Real Time Positioning and Motion Tracking for Simulated Clay Pigeon Shooting Environments

Final Project Report

Simon Coulson (sac99)

Project Supervisor: Tony Field

Abstract

This report details the design and implementation of a high accuracy point location and motion tracking system an its application to monitoring the position of a users shotgun in a simulated clay pigeon shooting environment. A thorough analysis of the issues surrounding the development of such a system is presented and a number of tracking methods and their merits are discussed.

An implementation using a stereographic approach with off-the-shelf CCD cameras and infrared point sources, coupled with analysis software is described. The implemented system is an accurate, versatile positioning system suitable for a number of positioning and tracking tasks.

An extension to the system has been produced to enable it to monitor the motion of a shooter's weapon in a simulated shooting environment. A novel way of calibrating the position of the projection screen is described, as are numerous additional features that help to enhance the simulation.

The performance and accuracy of the system are evaluated. The accuracy of the system was found to be comparable to many commercial positioning systems. Opportunities for further work and use of the system are also discussed.

Acknowledgements

Firstly I'd like to thank Dr. Tony Field for his help, advice and enthusiasm during the project. A special mention must also go Jay Jayasundera and everybody else in CSG who helped me to buy, borrow and steal all the equipment I needed to get the project off the ground.

A big thank you to Roland Smith in Physics for his invaluable advice on optics. Thanks to Philip Proctor in Mechanical Engineering for helping me with the mount for the cameras when everybody else had told me they were too busy. Thanks also to Asa Denton for endless discussions and heated arguments about the project.

Finally, thank you to beer.

Contents

List of Figures

Chapter 1

Introduction

> I always keep a supply of stimulant handy in case I see a snake–which I also keep handy.
>
> W. C. Fields (1880 - 1946)

1.1 Clay Pigeon Shooting

There's something intrinsically appealing about the concept of shooting. Perhaps it is the challenge involved, or maybe it just satisfies a primal hunting instinct. Whatever the cause, millions of people participate in shooting, be it in the field, on a range or even in a computer game.

Clay pigeon shooting involves the participant firing a shotgun at a disc of clay launched into the air. With sufficient skill, the shooter can track, fire at and hit the target. As the participant becomes more proficient, the distance to the target can become greater, the clay can move more quickly and several clays can be launched at once.

Teaching somebody how to shoot is a complex task and requires patience, experience and a keen eye. If the shooter hits the target it is quite apparent - the clay breaks and the shooter has instant feedback on the quality of their shot. Things become more complex if the shot misses the target. How did it miss? Was the shot fired too early or too late? Was the participant aiming too high or too low?

In reality it is often difficult to answer these questions, as the pellet cloud moves at such high velocity that is nearly invisible to the human eye. To make the task of analysing a shooter's performance easier, a shooting environment can be mimicked using a computer.

Figure 1.1: Teacher and student, clay pigeon shooting in the field

1.2 Computer Simulation

Simulation :- *"Representation of the operation or features of one process or system through the use of another."*

In computer simulation, the system to be simulated is the real world shooting environment, and the simulating system is the computer. Provided that the simulation is well devised, close approximations can be made to the real world environment whilst removing much of the complexity.

Simulation is a good solution to the problem of determining what happened to the shot once the trigger was pulled. As the computer calculates where the pellet cloud travels it is possible to find out what happened if the shot missed. Using a computer also ensures that the participant does not have to brave wind, rain and mud to practice their sport. There's only one problem - it's difficult to convince a hardened clay pigeon enthusiast that a swivel chair and the Enter button is any substitute for the real thing.

The interface between the user and the simulation needs to be realistic. As well as accurately modelling the physics of the shooting environment, we must also interface the shooter in the most natural way possible. We could employ convincing graphics and sound, but if the interface is via the keyboard and mouse the simulation is little better than a computer game and will not enable the user to improve as a clay pigeon shooter.

Dispensing with conventional input devices has it's drawbacks. How can the simulation get the information that is needed to determine when the gun is fired, and more importantly in which direction it was fired? Accurate simulation will depend on three equally important pieces of information:

1. The location of the shooter in the environment.
2. The location and orientation of the shotgun in the environment when the shot is fired.
3. The velocity of the gun when the shot is fired.

The first piece of information is necessary to ensure that the scene is accurately represented. For instance, if the user moves their head to the right, the projected scene should also shift to the right. The latency of the user moving their head and the projected scene being updated is a crucial issue. If there is too much delay between head movement and image update, nausea can result (referred to as "simulator sickness" [20]). This style of user based projection is known as "head slaved" projection and updates on the position of the head need to be constantly acquired.

The location and direction of the shotgun will determine the path that the pellet cloud travels. If the shotgun is moving when the shot is fired this will cause the pellet cloud to posses a component of that motion, so the velocity of the gun is also needed.

With this information, the simulation software can calculate what path the pellet cloud will follow and if the target will be hit. By replaying the simulation at a much reduced speed, the user will be able to find out where the shot travelled and use this information to improve their technique.

1.3 Scope

The main objective of the project is to develop a tracking system suitable for monitoring the position of a clay pigeon shooter and their shotgun within a given simulation environment. There a number of factors that need to be considered when implementing such a system.

