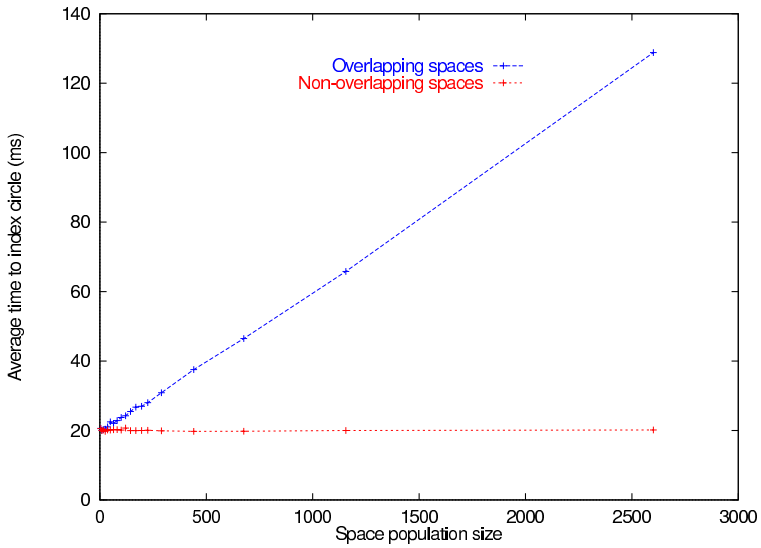


Figure 8.7: Spatial index performance with varying space population size

8.4 Summary

This chapter has described a scalable location system that may be deployed in an indoor environment. The system implementation demonstrates many of the principles set out in previous chapters, and the performance of features such as registration and spatial indexing has been determined and found to be satisfactory. Based on this location system, the next chapter explores possible applications of location-aware technologies.

Chapter 9

Applications

9.1 Introduction

This chapter examines applications that might make use of data gathered by an indoor location system. Experience with Active Badges suggests that many potential uses of this information will only become evident after such systems have been fully deployed. Several application areas have already been identified, however, and are discussed below. In order to illustrate these possibilities, a number of prototype location-aware systems that have been developed using the current indoor tracker are also described.

9.2 Future Vision

The falling cost of microelectronics in real terms, improved low-power design methods, and miniaturisation of components may well allow the deployment of cheap, truly unobtrusive ultrasonic location systems. It would not be unreasonable to envisage buildings in which the positions and orientations of everything and everyone inside were accurately known.

Work towards next-generation location systems is ongoing. Figure 9.1 shows a miniaturised MPD, which has been developed in collaboration with Steve Hodges of the Olivetti and Oracle Research Laboratory, Cambridge. The MPD measures 5cm $\times$ 3cm $\times$ 2cm, weighs 35g, and is powered by a single 3.6V battery. Improvements in receiver signal processing techniques may permit distance measurements accurate to one ultrasonic wavelength (0.9cm at 40kHz) or less, with corresponding position accuracies in the

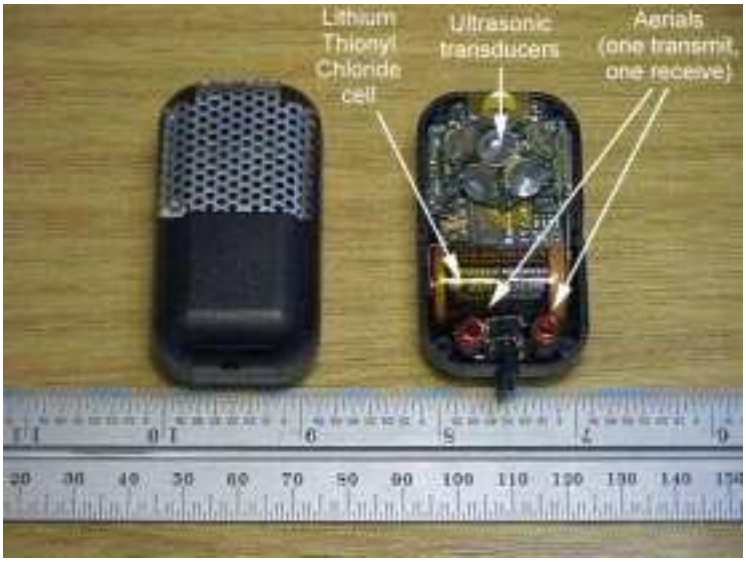


Figure 9.1: External and internal views of a miniaturised MPD

range 1–2cm, and MPD battery lifetimes of several years seem feasible. It is hoped that a building-wide location system will soon be deployed.

This section discusses how a highly accurate, ubiquitous location system might be used, and identifies possible limitations to such systems. Privacy and confidentiality issues associated with tracking systems are briefly examined.

9.2.1 Location-aware application areas

The availability of fine-grain location data will allow existing location-aware applications to be improved, and will permit development of new location-aware systems. Some potential application areas of an ultrasonic location system are investigated below.

Automation and personalisation

Many Active-Badge-based systems (e.g. nearest printer functions, telephone call routers, mobile desktops, etc.) could be enhanced by using fine-grain location information to make more appropriate automation and personalisation decisions. Complex communication systems that are currently infeasible, such as the walk-through videophone described in Chapter 2, could be automatically configured using accurate location data, and computing de-

vices could use knowledge of their position and orientation to modify their behaviour. Consider, for example, a pen-based portable PC tracked using several MPDs. If the device was rotated from a landscape orientation into a portrait orientation, the PC's operating system could be instructed to adjust the display and windowing system accordingly, allowing the device to be used normally in either state.

Construction-kit computing is a computing paradigm driven by fine-grain location information. The paradigm involves the connection of computing and communications computing elements by placing them together to form more complex systems, in the manner of children's construction sets. The spatial configuration of the component devices automatically determines their logical configuration. Consider, for example, a display tile device on which a video stream was being shown. By taking three additional tiles, and arranging the displays in a 2×2 rectangle, a larger screen would effectively be formed. A location-aware application could track displays to detect when such formations were created, and could then expand the video stream across all four tiles to make full use of the larger display area.

New human-computer interfaces

The idea of construction-kit computing may be applied to the intuitive control of data streams between networked devices. Cameras and displays with high-speed wireless network interfaces may well become available in the near future. Typically, a camera and display would be connected using knowledge of their network addresses. An alternative user interface, however, might allow a video stream to be connected between the two by momentarily bringing them close to each other, freeing users from the requirement of knowing the addresses of the devices.

Fitzmaurice has described 3D information areas around physical objects called *Spatially Situated Information Spaces* [Fitzmaurice93]. Users interact with these computer-synthesised spaces through location-aware PDA "portholes"; for example, a PDA placed near a printer might display the contents of the job queue. In the original research, the PDAs were located using tethered electromagnetic trackers—the use of a wireless location device such as that described in this dissertation would ease interaction with such spaces.

Gesture-based interfaces, driven by input from MPDs used as 3D mice, could be developed. Existing systems that use tethered trackers to monitor

the position of a user's hand (for example, the *CHARADE* system, which controls a computer whilst the user is giving a presentation [Baudel93]) could be made more mobile, and those which infer gesture information from video images (for example, the *Pfinder* tracker [Wren96]) could be made more robust in the presence of moving objects other than the user.

Fine-grain location information could be used to prevent unauthorised access to computing and communications resources or information. Telephones in a university department, for example, might allow outgoing calls only when used by a member of staff. Similarly, a confidential printing service could suspend the output of a document until the owner was stood next to the printer ready to collect it.

Resource tracking

Resource-tracking applications are used to identify the current location of objects, answering queries such as "Where is the multimeter?" Another application of this type is a moving indoor map, attached to a shopping trolley in a supermarket, which displays a route to some selected product in the shop based on the current position of the trolley. The chosen route would, of course, be engineered to pass through the special offer section. Abowd *et al.* describe a tour guide based around a moving map [Abowd97] that would also benefit from the use of detailed indoor location information. 3D surveying of an indoor environment, using a single MPD to determine the positions of desks, cupboards, and other relatively fixed objects is a further example of a resource-tracking application that uses fine-grain location information.

Monitoring and control applications

An *Interactive Office* described by Hodges and Louie [Hodges94] uses data from simple sensors such as motion detectors and reed switches, distributed around a workplace, to perform tasks such as the transfer of telephone calls to answering machines during meetings. Section 1.1 highlights the installation and data management problems associated with a diverse set of sensors. The location system described in this dissertation, however, could provide information about the state of doors (i.e. open or closed) and levels of motion in offices using a single type of untethered sensor, thus simplifying the hardware and software requirements of Interactive Offices.

Existing systems that optimise lighting and heating within offices using Active Badge information [Elrod93] could take advantage of fine-grain location information to make more intelligent control decisions. Furthermore, memory prostheses could use data from an ultrasonic location system to record more detail about the surroundings of their users than is available from, say, Active Badge sources, aiding information retrieval at later times.

Simple robots could be guided around buildings by equipping them with one or more MPDs. Each robot would have little on-board intelligence or sensing capability, but would rely instead on a centralised service that instructed them on how to perform their tasks. The service's instructions would be based upon its knowledge of the position, orientation and surroundings of each robot, gathered using sensor systems installed throughout buildings. Similarly, robots could be trained to follow a path laid out by moving an MPD through space.

Entertainment

Many possibilities exist for the use of an indoor location system in entertainment systems. Perhaps the location-aware *tamagotchi* (virtual pets) of the future will need to be taken into the kitchen to be fed!

9.2.2 System limitations

As mentioned in Section 2.3.1, the current ultrasonic location system has not been designed to meet the rigorous position and orientation demands of VR and AR applications involving head-mounted displays, and is therefore unsuitable for these tasks. The system is also unsuitable for use in industrial environments where grinding, laser etching or high-speed spraying processes are operated, because those processes are known to generate large amounts of ultrasonic noise around 40kHz [Bass85].

9.2.3 Acceptability of invasive technologies

Location systems such as the one described in this dissertation can be viewed as being invasive, threatening personal privacy. If not addressed, these concerns might prevent widespread acceptance of location systems. Spreitzer and Theimer [Spreitzer93], and Bellotti and Sellen [Bellotti93] present good

overviews of the problems surrounding the preservation of privacy in ubiquitous computing environments. Technical features, such as user agents designed to control personal information [Spreitzer94], provide some protection against misuse of data, and could be incorporated into system designs. If abuse of location systems became commonplace, it might be necessary to introduce appropriate legislation, perhaps preventing employers from forcing employees to carry MPDs or similar devices [Want92a].

It should be noted that other technologies that can reveal location information (such as mobile telephones and credit cards) are used on a daily basis by many people. In these cases, users are presumably unaware of the privacy risks of the systems, or are prepared to forgo assured privacy in exchange for the services provided by them. Since the infrastructure and risks associated with an indoor location system are relatively obvious, location-aware applications will have to provide users with tangible benefits if they are to be widely embraced. Further discussion of privacy topics is beyond the scope of this dissertation.

9.3 Working Prototypes

A number of prototype applications have been implemented using the current location system to assess its suitability for supporting location-aware devices and services.

9.3.1 Moving indoor map

A mobile display unit showing the current location of the device in relation to its environment has been implemented using a pen-based PC with a wireless LAN interface, as shown in Figure 9.2. Two MPDs are placed on the display, allowing its position and orientation around a vertical axis to be determined twice a second. This information is then passed to a process running on the portable PC, which correctly positions and orients a map of the building displayed in a VRML (Virtual Reality Modelling Language) browser.

The information presented by the moving map was found to represent satisfactorily the surroundings of the device for the purposes of navigation. Further evaluation using a wider deployment of the location system would be required to determine the general usefulness of such a map.

Figure 9.2: A moving indoor map

9.3.2 Workstation personalisation

The *Virtual Network Computing* (VNC) system [Richardson98] allows a user to access their personal desktop from any Internet-connected machine. A server manages a desktop environment, and renders it in a framebuffer held in memory. Changes to the framebuffer contents are sent to clients using a simple protocol. By rendering these changes on a local display, clients can duplicate the visual state of the centrally-served desktop. Keystrokes, mouse clicks and other input events are sent back to the server from clients, and are routed to the desktop, which then acts as though it was hosted by the local machine.

In one mode of VNC operation, clients listen for incoming connections from servers. VNC servers have a CORBA interface, which can be used to initiate connections with particular clients or close connections down. Information from the location system can be used to determine the set of displays in the immediate environment of a person, and, using the server's CORBA interface, their desktop can be moved automatically to one of those displays. Workstations can therefore be personalised to a user as they walk up to them, without the need for users to log themselves in.

VNC clients were installed on several machines within the volume of the deployed location system. Two MPDs were placed on the monitor of each machine, allowing the positions and orientations of those displays to

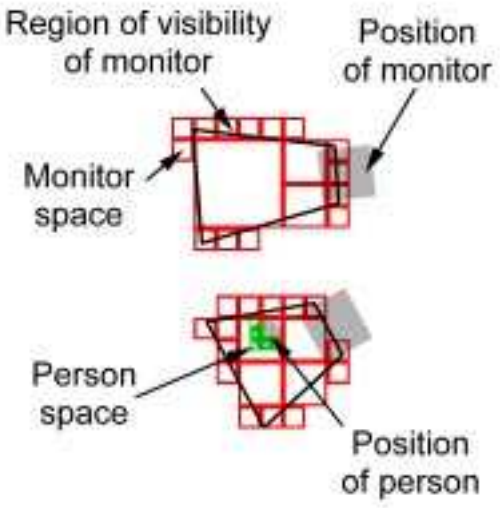


Figure 9.3: Spatial index entries for workstation personalisation

be determined. Using this information, trapezoidal regions (Monitor spaces) corresponding to the areas in which information displayed on each monitor could easily be seen were entered into the spatial index. Circular Person spaces corresponding to users were also placed in the index store. Figure 9.3 shows a view of the indexed regions. The locations of people were determined three times every second, whilst monitors were located only once a second.

The workstation personalisation application registers interest in positive containment and negative overlapping events between Person and Monitor spaces. A positive containment event between a Person and a Montior space indicates that the user is in a position to view their desktop on the display associated with the space. When such an event occurs, the application instructs the VNC server (via the CORBA interface) to move the user's desktop on to that display, as shown in Figure 9.4. Note that the orientation of the user is not taken into account by this application.

A negative overlapping event between a Person and a Monitor space indicates that the user can no longer view their desktop, and causes it to be removed from the display associated with the space. The use of the positive containment and negative overlapping events ensures some degree of hysteresis in the selection of suitable displays by the personalisation application, as shown in Figure 9.5.

The personalisation system described above was found to be easy and

Figure 9.4: Personalising a workstation

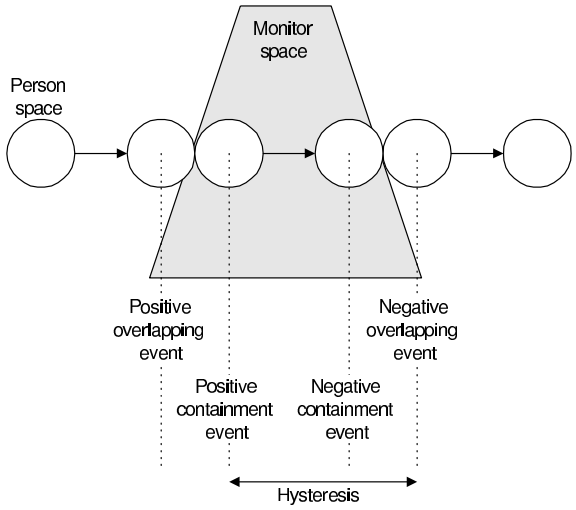


Figure 9.5: Spatial events driving personalisation

intuitive to use. However, its mode of operation is promiscuous, i.e. other than their being near a display, users play no part in selecting when their desktop should appear. This behaviour might be confusing or irritating—as people walked around a building, their desktop would flash up briefly on each screen they passed. It is therefore suggested that in practice, users would indicate that they wished to access their desktop and applications, perhaps by pressing one of the MPD buttons whilst standing near a display. Only at that time would the desktop be made visible, although the personalisation application would continuously monitor spatial events to determine the set of suitable displays.

9.3.3 Construction-kit computing

The VNC system allows a desktop to be rendered on more than one display simultaneously. All input devices associated with those displays can be used to control the desktop. Experiments in construction-kit computing may then be performed by equipping PCs with VNC viewers and wireless infra-red keyboards.