- Accuracy
- Latency
- Refresh Rate
- Portability
- Flexibility
- Cost

Of these, the most fundamental issue is accuracy. In a real world shooting environment, the range involved can be in excess of 50 metres. If the shooters aim is out by only a few millimetres they may miss the target by up to a metre. Any tracking system must be able to resolve the position and direction of the gun to within an acceptable limit.

A low latency and high refresh rate are needed for smooth and responsive simulation. A high latency will result in noticeable lag in the simulation, whilst a low refresh rate will result in erratic animation.

Portability and flexibility are both desirable, though not essential for the development of a prototype system. The cost of the system should be competitive with tracking systems of a similar performance.

1.4 Reading Guide

This report will cover a large range of information, spanning fields from optics to trigonometry. An overview of each chapter is as follows:

Chapter 2 lays out a detailed specification for the tracking system.

Chapter 3 includes background research, a discussion on viable tracking methodologies and analysis of commercial tracking systems and shooting simulators. The chapter concludes with the selection of a method for the implementation.

Chapter 4 discusses of the hardware requirements of an electro-optical tracking system.

Chapter 5 explains the importance of calibration and how it is performed.

Chapter 6 describes the software component of the implemented tracking system.

Chapter 7 deals with the issues involved in adapting the tracking system for use in a clay pigeon shooting simulation package.

Chapter 8 comprises an evaluation of the results of tests on accuracy and performance of the system.

Chapter 9 is a conclusion on the effectiveness of the work carried out and recommendations for future work.

Appendix A contains tables including all of the information gathered during testing.

Appendix B gives details on where to obtain the project source code and how to compile it.

Appendix C is a quick reference guide for those interested in running the tracking system.

Chapter 2

Specification of the Project

Good plans shape good decisions. That's why good planning helps to make elusive dreams come true.

The Nine Master Keys of Management
Lester R. Bittel

The aim of the project is to produce a tracking system suited to monitoring a shooter and his shotgun in a simulated clay pigeon shooting environment. There will be a hardware component that gathers information about the objects being tracked and a software component that calculates the actual position of the objects based on that information.

In particular, the system should deliver regular updates on the following information:

A0 The position of the shooter's eye.

A1 The position of the end of the gun barrel.

A2 The direction that the gun is pointing.

Additionally the system should also be able to:

A3 Determine when the trigger is pressed.

This information is essential to providing an accurate simulation and the tracking system must fulfil all of the aims A0 to A3. In addition, for the tracking system to be effective for use in accurate simulation, the following requirements should also be met:

B0 Inaccuracies of only a few millimetres will be amplified over the simulated distances of the shooting range. For this reason, a spatial accuracy of $\pm$

10mm or better is required for the location of the shooter and his shotgun. Assuming that the system was at the outside of this limit, over a simulated shooting distance of 50 metres the path of the shot at the target would be approximately 50 cm out.

B1 The shooter should be free to move around the environment, but some constraints have to be introduced. For the purposes of this project, it can be assumed that the shooter and his gun are free to move within a specified bounding box - initially three metres in size.

B2 The update rate needs to be as high as possible, for smooth animation, a rate of not less than 20 Hz is required. Although this figure may be difficult to achieve without appropriate hardware.

B3 The latency between acquiring positioning information and calculating the results needs to be less than 4 milliseconds to avoid "simulator sickness" [20].

B4 Simulation of the scene and implementation of the ballistics models are being worked on by another student [8]. Information generated by the positioning equipment is needed in this simulation, so a method of communicating the results between the simulator and positioning software will have to be negotiated.

B5 The system should cope with occlusion of target objects. This could be either by reporting the target as missing, or by predicting its position based upon its previous movement.

Aims A0 - A3 together with requirements B0 - B4 define a minimum specification for the project to be effective for its intended purpose. There are a number of additional features that the tracking system would benefit from:

C0 Whilst accuracy and responsiveness are important these must be balanced against the cost of the system. The cost of the system should be considerably less than comparable commercial alternatives.

C1 When in use, the system should feel as unobtrusive as possible. The shooter should be abe to use their own equipment, and obvious physical additions and trailing cables should be kept to a minimum.

C2 The system should be safe to use. For example, any energy emissions (such as infrared radiation) should be kept within safe limits.

C3 The system should be portable. Not only should the equipment used in the system be easily moved, but the system should not be dependent on a particular location to function properly.

C4 The visual impact of the tracking system on the simulation environment should be minimal. This may involve using infrared light instead of visible light.

C5 When moved to a new environment the system should be able to recalibrate important data automatically.

If time allows, and assuming all of A0 - A3 and B0 - B4 have been satisfied, there are several possible ways that the project could be extended:

E0 A mouse driver could be developed to enable the shotgun to be used as a mouse on the simulation computer. The user points the mouse by aiming the gun and the trigger acts as the left mouse button.

E1 Once the system has been developed, it may be possible to extend it into a generic positioning and tracking system with uses in other fields of study.

E2 If the accuracy of the system is high enough, contour mapping of three dimensional objects may be possible. This would involve finding the position of regular points on the surface of a static object and constructing a three dimensional geometric representation of the object from the information.