Consider the following workstation personalisation rule—a user's desktop is rendered on a display if they are either within the viewing region of that display *or* they are using an input device associated with that display. A circular Keyboard space is indexed for each keyboard, as shown in Figure 9.6, and the personalisation application reacts additionally to positive containment and negative overlapping events between Keyboard and Person spaces, in order to determine when a person is using a keyboard. By selecting any keyboard and bringing it up to a display, a user's desktop and applications will then appear on both the keyboard's "home" display and the display in front of the user. Input from keyboards associated with both displays will be routed to the desktop. Effectively, the keyboard carried by the user is coupled with the display in front of them simply by placing the components together.

This behaviour might be useful in office situations where personnel are highly mobile but each have a set of preferred input devices, such as keyboards with particular layouts or tactile characteristics. Ideally, keyboards and displays would be treated completely independently by the VNC system. It would then be possible to retain the construction-kit functionality described above, whilst ensuring that a user's desktop only ever appeared

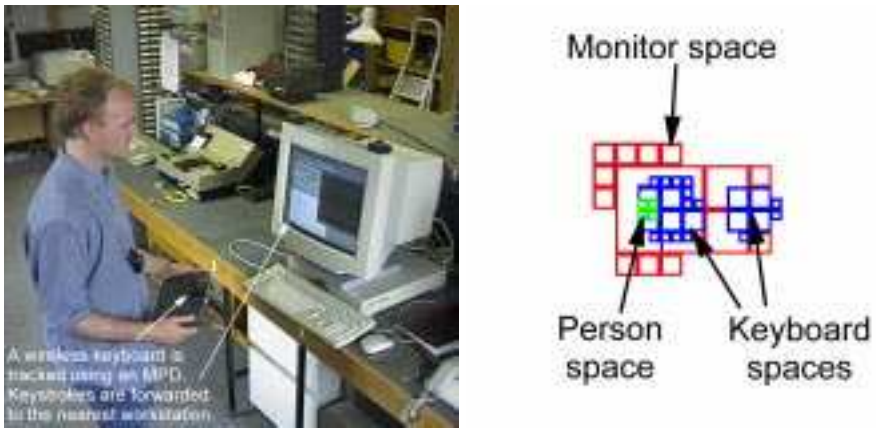


Figure 9.6: Construction-kit computing

on the screen in front of them.

9.3.4 Automatic camera selection

One component of the walk-through videophone system described in Chapter 2 is an automatic camera selector, which continuously chooses the most suitable camera with which to view a user. Whenever possible, the selector should pick a camera that has a view of the user and that the user is facing. This application therefore requires position and orientation information for both cameras and users.

A distributed multimedia environment called *Medusa* [Wray94] is used to manage an LCD display tile and four cameras, placed around an office and connected using an ATM network. A software video stream multiplexer placed in the data path between the cameras and the display tile allows the selector application to route the video stream from a chosen camera to the display tile. Trapezoidal Camera regions corresponding to the fields-of-view of cameras are placed in the spatial index[1], as are circular Person regions corresponding to people, as shown in Figure 9.7.

By registering interest in positive containment and negative overlapping events between Person and Camera spaces, the selector continuously maintains a list of the cameras that can see a user. The orientation of the user,

[1]Cameras were fixed at particular points, in set orientations, although in principle their positions and orientations could be determined using information from the location system.

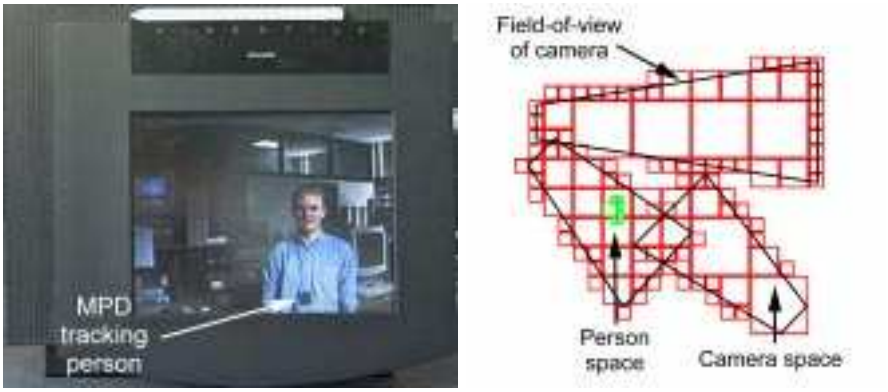


Figure 9.7: Automatic camera selection

found using the techniques outlined in Section 4.4.2, determines which of those cameras should be selected as the video stream source. The following rules are used by the selector to make this choice:

1. If the current camera is in the list of suitable cameras, and the user is not facing more than 90° away from the camera, retain that choice.

2. Otherwise, select the camera that the user is facing most directly from the list of suitable candidates.

Once a camera has been chosen, the selector arranges that its output is sent to the display tile by enabling the appropriate path through the video multiplexer.

The automatic camera selector was found to work extremely well. The rules used by the selector introduce hysteresis into the system, minimising "flicking" between camera views. Unlike other proposed camera selectors, which interpret the contents of the video stream from each camera to deduce the state of the environment (see, for example, the work of Pinhanez and Bobick [Pinhanez97]), the location-aware selector cannot be confused by people other than the user moving in the fields-of-view of cameras. Furthermore, the above approach can be adapted to select suitable displays, microphones and loudspeakers as input and output devices for the videophone application.

9.4 Summary

The deployed ultrasonic location system has been used to develop applications that, due to their location-aware nature, either enhance existing systems (such as the VNC system) or perform tasks that would otherwise be infeasible (e.g. the moving indoor map). The prototype applications were found to be extremely intuitive to use, and demonstrated correct operation of both hardware and software elements of the underlying location system. Positive experiences gained with the prototype applications suggest that information gathered by ultrasonic location sensors would be suitable for use in the wide range of location-aware applications identified in this chapter.

Chapter 10

Conclusion

This dissertation has described an ultrasonic location system and supporting software architecture that can be used to implement new types of location-aware application. This chapter summarises this work and its conclusions, and highlights areas that deserve further investigation.

10.1 Summary

Previous research into location-aware computing has made use of infra-red-based tracking systems such as the Active Badge, which provide location information with room-scale granularity. Chapter 2 asserts that the coarse granularity of this data limits the extent to which location-aware devices and applications can adapt to their environment, and that a new type of location sensor, which provides more detailed information, is needed. The chapter also describes the properties that such a sensor must possess if it is to be useful for location-aware computing.

Given that a new sensor is required, it is natural to attempt to identify existing technologies that can be adapted to serve this purpose. Chapter 3 gives an overview of current tracking systems that can operate in indoor environments, but none is directly suitable for the task of providing fine-grain location information for context-aware systems. Ultrasonic schemes, however, seem to have many desirable characteristics, such as low-cost, ease of implementation and relatively high accuracy.

Chapter 4 describes the principles of operation of a new fine-grain tracking system based on ultrasonic multilateration. This sensor is designed to

be wireless, low-powered and easily integrated into existing environments. Ultrasonic pulses are sent from transmitters attached to objects to a set of fixed receivers, and measurements of pulse times-of-flight are used to calculate transmitter-receiver distances. Nonlinear regression is then used to convert these distances into 3D transmitter positions. Outlier analysis and geometrical methods are used to reject erroneous distance measurements caused by reflections of ultrasound in the indoor environment, and the tracker can determine the orientations of objects in a number of ways.

A description of an implementation of the tracking system is given in Chapter 5, and a thorough analysis of its performance is made. The accuracy of the tracker's position and orientation measurements have been evaluated, as have the effects of measurement geometry and averaging. Further investigation has determined the levels of spatial distortion of measurements, and the degree to which readings are temporally correlated. This detailed knowledge of the operational capabilities of the sensor will allow location-aware applications to be tailored to the information it generates.

Chapter 6 describes the extension of the ultrasonic tracker into a scalable location system. The rate at which objects can be located by the system is finite, and so it supports different location rates for different objects, in order to use the available tracking resources more effectively. The system can track a changing population of objects by registering the arrival of new objects, and reclaiming location resources allocated those which leave the tracking space. This chapter also discusses methods for increasing the rate at which objects can be located, using near-simultaneous triggering of transmitters in distinct ultrasonic spaces, and describes multi-cellular radio techniques that allow the tracking system to cover larger areas.

A software architecture that controls the management and dissemination of location information is presented in Chapter 7. A flexible per-object modelling approach is used to convert ultrasonic transmitter positions into object positions and orientations. The use of an indexing structure to optimise queries involving spatial relationships between objects is described, as are mechanisms for evaluating proximity queries. It should be noted that the software architecture presented in this dissertation will be suitable for the management of fine-grain location information generated using other tracking technologies, should they become available in the future.