Chapter 3

Background

> The trouble with facts is that there are so many of them.
>
> *The Gentle Reader*
> Samuel McChord Crothers

3.1 Shotguns, Cartridges and Clays

3.1.1 The shotgun

A typical shotgun used for clay pigeon shooting is between 0.8 metres and 1 metre in length and fires ammunition known as shells or cartridges. It can come in either a single barrelled, or the more popular double barrelled design. The twin barrels can be arranged in a "side by side" or "over and under" fashion and are fired separately using one or two triggers.

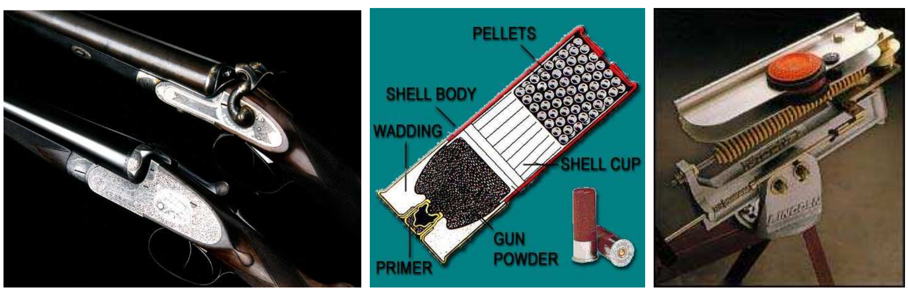


Figure 3.1: Antique shotguns, a cutaway shell and clay pigeons in the 'trap' ready for launch.

3.1.2 Cartridges

Shotgun cartridges contain a mass of small pellets collectively known as shot. When the gun is fired the shot leaves the barrel in a tightly packed cluster and fans out over distance to form a pellet cloud. Changing the size and number of pellets in the shell will alter the velocity, spread and power of the pellet cloud.

3.1.3 Clay Pigeons

A clay pigeon is a disc of compressed chalk and charcoal that is fired from a launcher. Its composition is such that it can survive the forces imparted by the launcher, but easily shatter when hit by the pellet cloud. The clay is designed to fly smoothly through the air. The size and shape of the clay affects the path of its flight.

3.1.4 Ballistics

Both the pellet cloud and the clay pigeon are subject to the laws of physics. Gravity, air resistance, temperature and humidity will all affect their flights. Accurately modelling ballistics is a complex research field. For the purposes of the simulation, a model developed by R.A. Giblin and D.J. Compton [11][10] will be used to calculate the pellet cloud dynamics. The flight of the clay pigeon will be modelled using simple Newtonian physics.

3.2 Conventional light gun technology

More realistic interfaces to shooting simulations have been in existence for a long time. Companies such as Nintendo have been producing systems known as lightguns for decades (figure 3.2). These work on a simple yet robust technology.

Inside the barrel of the light gun is a photodiode that senses the amount of light entering the gun. The user aims the gun at targets displayed on the screen. When the user presses the trigger on the gun, the screen is momentarily blanked out. Then the target on the screen is drawn in white, with the rest of the area left blank. This all happens in just two frames, so the user will only detect a blink of the screen. If the level of light entering the lightgun increases significantly between showing the blank screen and showing the white target, then it is assumed that the gun was aimed at the target.

This kind of technology is perfect for the home gaming market. The lightguns are cheap to produce, robust and surprisingly accurate. This method is not suitable for this project for two reasons. Firstly, it only provides analysis of a two dimensional environment, scene depth and the location of the user are not taken into account. Secondly, there is no way to use lightgun technology to

Figure 3.2: A screenshot of the classic Nintendo game Duck Hunt and a lightgun.

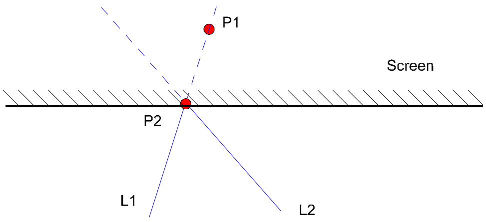


Figure 3.3: The problem with two dimensional shot analysis.

calculate the motion of the gun itself.

3.3 Three Dimensional Environments

The issue of depth in the simulation is an important one. Lightgun technology simply identifies where the shot would intersect the viewing plane. It does not provide information on which direction the gun was aimed. This is fine for two dimensional projection, but three dimensional analysis needs more information.

To provide an accurate simulation, we must provide as close an approximation to a real world environment as possible. This means we need to model a three dimensional environment and take the depth of the scene into account, which is difficult to represent on a two dimensional viewing surface.

With depth taken into account, lightgun technology becomes ineffective. In figure 3.3 the shot travelling along line L_1 will hit the target P_1, whilst a shot travelling along line L_2 will not - even though it will hit the projection of the target P_2.