The ultrasonic location system and software architecture have been im-

plemented and integrated, as described in Chapter 8. The demonstrated location system has a greater operational volume than any other fine-grain, indoor tracking system described in the literature, whilst retaining the low-power, wireless characteristics required for use in location-aware computing environments.

Finally, Chapter 9 describes several location-aware applications that are based upon the prototype location system. These applications proved to be very intuitive to use, and could not have been implemented using data from existing location sensors. They therefore demonstrate both that fine-grain location information can be used to develop novel sensor-driven systems, and that ultrasonic tracking is a viable method of gathering that information.

10.2 Further Work

The ultrasonic location system described in this dissertation is an operational prototype, and further work will need to be undertaken before it can be deployed as part of the infrastructure of a computing environment.

The MPDs attached to objects are the most obtrusive element of the location system, and further miniaturisation of these devices is necessary. The largest individual components of the most recent design (see Section 9.2) are the battery and piezoceramic transducer array. By decreasing the current consumption of the MPD components, a smaller battery could be used whilst retaining the same battery lifetime, and omnidirectional, flexible-film transducers constructed using polyvinylidene flouride (PVDF) material [Paradiso96] may reduce the volume of the ultrasound source.

The relative merits of using other types of channels to send addressing messages to MPDs should be evaluated. In particular, infra-red signals have containment properties similar to those of ultrasound, and an infra-red control network might simplify simultaneous operation of MPDs in neighbouring ultrasonic spaces. However, the use of infra-red signals for communication is unregulated, and interference with other infra-red-based systems is likely.

Receivers employing correlation techniques (see Section 3.4.1) should be developed to increase the MPD-receiver distance measurement accuracy, and hence increase the position accuracy of the system. These receivers would also be more effective in separating positioning signals from background noise, thus increasing the detection range of receivers and/or allowing the

use of lower intensity ultrasonic pulses. Experiments should be performed to evaluate the location system's performance in different types of building, such as open-plan and warehouse environments, and ceiling self-calibration (Section 5.4.3) should be evaluated as a method of simplifying the task of system installation.

Several issues regarding the scalability of the ultrasonic location system deserve further investigation. Although this dissertation describes a simple deployment using radio cells of two colours, separated using time-division multiplexing, a building-wide deployment might require ten or more colours, and experience of such a system would be valuable. Other, more complex multiplexing strategies should also be studied. For example, schemes that permit cells of more than one colour to be active simultaneously (see Section 6.5.3) may allow trackers with higher aggregate location rates to be developed.

Other sensing technologies that can supply fine-grain location information may become available in the future. Of particular interest are *ultra-wideband* radio systems [James95], which perform range-finding and communication using extremely short pulses of radio energy and complex pseudo-random pulse trains. Although this technology is immature, it appears that radiolocation systems developed from it may be relatively immune to the multipath problems that affect other radio-based trackers in indoor environments (see Section 3.3.2).

It would be desirable to have cooperation between different domains in which fine-grain location systems were deployed. It is unlikely that real-time location information would be sent between sites, because bandwidth is limited and data latencies are high in the wide-area, although more coarse-grain views of that data, such as "Andy Ward is in the New York Hilton", might be usefully distributed. However, suppose that a user carrying an MPD visited a new domain in which a location service was available. MPD registration messages might include an address for a remote service that indicated to local systems the behaviour which the user would expect of them, thus enabling a truly ubiquitous computing experience. By combining an indoor tracking system with GPS technology, location updates could also be provided as objects moved between sites—interestingly, both GPS receivers and MPDs should be placed facing upwards for maximum location system performance, suggesting that the two could be integrated conveniently. Clearly,

there is substantial scope for research in this area.

The software architecture presented in Chapter 7 could be extended in a number of ways. It would be desirable to provide some level of system support for the variation of location priorities that should occur when, say, a person walked up to a workstation (see Section 6.2.1). Applications would then be freed from the task of watching for such occurrences and acting upon them. A service that balances demands for location information from different applications is required to ensure predictable behaviour of the location system with many clients, and Section 7.2.3 sets out a number of possible arbitration criteria. Furthermore, improved management of historical location information would allow location-aware applications to take account of past movement patterns when changing their behaviour.

Finally, the application space of the location system should be expanded. This research must address the privacy concerns that are associated with the collection of fine-grain location information—only if location-aware systems are acceptable to their potential users will be they be widely adopted.

Appendix A

The Position Dilution Of Precision

The Position Dilution Of Precision (PDOP) is a quantity that relates the radial range measurement accuracy of receivers to the position measurement accuracy of the ultrasonic location system. The PDOP of a position measurement depends on the relative geometry of the MPD and the receivers to which its distance is measured. This section derives an expression for PDOP in the context of the ultrasonic location system, and is based on the derivation given in [NATO91] of the PDOP formula used in the Global Positioning System.

The basic multilateration equation, described in Section 4.3.2, is nonlinear. However, the effects on an MPD's position of *small* changes in its ranges from a set of receivers can be approximated using a set of linear equations. Consider an MPD, whose position in the reference frame $\mathcal{W}$ (described in Section 4.3.2) is $(\hat{u}, \hat{v}, \hat{w})$. Suppose that the distances from this MPD to receivers at positions $(x_1, y_1, 0), \ldots, (x_n, y_n, 0)$ (where $n \geq 3$) are $\hat{l}_1, \ldots, \hat{l}_n$. Suppose also that a small change, $(\Delta u, \Delta v, \Delta w)$ is made to the MPD's position, moving it to $(\hat{u}', \hat{v}', \hat{w}')$. This will result in changes, $\Delta l_1, \ldots, \Delta l_n$, in each of its distances from receivers, which become $\hat{l}'_1, \ldots, \hat{l}'_n$. Then, for $i = 1 \ldots n$,

$$\hat{l}_i = \sqrt{(\hat{u} - x_i)^2 + (\hat{v} - y_i)^2 + \hat{w}^2} \tag{A.1}$$

and

$$\hat{l}'_i = \sqrt{(\hat{u}' - x_i)^2 + (\hat{v}' - y_i)^2 + (\hat{w}')^2} \tag{A.2}$$

where

$$\hat{l}'_i = \hat{l}_i + \Delta l_i$$

$$\hat{u}' = \hat{u} + \Delta u$$

$$\hat{v}' = \hat{v} + \Delta v$$

$$\hat{w}' = \hat{w} + \Delta w$$

Fully expanding Equation A.2,

$$\hat{l}_i + \Delta l_i = \sqrt{(\hat{u} + \Delta u - x_i)^2 + (\hat{v} + \Delta v - y_i)^2 + (\hat{w} + \Delta w)^2} \quad \text{(A.3)}$$

Ignoring second-order terms, Equation A.3 can then be written as

$$\begin{aligned} \hat{l}_i + \Delta l_i &= \sqrt{(\hat{u} - x_i)^2 + (\hat{v} - y_i)^2 + \hat{w}^2} \\ &\quad + \frac{\Delta u(\hat{u} - x_i) + \Delta v(\hat{v} - y_i) + \Delta w \cdot \hat{w}}{\sqrt{(\hat{u} - x_i)^2 + (\hat{v} - y_i)^2 + \hat{w}^2}} \end{aligned} \quad \text{(A.4)}$$

Therefore,

$$\hat{l}_i + \Delta l_i = \hat{l}_i + \frac{\Delta u(\hat{u} - x_i)}{\hat{l}_i} + \frac{\Delta v(\hat{v} - y_i)}{\hat{l}_i} + \frac{\Delta w \cdot \hat{w}}{\hat{l}_i} \quad \text{(A.5)}$$

The n relationships with the form of Equation A.5 can be expressed using matrix notation, as below:

$$\begin{pmatrix} \Delta l_1 \\ \vdots \\ \Delta l_n \end{pmatrix} = \begin{pmatrix} \frac{\hat{u}-x_1}{\hat{l}_1} & \frac{\hat{v}-y_1}{\hat{l}_1} & \frac{\hat{w}}{\hat{l}_1} \\ \vdots & \vdots & \vdots \\ \frac{\hat{u}-x_n}{\hat{l}_n} & \frac{\hat{v}-y_n}{\hat{l}_n} & \frac{\hat{w}}{\hat{l}_n} \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \end{pmatrix} \quad \text{(A.6)}$$