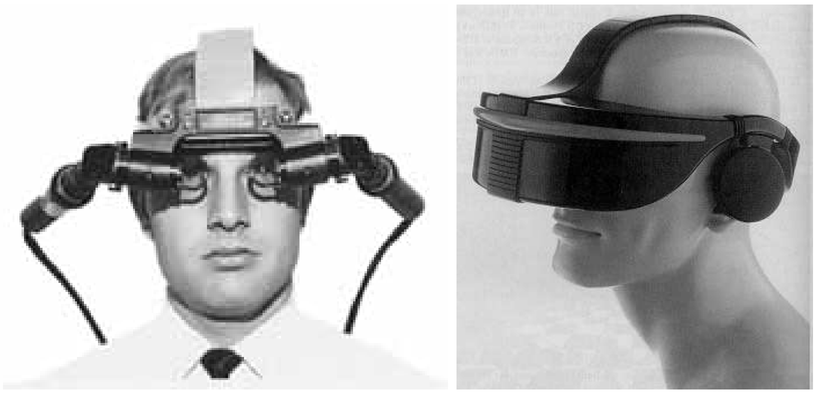


Figure 3.4: Sutherland's original head mounted display and a more recent version.

3.3.1 Virtual Reality

> *The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked.*
> Ivan Sutherland 1970

The term "Virtual Reality" was coined by Ivan Sutherland [22] in the late 1960's when he was an Associate Professor at Harvard. He and his student, Bob Sproull, continued work on a project called "Remote Reality" developed at the Bell Helicopter Company. This system utilised a head mounted display coupled with infrared cameras mounted on the underside of a helicopter. The cameras moved when the pilot's head moved and the system enabled the pilot to land their helicopter at night on rough terrain.

In 1966, Sutherland and Sproull replaced the cameras with generated computer graphics and devised the first Virtual Reality system. The Virtual Reality environment was no more than a wire-frame room with the compass directions North, South, East, and West initialed on the walls.

3.3.2 Head Coupled Virtual Reality

Since Sutherland and Sproull conducted their work into Virtual Reality the head mounted display system has become known as *immersive* Virtual Reality as it removes all other visual cues from the user, leaving them with only the computer generated images. Immersive headsets are costly to manufacture and often not practical for use in a simulation. Another approach is to use Head

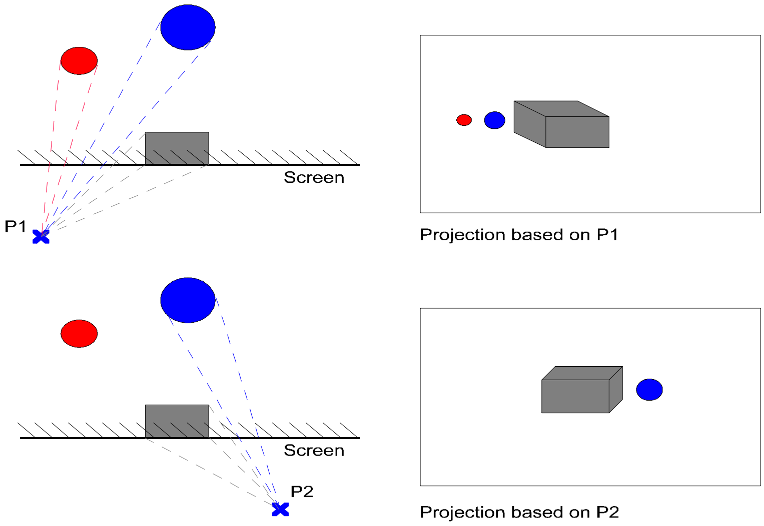


Figure 3.5: How changes in user position affect scene projection in a user centred simulation.

Coupled Virtual Reality, which uses conventional display technology such as monitors and projection screens.

This allows the user to see objects in the room around them and also virtual objects displayed on the screen. These objects are rendered correctly based upon the position of the users eye. The screen can be thought of as a window into another three dimensional world.

For this reason it is important that the projection of the scene is based around the location of the user. If the user moves their head, objects near the window will not move much in respect to the window, whereas objects far away such as a distant tree will appear to move much further (figure 3.5). This movement may even cause objects to be occluded in the view.

A major advantage of Head Coupled Virtual Reality is that it provides an Augmented Reality of the real world. The user can still see other objects in the room, such as the shotgun they are holding. If immersive Virtual Reality were used then the shotgun would have to be modelled inside the program. This modelling will make the interface to the simulation less realistic.

3.3.3 Stereoscopic Display

Stereoscopic display is a technique used to increase the realism of the two dimensional projections. Separate views are constructed for the right and left eyes and either mechanical or electronic glasses ensure that each eye receives the correct image. Stereoscopic display enhances realism by providing depth cues to the user. This is an important aspect in the simulation of clay pigeon