Let

$$\mathbf{r} = \begin{pmatrix} \Delta l_1 \\ \vdots \\ \Delta l_n \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} \frac{\hat{u}-x_1}{\hat{l}_1} & \frac{\hat{v}-y_1}{\hat{l}_1} & \frac{\hat{w}}{\hat{l}_1} \\ \vdots & \vdots & \vdots \\ \frac{\hat{u}-x_n}{\hat{l}_n} & \frac{\hat{v}-y_n}{\hat{l}_n} & \frac{\hat{w}}{\hat{l}_n} \end{pmatrix}$$

$$\mathbf{x} = \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \end{pmatrix}$$

Then, from Equation A.6,

$$\mathbf{r} = \mathbf{A}\mathbf{x} \tag{A.7}$$

Hence,

$$\mathbf{A}^{\mathrm{T}}\mathbf{r} = \mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} \tag{A.8}$$

and

$$(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\,\mathbf{A}^{\mathrm{T}}\mathbf{r} = \mathbf{x} \tag{A.9}$$

Equation A.9 is a linear relationship that describes the effects of small changes in distance measurements from receivers on the MPD position. Since the relationship is linear, it can be used to describe the link between a distance measurement error vector (ϵ_r) and an MPD position error vector (ϵ_x), as below:

$$\epsilon_x = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\,\mathbf{A}^{\mathrm{T}}\epsilon_r \tag{A.10}$$

Consider now the covariance matrix describing the expected errors in distance measurements and the covariance matrix describing the expected errors in the MPD position. These are

$$\mathrm{cov}\,(\mathbf{x}) = \mathrm{E}\,\{\epsilon_x \epsilon_x^{\mathrm{T}}\}$$

and

$$\mathrm{cov}\,(\mathbf{r}) = \mathrm{E}\,\{\epsilon_r \epsilon_r^{\mathrm{T}}\}$$

where the $\mathrm{E}\{\ \ \}$ designates the expected value of the quantity inside the braces. Since

$$\epsilon_x \epsilon_x^{\mathrm{T}} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\,\mathbf{A}^{\mathrm{T}}\epsilon_r \epsilon_r^{\mathrm{T}}\mathbf{A}\,(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1} \tag{A.11}$$

it follows that

$$\mathrm{cov}\,(\mathbf{x}) = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\,\mathbf{A}^{\mathrm{T}}\mathrm{cov}\,(\mathbf{r})\,\mathbf{A}\,(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1} \tag{A.12}$$

Assume, then, that the individual measurements of distances from the MPD to receivers have a one-sigma error of unity and expected mean of zero, and that there is no correlation of errors between distance measurements. The covariance matrix $\mathrm{cov}(\mathbf{r})$ is the identity matrix, and

$$\mathrm{cov}(\mathbf{x}) = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1} \tag{A.13}$$

PDOP is defined as the trace of the MPD position covariance matrix when the distance measurement covariance matrix is the identity matrix. It can therefore be seen that

$$\mathrm{PDOP} = \sqrt{\mathrm{Trace}\left((\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\right)} \tag{A.14}$$

This quantity is dependent only on the relative geometry of the MPD and receivers, and provides an estimate of the magnification of distance measurement errors into MPD position errors due to that geometry.

Equivalently, if V_x, V_y and V_z are the variances of the MPD position along the axes of the reference frame $\mathcal{W}$, given by the values on the diagonal of $\mathrm{cov}(\mathbf{x})$, then

$$\mathrm{PDOP} = \sqrt{V_x + V_y + V_z}$$

Other measures of the impact of the measurement geometry on the position error can be determined. The *Horizontal and Vertical Dilutions Of Precision*, HDOP and VDOP, which indicate the effects of the measurement geometry on the horizontal and vertical components of the position error, are given by:

$$\mathrm{HDOP} = \sqrt{V_x + V_y}$$

$$\mathrm{VDOP} = \sqrt{V_z}$$

The square roots of each of the variances V_x, V_y and V_z also define the dilution of precision in each of the corresponding coordinate axes.

Appendix B

A Simple Scheduling Algorithm

This section describes a simple scheduling algorithm that has been developed for use by the ultrasonic location system.

The scheduling table

The scheduler maintains a *scheduling table* stored in memory. Each row of the table contains six entries:

1. An MPD address, a.
2. An MPD group identifier for that MPD, g.
3. The LQoS for the MPD, l.
4. A number called the *Effective LQoS* or *ELQoS*, e, which represents the effective rate at which the location system will address an MPD. The ELQoS will be equal to the LQoS if the scheduler is not overloaded, or less than the LQoS otherwise.
5. A number called the *score*, s.
6. A bit called the *slotswap* bit, b.

In this appendix, a row in the scheduling table is described by an ordered six-tuple (a, g, l, e, s, b). A register called *lastgroup* is also managed by the scheduler. MPD address 0 is reserved, as is MPD group identifier 0.

Scheduler initialisation

When the scheduler is initialised, the scheduling table contains a single row, $(0, 0, 1, 1, 0, \texttt{false})$—this entry represents a dummy MPD. The *lastgroup* variable is set to zero.

Adding or changing a request

Messages can be sent to the scheduler to indicate that the location requests for an MPD group should be added or changed. It should be noted that if two or more MPDs managed by a ZM are members of the same MPD group, then their LQoS requests may not be added or changed separately. Furthermore, requests involving the reserved MPD address 0 and reserved MPD group identifier 0 will be ignored. The following information must be passed to the scheduler:

- g, the MPD group identifier.
- l, the desired LQoS for members of this MPD group.
- $a_1, \ldots, a_n$, the MPD addresses of the members of this MPD group that are managed by the ZM associated with the scheduler.

For each MPD address $a_1, \ldots, a_n$, the scheduler then adds or modifies an entry in the scheduling table. For $1 \leq i \leq n$, if a row of the form $(a_i, g_i, l_i, e_i, s_i, b_i)$ is present in the table, it is updated to $(a_i, g, l, e_i, s_i, b_i)$, otherwise a row $(a_i, g, l, 0, 0, \texttt{false})$ is added.

The scheduler must then recalculate the ELQoS for each of the m entries in the table. First, it calculates the total level of LQoS demand, t:

$$t = \sum_{i=1, a_i \neq 0}^{m} \frac{1}{l_i} \tag{B.1}$$

It then updates the ELQoS for each of the entries in the scheduling table, e_i, with a new value, e'_i. If $t < 1$ (i.e. the total demand is less than the available location resources):

$$e'_i = \begin{cases} l_i & \text{if } a_i \neq 0 \\ \frac{1}{(1-t)} & \text{if } a_i = 0 \end{cases} \tag{B.2}$$

Otherwise, $t \geq 1$, and the effective location rates are scaled so that those demands can be met by the location system:

$$e'_i = \begin{cases} l_i \times t & \text{if } a_i \neq 0 \\ 0 & \text{if } a_i = 0 \end{cases} \tag{B.3}$$

Deleting a request

Messages containing an address, a_r, may be sent to the scheduler to indicate that any location request for that MPD should be deleted (entries corresponding to MPD address 0 may not be removed). If the scheduler finds that an entry for that MPD is present in the table, it removes the corresponding row $(a_r, g_r, l_r, e_r, s_r, b_r)$ from the table. The scheduler then updates every other row in the table $(a_x, g_x, l_x, e_x, s_x, b_x)$ to $(a_x, g_x, l_x, e_x, s'_x, b_x)$ where

$$s'_x = s_x - \frac{(s_x - s_r)}{n} \tag{B.4}$$

where n was the number of rows in the table *before* removal of the location request. The scheduler then recalculates the effective rates based on the remaining table entries as described in the previous section.

Allocating timeslots

Once every timeslot, the scheduler process must tell the ZM the address of the MPD that should be located next. It does this by examining the contents of the scheduling table—any attempts to update the table are blocked until this process is complete.

First, the scheduler steps through each row in the scheduling table. If the *lastgroup* register does not contain 0, and the scheduler encounters a row in which the MPD group ID is the same as the contents of the *lastgroup* register, and in which the *swapslot* bit is `false`, it sends the MPD address contained in that row to the ZM, and sets the *swapslot* bit in that row to `true`. This step swaps the positions of MPDs in the schedule to satisfy the constraint that members of the same MPD group should be located together, without altering the long-term rates at which individual MPDs are located.

Otherwise, the scheduler performs the following algorithm:

1. Choose the row in the table with the highest score, $(a_c, g_c, l_c, e_c, s_c, b_c)$. If more than one row shares the highest score, the row with the smallest MPD address is selected.

2. If b_c is `false`, for each row of the table that has the MPD group ID g_c set the *swapslot* bit to `false`. Then, update each row in the table $(a_x, g_x, l_x, e_x, s_x, b_x)$ to $(a_x, g_x, l_x, e_x, s'_x, b_x)$ where $s'_x = s_x + \frac{1}{e_x}$. Next, the single chosen row in the table is further updated to $(a_c, g_c, l_c, e_c, s''_c, b_c)$, where $s''_c = s'_c - 1$. The *lastgroup* register is set to g_c. Finally, the ZM is informed of the address a_c of the MPD that should be located next (if a_c is zero, this indicates that no MPD should be addressed in the next time slot).

3. If b_c is `true`, update each row in the table $(a_x, g_x, l_x, e_x, s_x, b_x)$ to $(a_x, g_x, l_x, e_x, s'_x, b_x)$ where $s'_x = s_x + \frac{1}{e_x}$. Next, the single chosen row in the table is further updated to $(a_c, g_c, l_c, e_c, s''_c, \texttt{false})$ where $s''_c = s'_c - 1$. The row in the scheduling table with the highest score is chosen again, and the above steps are retaken.

Calculating sleep times

The score, s, and ELQoS, e, values associated with the MPD selected by the scheduling algorithm to be addressed in a timeslot can be used to estimate a time for which that MPD may go to sleep. Assuming the LQoS demands made of the system do not change, it can be shown that the MPD will not be addressed in *at least* the next st timeslots, where:

$$st = \begin{cases} \lfloor -s \times e \rfloor & \text{if } s < 0 \\ 0 & \text{otherwise} \end{cases} \tag{B.5}$$

Scheduler behaviour

A mathematical proof may be developed to show that the long-term behaviour of the scheduling algorithm (when the total resource demand is equal to the system capacity) is to match the LQoS demand for each MPD. The full proof is omitted from this appendix for reasons of brevity, but an outline is sketched below:

- From the initial scheduler state, it may be shown that the timeslot allocation rules and request addition/deletion rules ensure that the sum of the scores in the scheduling table is equal to zero after each round of the algorithm.

- The previous observation implies that an MPD will only be allocated a timeslot when its score is greater than or equal to zero. In turn, an MPD's score must lie in the range $(-1, n-1)$, where n is the number of MPDs managed by the scheduler. The request addition/deletion rules preserve these bounds on the score.

- Since an MPD's score has an upper bound, and increases in the absence of timeslot allocation, no MPD can be ignored by the scheduler.

- The sleep time formula of Equation B.5 can be obtained as a corollary of the previous observations.

- Since the score is bounded, the difference between an MPD's score at distinct times must also be bounded. From this observation, it can be shown that in the long-term, the MPD is, on average, located every e timeslots, where e is the ELQoS of the MPD. If the ELQoS is equal to the LQoS (i.e. the location system is not overloaded), the MPD will therefore be located at the requested rate. Timeslot swapping complicates this analysis somewhat, but the result remains unchanged.

Bibliography

[Abowd97] Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R., Pinkerton, M. *Cyberguide: A Mobile Context-Aware Tour Guide*. Wireless Networks, Vol. 3, No. 5, October 1997. pp. 421–433 (*cited on page 137*)

[Acton74] Acton, W. *The Effects of Industrial Airborne Ultrasound on Humans*. Ultrasonics, May 1974. pp. 124–128 (*cited on pages 76, 77*)

[Adams93] Adams, N., Gold, R., Schilit, B., Tso, M., Want, R. *An Infrared Network for Mobile Computers*. Proceedings of the USENIX Mobile & Location-Independent Computing Symposium, Cambridge, MA, August 1993. pp. 41–51. (*cited on page 6*)

[Addlesee97] Addlesee, M., Jones, A., Livesey, F., Samaria, F. *The ORL Active Floor*. IEEE Personal Communications Magazine, Vol. 4, No. 5, October 1997. pp. 35–41 (*cited on page 22*)

[Asher97] Asher, R. *Ultrasonic Sensors*. Institute Of Physics Publishing, 1997. (*cited on pages 24, 32*)

[Azuma97] Azuma, R. *A Survey of Augmented Reality*. Presence: Teleoperators and Virtual Environments, Vol. 6, No. 4, August 1997. pp. 355–385 (*cited on pages 11, 16*)

[Bacon97] Bacon, J., Bates, J., Halls, D. *Location-Oriented Multimedia*. IEEE Personal Communications Magazine, Vol. 4, No. 5, October 1997. pp. 48–57 (*cited on page 8*)

[Banks89] Banks, K. *DATATRAK Automatic Vehicle Location and Position Reporting System.* Proceedings of the 1st Vehicle Navigation and Information Systems Conference (VNIS'89), Toronto, October 1989. pp. 214–218 (*cited on page 19*)

[Bass85] Bass, H., Bolen, L. *Ultrasonic Background Noise in Industrial Environments.* Journal of the Acoustic Society of America, Vol. 78, No. 6, December 1995. pp. 2013–2016 (*cited on page 138*)

[Baudel93] Baudel, T., Beaudouin-Lafon, M. *CHARADE: Remote Control of Objects using Free-Hand Gestures.* Communications of the ACM, Vol. 36, No. 7, July 1993. pp. 28–35 (*cited on page 137*)

[Beadle97] Beadle, P., Maguire, G., Smith, M. *Using Location and Environment Awareness in Mobile Communications.* Proceedings of the EEE/IEEE International Conference on Information, Communications and Signal Processing, Singapore, September 1997. pp. 1781–1785 (*cited on page 6*)

[Bellotti93] Bellotti, V., Sellen, A. *Design for Privacy in Ubiquitous Computing Environments.* Proceedings of the Third European Conference on Computer Supported Cooperative Work (ECSCW'93), (G. de Michelis, C. Simone & K. Schmidt, Eds.), Kluwer:Dordrecht. pp. 77–92 (*cited on page 138*)

[Bhatnagar93] Bhatnagar, D. *Position Trackers for Head Mounted Display Systems: A Survey.* Technical Report 93-010, Department of Computer Science, University of North Carolina, Chapel Hill, NC, 1993. (*cited on page 16*)

[Bishop84] Bishop, G., Fuchs, H. *The Self-Tracker: A Smart Optical Sensor on Silicon.* Proceedings of the 1984 Conference on Advanced Research in VLSI, Cambridge, MA, January 1984. pp. 65–73 (*cited on page 18*)

[Bohn88] Bohn, D. *Environmental Effects on the Speed of Sound.* Journal of the Audio Engineering Society, Vol. 36, No. 4, April 1988. pp. 223–231 (*cited on page 35*)

[Brown97] Brown, P., Bovey, J., Chen, X. *Context-Aware Applications: From the Laboratory to the Marketplace.* IEEE Personal Communications Magazine, Vol. 4, No. 5, October 1997. pp. 58–64 (*cited on page 9*)

[Bryson92] Bryson, S. *Measurement and Calibration of Static Distortion of Position Data from 3D Trackers.* Proceedings of SPIE, Vol. 1669, 1992. pp. 244–255 (*cited on page 21*)

[Challis95] Challis, J. *A Procedure for Determining Rigid Body Transformation Parameters.* Journal of Biomechanics, Vol. 27, No. 6, 1995. pp. 733–737 (*cited on page 43*)

[Christ93] Christ, T., Godwin, P, Lavigne, R. *A Prison Guard Duress Alarm Location System.* Proceedings of the 1993 IEEE International Carnahan Conference on Security Technology, Ottowa, October 1993. pp. 106–116 (*cited on page 20*)

[Davis81] Davis, R., Foote, F., Anderson, J., Mikhail, E. *Surveying, theory and practice.* Sixth ed., McGraw-Hill, 1981. (*cited on page 59*)

[Doussis93] Doussis, E. *An Ultrasonic Position Detecting System for Motion Tracking in Three Dimensions.* PhD Thesis, Tulane University, July 1993. (*cited on pages 27, 40*)

[Elrod93] Elrod, S., Hall, G., Costanza, R., Dixon, M., Rivieres, J. *Responsive Office Environments.* Communications of the ACM, Vol. 36, No. 7, July 1993. pp. 84–85 (*cited on pages 8, 138*)