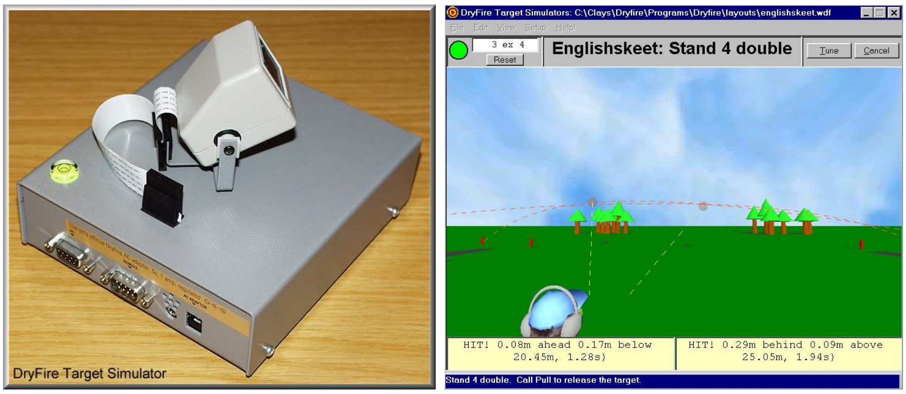


Figure 3.6: The DryFire shooting simulator and an image from its simulation software.

shooting where distance plays a crucial factor in determining where to aim the shot.

3.4 Commercial Shooting Simulators

There are several shooting simulators available on the open market. Some are for simple target practice, others are designed as training aids specifically for clay pigeon shooting. What follows is a brief examination of a number of the more advanced systems and a concluding discussion on the features that any implementation of a clay pigeon shooting simulation should strive to achieve.

3.4.1 DryFire

The strength of the DryFire [7] system (figure 3.6) is in the simulation software, which can simulate the use of a huge variety of shooting equipment. The user can select any combination of shotgun, clay and cartridge type from the simulation database and new models can be written by the user if needed. This enables a shooter to select the equipment that they would use in the field.

A laser dot is projected onto a screen or wall at a set distance of ten feet from the shooter. This is the target, and it arcs across the wall. The shooter tracks the target and fires, which projects a second laser dot showing where the gun was aiming. A camera is used to transmit an image of the wall back to the computer, which then determines if the shot intersected the target.

Figure 3.7: The pistol version of the CURT simulation package (left) and the LaserShot system in use.

3.4.2 CURT

CURT [12] is a computer based simulator that relies on a specialised rifle and target (figure 3.7). The rifle emits a beam of infrared light which strikes the target and scatters reflected light. Sensors measure the amount of reflected light and use this information to determine where the rifle is aiming at the target. The monitoring software provides an analysis of the movement of the shooters weapon across the target.

3.4.3 LaserShot

LaserShot [13] is one of the most graphically impressive shooting simulators on the market (figure 3.7). The simulated scene is projected onto a large screen, which the shooter aims at. When the trigger is pulled a pulse of laser light is fired at the screen. This light is detected by a camera, which compares the position of this point from each edge of the screen to calculate where the shot would hit. There is no concept of distance within the simulation as the projected scene is two dimensional.

3.4.4 Trojan ShotPro 2000

The Trojan [3] simulation system projects a simulated view onto a 180° wide screen (figure 3.8). The shooter stands at the centre of the scene and uses their own shotgun. The shotgun is loaded with a cartridge that fires infrared laser light down the barrel and onto the screen. Sensors track the movement of the shotgun as the infrared light moves over the projection screen. When a shot is fired, the light increases in intensity for a short time - which is picked up by the sensing equipment.

Once the user has finished shooting they can review their performance using the

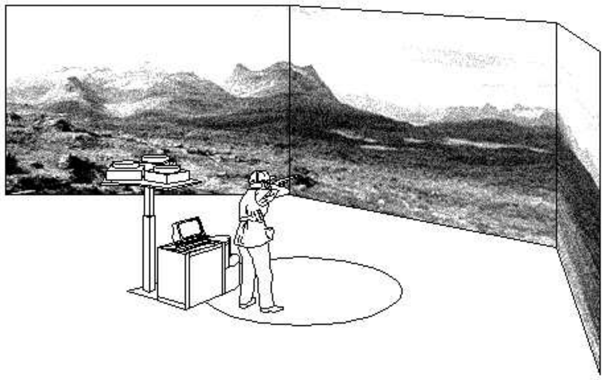


Figure 3.8: Sketch of the Trojan ShotPro simulator setup.

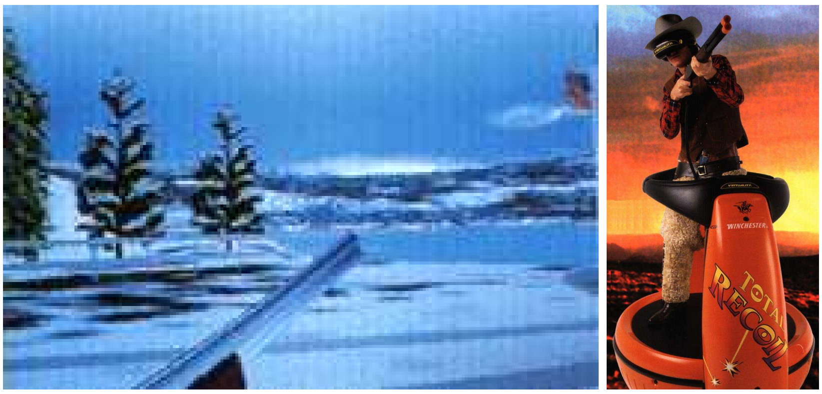


Figure 3.9: The Total Recoil simulator and a screen shot of the simulation software in action.

simulation software. It is possible to experiment with settings such as calibre and choke and various types of ammunition.

3.4.5 Total Recoil

The Total Recoil system [6] developed by the Winchester Rifle Company is a fully immersive Virtual Reality simulation (figure 3.9). The system uses a head mounted stereoscopic display and a rifle fitted with a modified version of the Polhemous tracking system (see section 3.7.1).

The simulation software uses a ballistics model developed by Winchester. The system's strongest asset lies in how it interfaces with the user. The participant can call out "pull" to launch a clay and when the trigger is squeezed, compressed gas is used to simulate the kick of the rifle. The system has won numerous awards within shooting circles for the quality of the simulation.

3.4.6 Discussion of Commercial Simulators

Most of the systems described above offer only two dimensional shot analysis in that the path of the shot is determined by the intersection of the line of the weapon with the viewing plane. Only Winchester's Total Recoil takes the actual position of the weapon into account, relying on the Polhemous Fastrak system for positioning information. This is a magnetic field based tracking system and so does not suffer from occlusion which is the main problem in biometric tracking.

At close ranges Fastrak is accurate to within a tenth of a millimetre with the accuracy decreasing sharply with distance. This places restrictions on where the user can move, as they need to remain within a radius of 50cm from the main tracking unit. Total Recoil imposes this restriction by placing the user in a stand, the user is free to point the gun in any direction but they can not move about the environment.

The immersive approach adopted by Total Recoil has one major disadvantage in that the shooter sees a computer generated version of the weapon and not the real thing. In addition, the head mounted display unit is bulky and tends to be obtrusive when the user is aiming a shot.

Visually, the Total Recoil, ShotPro and LaserShot systems are impressive. ShotPro's innovative 180° wide screen provides a highly immersive environment whilst still allowing the participant to use their own shotgun, however, the dimensions of the projection area needed restrict the flexibility of the system. Dryfire's software package allows users to select any combination of equipment, which is a desirable feature, but the simulation with a laser dot projected onto a wall lacks realism.

The pellet cloud ballistics model that will be used in this project is purported to be the most accurate model of pellet cloud dynamics available. All of the systems mentioned use older, more basic models. Some simply treat the pellet cloud as a sphere that increases with size as it travels. The accuracy of the ballistics model is a fundamental part of a good simulation.

3.5 Positioning and Motion Tracking

Positioning involves determining the location of an object relative to known reference points. This is usually achieved by detecting energy emissions from the target object with several detectors at known locations and using software to triangulate the position of the object relative to the detectors. The energy emissions can be anything from sound waves to gamma rays, although some are obviously more practical than others.

Positioning systems are referred to as being either active or passive. A passive tracking system relies on energy signals from the environment to make inferences about the location of target objects. An active system adds energy into the

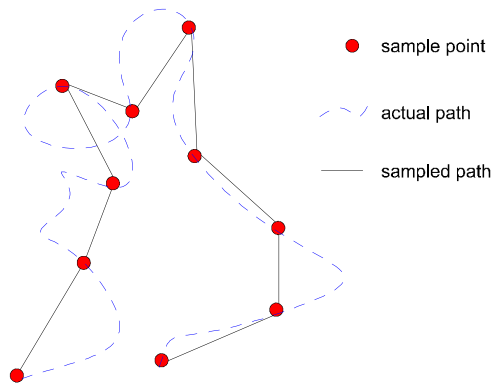


Figure 3.10: The effects of coarse subsampling on the estimation of object motion.

environment to simplify or speed up the locating process.

3.5.1 The Fourth Dimension

When the target is moving, it is necessary to regularly check the position over time. In general, the faster the object is moving, the more frequent the checks should be. A fast moving object does not require high frequency measurements if its velocity is constant as its position can be estimated from previous measurements. If the object changes its velocity rapidly, such as a bee gathering pollen then the checks need to more frequent as complex paths of motion will be flattened by this subsampling (figure 3.5.1).

3.6 Possible Methods for Motion Tracking

A number of methods have become popular for location and motion tracking applications. This section describes the merits of the most established methods.

3.6.1 Magnetic Field Interference

A base station emits a magnetic field through the simulation area. Sensors are placed on the objects that are to be tracked. Each sensor contains three orthogonal wire coils. A current is induced in these coils due to the magnetic field. From the current generated in each coil it is possible to determine the position and orientation of the sensor relative to the field.

Any ferrite materials (such as the barrel of a gun) will have an impact on the magnetic field and will thus produce inaccuracies in the results. Some electrical equipment (CRT monitors for example) also effects the field. Systems using this method can be accurate to within fractions of a millimetre at short ranges, although the accuracy drops off sharply at longer ranges. Most systems of this type are only effective within a range of a few metres at most.

A major benefit of such systems is that as the magnetic field can pass unaffected through non ferrite bodies, there is no problem with occlusion. This makes magnetic field interference methods particularly suitable for analysing the position and movement of the human body.

3.6.2 Acoustic

In their simplest form, acoustic positioning systems involves emitting a pulse of sound from the target and using three or more microphone sensors which record the time when they receive the sound. As the speed of sound is known, it is possible to work out the distance of the source from each sensor based on the time taken for the sound to reach them. The location of the source can be triangulated from these measurements.