[Engelson92] Engelson, S. *Active Place Recognition using Image Signatures.* Proceedings of SPIE, Vol. 1828, 1992. pp. 393–404 (*cited on page 18*)

[Ernst80] Ernst, B. *VHF and UHF Doppler Direction Finders for Position Finding in Built-up Areas.* News from Rohde & Schwarz, Vol. 20, No. 91, 1980. pp. 26–30 (*cited on page 20*)

[Esperança97] Esperança, C., Samet, H. *Orthogonal polygons as bounding structures in filter-refine query processing strategies.* Advances in Spatial Databases—5th Symposium, SSD'97, (M. Scholl & A. Voisard, Eds.), Lecture Notes on Computer Science 1262, Springer-Verlag, Berlin, 1997. pp. 197–228 (*cited on page 114*)

[FCC96] Federal Communications Commission *Guidelines for Evaluating the Environmental Effects of Radiofrequency Radiation.* Report no. FCC96-326, August 1996. (*cited on page 76*)

[Feiner93] Feiner, S., MacIntyre, B., Seligmann, D. *Knowledge-Based Augmented Reality.* Communications of the ACM, Vol. 36, No. 7, July 1993. pp. 52–62 (*cited on page 27*)

[Feng94] Feng, L., Borenstein, J., Everett, H. *"Where am I?" - Sensors and Methods for Autonomous Mobile Robot Positioning.* University of Michigan Technical Report No. UM-MEAM-94-21, December 1994. (*cited on pages 16, 21, 25*)

[Ferrin91] Ferrin, F. *Survey of Helmet Tracking Technologies.* Proceedings of SPIE, Vol. 1456, 1991. pp. 86–94 (*cited on page 16*)

[Feuerstein89] Feuerstein, M., Pratt, T. *A Local Area Position Location System.* IEE Conference Publication No. 315, 1989. pp. 79–83 (*cited on page 19*)

[Figueroa94] Figueroa, F., Mahajan, A. *A Robust Navigation System for Autonomous Vehicles using Ultrasonics.* Control Engineering Practice, Vol. 2, No. 1, 1994. pp. 49–59 (*cited on pages 27, 40*)

[Finney96] Finney, J., Davies, N. *FLUMP: The FLexible Ubiquitous Monitor Project.* Internal report MPG-96-18, Lancaster University, 1996. (*cited on page 8*)

[Fitzmaurice93] Fitzmaurice, G. *Situated Information Spaces and Spatially Aware Palmtop Computers*. Communications of the ACM, Vol. 36, No. 7, July 1993. pp. 38–49 (*cited on page 136*)

[Foley90] Foley, J., van Dam, A., Feiner, S., Hughes, J. *Computer Graphics: Principles and Practice*. Second ed., Addison Wesley, 1990. (*cited on pages 100, 106, 110*)

[Gallant90] Gallant, A. R. *nlmdl Statistical Methods Package*. Available via anonymous ftp at `lib.stat.cmu.edu` in directory `general/Nlmdl`. (*cited on page 56*)

[Getting93] Getting, I. *The Global Positioning System*. IEEE Spectrum, Vol. 30, No. 12, December 1993. pp. 36–47 (*cited on page 19*)

[Glantz90] Glantz, S., Slinker, B. *Primer of Applied Regression and Analysis of Variance*. McGraw-Hill, 1990. (*cited on pages 38, 62*)

[Güting94] Güting, R. *An Introduction to Spatial Database Systems*. VLDB Journal, Vol. 3, 1994. pp. 357–399 (*cited on pages 105, 111*)

[Harter94] Harter, A., Hopper, A. *A Distributed Location System for the Active Office*. IEEE Network, Special Issue on Distributed Systems for Telecommunications, Vol. 8, No. 1, January 1994. pp. 62–70 (*cited on pages 6, 7, 8, 14*)

[Hayton96] Hayton, R. *OASIS: An Open Architecture for Secure Interworking Services*. PhD Thesis, Cambridge University, 1996. (*cited on page 110*)

[Hodges94] Hodges, S., Louie, G. *Towards the Interactive Office*. Proceedings of SIGCHI'94, Boston, MA, April 1994. pp. 305–306 (*cited on page 137*)

[IDA96] Institute for Defense Analyses *Review of Virtual Environment Interface Technology*. IDA Paper P-3186, Institue for Defense Analyses, Alexandria, VA, 1996. (*cited on page 16*)

[IEEE98] Various authors. *Wireless Geolocation Systems and Services.* IEEE Communications Magazine, Vol. 36, No. 4, April 1998. (*cited on page 20*)

[Intersense98] Intersense, Inc. *IS-900CT Preliminary Data Sheet.* Burlington, MA., 1998. (*cited on page 27*)

[IRPA84] International Non-Ionizing Radiation Committee of the International Radiation Protection Association *Interim Guidelines on Limits of Human Exposure to Airborne Ultrasound.* Health Physics, Vol. 46, No. 4, April 1984. pp. 969–974 (*cited on pages 76, 77*)

[Ishii84] Ishii, M., Sakane, S., Kakikura, M. *A New 3D Sensor for Teaching Robot Paths and Environments.* Proceedings of the 4th International Conference on Robot Vision and Sensory Controls, London, October 1984. pp. 155–164 (*cited on page 17*)

[James95] James, R., Mendola, J. *Ultra-wideband Technology for Vehicle-to-Vehicle Communication and Sensing.* Proceedings of SPIE, Vol. 2591, 1995. pp. 108–112 (*cited on page 150*)

[Janin94] Janin, A., Zikan, K., Mizell, D., Banner, M., Sowizral, H. *A Videometric Head Tracker for Augmented Reality Applications.* Proceedings of SPIE, Vol. 2351, 1994. pp. 308–315 (*cited on pages 17, 24*)

[Kaiser95] Kaiser, U., Steinhagen, W. *A Low-Power Transponder IC for High-Performance Identification Systems.* IEEE Journal of Solid-State Circuits, Vol. 30, No. 3, March 1995. pp. 306–310 (*cited on page 14*)

[Kirsch97] Kirsch, D., Starner, T. *The Locust Swarm: An Environmentally-powered Networkless Location and Messaging System.* Technical Report No. 430, MIT Media Lab, 1997. (*cited on page 6*)

[Lamming94] Lamming, M., Flynn, M. *"Forget-me-not"—Intimate Computing in Support of Human Memory.* Proceedings of

FRIEND21, International Symposium on Next Generation Human Interface, Meguro Gajoen, Japan, 1994. (*cited on page 9*)

[Lanzl98] Lanzl, C., Webb, J. *Position Location Finds Applications.* Wireless System Design, June 1998. p. 55 (*cited on page 20*)

[Leick95] Leick, A. *GPS Satellite Surveying.* Wiley, 1995. (*cited on page 16*)

[Lin97] Lin, J. *Biological Aspects of Mobile Communication Fields.* Wireless Networks, Vol. 3, No. 6, 1997. pp. 439–453 (*cited on page 76*)

[Logitech91] Logitech, Inc. *Logitech 2-D/6-D Mouse Technical Reference Manual.* Fremont, CA., 1991. (*cited on page 27*)

[Lynnworth89] Lynnworth, L. *Ultrasonic Measurements for Process Control: Theory, Techniques, Applications.* Academic Press, 1989. (*cited on page 23*)

[Mäkynen95] Mäkynen, A., Kostamovaara, J., Myllylä, R. *Laser-radar-based Three Dimensional Sensor for Teaching Robot Paths.* Optical Engineering, Vol. 34, No. 9, September 1995. pp. 2596–2602 (*cited on page 19*)

[Mitchell98] Mitchell, J. *Shortest Paths and Networks.* Handbook of Discrete and Computational Geometry, (J. Goodman & J. O'Rourke Eds.), CRC Press, 1997. (*cited on pages 112, 117*)

[Mulder94] Mulder, A. *Human Movement Tracking Technology: Resources.* Addendum to Technical Report 94-1, School of Kinesiology, Simon Fraser University, July 1994. (*cited on page 16*)

[Murphy95] Murphy, J. *Tracking and Location Technologies for the Criminal Justice System.* Proceedings of SPIE, Vol. 2497, 1995. pp. 135–144 (*cited on page 19*)

[Myers90] Myers, R. *Classical and Modern Regression with Applications.* PWS-KENT, 1990. (*cited on pages 36, 62*)