An alternative approach is to emit sound from a base station and detect sound waves echoed back off the target objects. This technique is employed in SONAR. Sound can also be used to measure the velocity of the object by examining the Doppler shift of the transmitted pulse. Although not completely disrupted by occlusion, the sound waves will diffract around and produce echoes off any obstacles in their path. Accuracy is increased if very high frequency sound waves are used. This has the added advantage of being undetectable by the human ear. In practice the accuracy of such systems is limited to around a few millimetres.

3.6.3 Lasers

A laser is an intense, focused beam of light of a particular wavelength. By shining a laser at an object and detecting the reflected light it is possible to determine distances to a very high degree of accuracy (in the region of a few nanometres). The sensing equipment is complex, as the phase shift of the laser needs to be measured. By using three lasers fired from known points at the target it is possible to triangulate its position. Occlusion is an obvious problem here as the laser beam needs an unobstructed path to the target.

It is also possible to perform laser positioning using just a single laser beam if the direction of the beam is known. This is often achieved using a mirror on a precision adjustable mount. Complexity is added to the problem when the target begins to move, as the laser will have to track it.

As well as positioning single objects, lasers can be used to map out a set of points. By repeatedly scanning the laser beam across a target area, many measurements can be taken. This information can be used to build a model of the observed scene. Laser based tracking systems require precision engineering and are costly to implement.

3.6.4 Electro-Optical

Two or more cameras are used to observe the environment containing the objects to be tracked. The images received by the cameras are analysed to discover where the target object is in each image. From this information and information about the cameras themselves (such as their location in the environment) it is possible to determine the position of the targets using triangulation.

Finding where the target object lies in the images is a complex task, especially if it has to be carried out in real-time. To make the feature extraction process easier, highly distinguishable markers can be placed on the target objects. The accuracy of such systems depends on the resolution and placement of the cameras, and can be as high as 0.1 millimetres.

3.6.5 Mechanical

Mechanical tracking systems involve some kind of mechanical arm that is fixed to the target (often the human head). As the head moves, the arm moves with it so that no resistance is felt by the user. Mechanical measurements about the status of the arm, such as the angle of rotation of the joints, are analysed to find the position and orientation of the users head. The accuracy of such systems is determined by the precision of the engineering. Such systems are limited to tracking a single target and are restrictive in that the object must remain within a specified volume.

3.7 Commercial Tracking Systems

Tracking and positioning systems are used in a wide variety of applications, from art restoration to neurosurgery. The cost invariably increases with the accuracy and update rate of the system. Described below are three commercial tracking systems that rely on differing methods of positioning.

3.7.1 Polhemus Fastrak

The Fastrak [21] tracking system uses magnetic field interference (figure 3.11). It is useful for tracking objects in complex environments as it is not susceptible to occlusion. As well as measuring the position of it's sensors, it can also determine their orientation. The system refreshes at a rate of 120 Hz with a single sensor and can support up to four sensors (although refresh rate decreases).

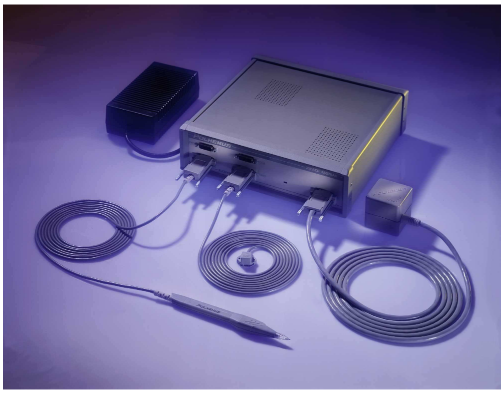


Figure 3.11: The Polhemus Fastrak system.

Figure 3.12: The NDigital Optotrak system.

Fastrak can compensate for distortions in the magnetic field caused by CRT monitors, which is unusual among positioning systems of this type. Polhemus claim it is the most accurate device of it's kind and can position objects to within ± 0.1mm at short ranges (less than 0.5 metres). This accuracy falls off sharply over larger distances due to the inverse square nature of the intensity of the magnetic field.

3.7.2 NDigital Optotrak

OptoTrak [18] is an electro-optical motion and position measurement system (figure 3.12). It comprises three high resolution CCD cameras and uses infra red markers which are attached to the objects to be tracked. Accuracy is given as ± 0.1mm. Most impressive is that the system is capable of tracking and uniquely identifying up to 256 individual markers. Optotrak is widely used in industry, universities and medical establishments around the world. It's excellent performance is reflected in the price, which is in the region of £50,000.

3.7.3 Shooting Star Technology ADL-1 Head Tracker

The ADL-1 [17] is a mechanical, head mounted positioning device. The user wears a head band, attached to which is a mechanical arm which can move through six degrees of freedom. High accuracy potentiometers are used to determine the location of the head relative to the base of the arm. The cost of such systems begins from £1,500.