[Mynatt97] Mynatt, E., Back, M., Want, R., Frederick, R. *Audio Aura: Light-Weight Audio Augmented Reality.* Proceedings of the 4th Annual Symposium on User Interface Software and Technology (UIST '97), Banff, October 1997. pp. 211–212 (*cited on page 9*)

[NATO91] NATO Navstar GPS Technical Support Group *Technical Characteristics of the Navstar GPS.* June 1991. (*cited on page 152*)

[Nelson98] Nelson, G. *Context-Aware and Location Systems.* PhD Thesis, Cambridge University, 1998. (*cited on pages 111, 118*)

[NIMA96] National Imaging and Mapping Agency *Naval Aviation Mapping, Charting and Geodesy Handbook.* National Imaging and Mapping Agency, St. Louis, MI, 1996. (*cited on page 16*)

[Nixon98] Nixon, M., McCallum, B., Fright, R., Price, B. *The Effects of Metals and Interfering Fields on Electromagnetic Trackers.* Presence: Teleoperators and Virtual Environments, Vol. 7, No. 2, April 1998, pp. 204–218 (*cited on page 21*)

[NTIS94] National Technical Information Service *Federal Radionavigation Plan 1994.* National Technical Information Service, Springfield, VI, 1994. (*cited on page 19*)

[OMG91] Object Management Group, *The Common Object Request Broker: Architecture and Specification.* Revision 1.1, OMG Document Number 91.12.1, December 1991. (*cited on page 121*)

[Paradiso96] Paradiso, J. *The Interactive Balloon: Sensing, Actuation and Behaviour in a Common Object.* IBM Systems Journal, Vol. 35, No. 3–4, 1996. pp. 473–487 (*cited on page 149*)

[Pinhanez97] Pinhanez, C., Bobick, A. *Intelligent Studios: Modelling Space and Action to Control TV Cameras.* Applied Artificial Intelligence, Vol. 11, No. 4, June 1997. pp. 285–305 (*cited on page 145*)

[ProWave89] Pro-Wave Electronics Corporation, *Air Ultrasonic Ceramic Transducers*. Taipai, Taiwan, 1989. (*cited on page 77*)

[Raab79] Raab, F., Blood, E., Steiner, T., Jones, H. *Magnetic Position and Orientation Tracking System*. IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-15, No. 5, September 1979. pp. 709–718 (*cited on page 20*)

[Rekimoto98] Rekimoto, J. *Matrix: A Realtime Object Identification and Registration Method for Augmented Reality*. To appear in Proceedings of the Asia Pacific Computer Human Interaction Conference (APCHI'98), Kanagawa, July 1998. (*cited on page 17*)

[Richardson94] Richardson, T., Mapp, G., Bennett, F., Hopper, A. *Teleporting in an X Window System Environment*. IEEE Personal Communications Magazine, Vol. 1, No. 3, Third Quarter 1994. pp. 6–12 (*cited on page 8*)

[Richardson98] Richardson, T., Stafford-Fraser, Q., Wood, K., Hopper, A. *Virtual Network Computing*. IEEE Internet Computing, Vol. 2, No. 1, 1998. pp. 33–38 (*cited on page 140*)

[Roberts66] Roberts, L. *The Lincoln Wand*. AFIPS Conference Proceedings, Vol. 29, 1966. pp. 223–227 (*cited on page 27*)

[Roberts75] Roberts, L. *ALOHA Packet System With and Without Slots and Capture*. Computer Communications Review, April 1975. (*cited on page 127*)

[Samfat95] Samfat, D., Molva, R. *A Method Providing Identity Privacy to Mobile Users during Authentication*. Mobile Computing Systems and Applications - Workshop Proceedings, IEEE, 1995. pp. 196–199 (*cited on page 14*)

[Schilit93] Schilit, B., Theimer, M., Welch, B. *Customizing Mobile Applications*. Proceedings of the USENIX Symposium on Mobile and Location-independent Computing, Cambridge, MA, August 1993. pp. 129–138 (*cited on page 7*)

[Schilit95a] Schilit, B., Adams, N., Want, R. *Context-Aware Computing Applications.* Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December 1994. pp. 85–90 (*cited on pages 2, 9*)

[Schilit95b] Schilit, B. *A System Architecture for Context-Aware Mobile Computing.* PhD Thesis, Columbia University, 1995. (*cited on page 7*)

[Sonnenberg88] Sonnenberg, G. *Radar and Electronic Navigation.* Butterworths, 1988. (*cited on page 19*)

[Sorensen89] Sorensen, B., Donath, M., Yang, G., Starr, R. *The Minnesota Scanner: A Prototype Sensor for Three-Dimensional Tracking of Moving Body Segments.* IEEE Transactions on Robotics and Automation, Vol. 5, No. 4, August 1989. pp. 499–509 (*cited on page 18*)

[Spreitzer93] Spreitzer, M., Theimer, M. *Providing Location Information in a Ubiquitous Computing Environment.* Operating Systems Review (ACM), Vol. 27, No. 5, December 1993. pp. 270–283 (*cited on page 138*)

[Spreitzer94] Spreitzer, M., Theimer, M. *Architectural Considerations for Scalable, Secure, Mobile Computing with Location Information.* Proceedings of the 14$^{\text{th}}$ IEEE International Conference on Distributed Computing Systems, Poznan, Poland, June 1994. pp. 29–39 (*cited on pages 7, 139*)

[Steele92] Steele, R. *Mobile Radio Communications.* Pentech Press, 1992. (*cited on page 91*)

[Steggles96] Steggles, P. *A Sensor Interpretation System for Personalisation and Mobility.* (Personal communication), The Olivetti and Oracle Research Laboratory, October 1996. (*cited on page 106*)

[Steggles98] Steggles, P., Webster, P., Harter, A. *The Implementation of a Distributed Framework to support 'Follow Me' Applications.* To appear in Proceedings of the 1998 International

Conference on Parallel and Distributed Processing Technique and Applications (PDPTA'98), Las Vegas, NV, July 1998. (*cited on page 130*)

[Varma90] Varma, H., Boudreau, H., Prime, W. *A Data Structure for Spatio-Temporal Databases.* International Hydrographic Review, January 1990. pp. 71–92 (*cited on page 118*)

[Walters97] Walters, R. *CTI In Action.* Wiley, 1997. (*cited on page 104*)

[Want92a] Want, R., Hopper, A., Falcão, V., Gibbons, J. *The Active Badge Location System.* ACM Transactions on Information Systems, Vol. 10, No. 1, January 1992. pp. 91–102 (*cited on pages 5, 8, 139*)

[Want92b] Want, R., Hopper, A. *Active Badges and Personal Interactive Computing Objects.* IEEE Transactions on Consumer Electronics, Vol. 38, No. 1, February 1992. pp. 10–20 (*cited on pages 5, 8*)

[Ward92] Ward, M., Azuma, R., Bennett, R., Gottscahlk, S., Fuchs, H. *A Demonstrated Optical Tracker with Scalable Work Area for Head-Mounted Display Systems.* Proceedings of the 1992 ACM SIGGRAPH Symposium on Interactive 3D Graphics, Cambridge, MA, March 1992, pp. 43–52 (*cited on page 18*)

[Weiser91] Weiser, M. *The Computer for the 21st Century.* Scientific American, September 1991. pp. 66–75 (*cited on page 1*)

[Wray94] Wray, S., Glauert, T., Hopper, A. *The Medusa Applications Environment.* Proceedings of the International Conference on Multimedia Computing and Systems, Boston, MA, May 1994. pp. 265–273 (*cited on page 144*)

[Wren96] Wren, C., Azarbayejani, A., Darrell, T., Pentland, A. *Pfinder: Real-Time Tracking of the Human Body.* Proceedings of SPIE, Vol. 2615, 1996. pp. 89–98 (*cited on pages 17, 137*)

[Yim88] Yim, W., Seireg, A. *Tracking Moving Objects in 3-D Space*. Computers in Mechanical Engineering, Vol. 7, No. 2, September/October 1988. pp. 8–17 (*cited on page 18*)

[Zimmerman94] Zimmerman, K., Cannon, R. *Differential Carrier Phase GPS Techniques for Space Vehicle Rendezvous*. Proceedings of Institute of Navigation GPS-94 Conference, Vol. 2 Salt Lake City, UT, September 1994. pp. 1693–1700 (*cited on page 20*)