3.7.4 Discussion of Tracking Alternatives

The possible methods of motion tracking described previously all have their merits. Laser tracking is by far the most accurate, but also costly and difficult to implement. Magnetic field interference methods benefit from their immunity to occlusion, whilst their accuracy is only guaranteed at close range. The Electro-optical method appears to be the most suitable option for this project as it can produce very good results without the need for precision engineering.

3.8 Proposed Implementation

The implementation will make use of an electro-optical approach to object location and tracking. This method is relatively inexpensive, requires no precision engineering and the equipment required is readily available.

There are three main stages in producing any positioning system:

- Equipment selection and placement,
- Calibration of equipment,
- Data collection and analysis.

To implement an effective electro-optical positioning system, at least two cameras will be required. This will give a binocular view of the target area. As long as the cameras are located at different points it is possible to triangulate the position of a target object. The accuracy and responsiveness of the system will depend on the location of the cameras, their resolution and their refresh rates.

Calibration of the system is essential to ensure a high level of accuracy. The aim of calibration is to produce a mathematical model for a camera that relates the real world to the image produced by the camera.

Once the system has been set up, image data needs to be acquired and analysed in software. To do this the cameras will need to interface with a computer. The most common way of doing this is through a video capture board (VCB).

To reduce the complexity of the feature extraction software, easily distinguishable markers can be used to mark the targets. Infrared (IR) markers would be invisible to the user, but highly visible to certain digital cameras (in particular black and white CCD cameras). Using filters it is be possible to remove unwanted light from the visible spectrum, making the task of locating the targets much easier.

The complete tracking system will have two major components:

The Data Acquisition Hardware which will consist of two CCD cameras communicating with a computer via a VCB. Discreet, but highly visible infrared markers will have to be devised. A mount for the cameras will also need to be produced.

The Analysis Software which will examine images acquired from the two cameras and (based upon the location of the cameras) determine the location of any markers detected in the images.

Chapter 4

Hardware

Reminds me of my safari in Africa. Somebody forgot the corkscrew and for several days we had to live on nothing but food and water.

W. C. Fields (1880 - 1946)

For this implementation of an electro-optical tracking system, two cameras will be used to capture images from the environment. The target objects for location and tracking will be tagged with small infrared markers. Both cameras shall be attached to a solid mount to increase portability. This chapter examines the major issues associated with the hardware components of the implementation.

4.1 Features of Black and White CCD Cameras

CCD is an acronym for Charged Coupled Device. A CCD is an array of photosensitive cells that can measure the intensity of light. Black and white CCD cameras are capable of detecting light in the infrared region of the spectrum, as well as visible light. Coupled with a filter it is possible to make them sensitive to only infrared light. Although a CCD is a digital sensor array, the output from a typical CCD camera is an analogue signal which can be viewed on a standard television set.

For this project, two Inter-M VBC272 monochrome CCD cameras will be used to capture stereoscopic views of the environment. These cameras have a vertical scan rate of 50Hz. However, the effective frame rate is only 25Hz however, as the images are interlaced. This means that the camera first reads the odd lines and then the even lines of the image. If the camera is viewing a moving scene, then ghosting can result. This occurs when an object is in one location when the odd lines are scanned, but moves to a different location by the time the even lines are scanned (figure 4.1). This can fragment objects in the image making

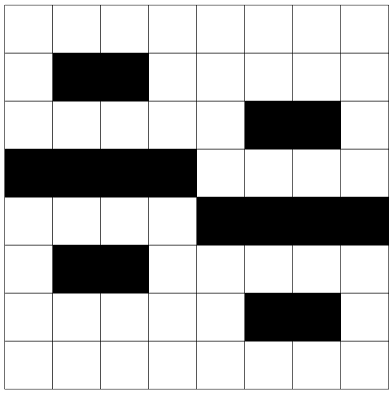


Figure 4.1: A ghosted marker in an image. The fragmentation is caused by the interleaved scanning of the camera sensor elements.

them harder to locate.

4.2 Video Capture Boards

To enable processing of the images on a computer, the analogue signal from the cameras must be converted back to a digital one. This is the purpose of a VCB. A good quality VCB is capable of capturing from many inputs simultaneously and can convert the analogue signal to a digital signal in real time. Matrox produce a highly regarded series of VCB's, one of which (the Matrox Meteor II Multi-Channel) is available within the Computing department and will be used for the implementation. This board is capable of receiving input from as many as eight separate capture devices.

As mentioned (section 4.1), image ghosting causes problems when trying to find the locations of markers in the image. Using the VCB it is possible to separate a complete interlaced frame into two half height frames known as fields. One field consists of the odd image lines and the other the even lines. As well as eliminating the problems associated with ghosting, this also has the effect of doubling the refresh rate to 50Hz. The sacrifice that has to made for the this increase in framerate is that the vertical resolution of the cameras will be effectively halved.

4.3 Lenses

Most CCD camera's have a standard fitting to allow any compatible lens to be used with the camera. The angle of the lens determines the field of view for the camera. A wide angle lens is used for panoramic views while a narrow angle lens is used for detailed imaging. For the purposes of this project, mid angle

Figure 4.2: An image with decentered, radial distortion and corrected image [2].

lenses of around 60 will be used. This ensures that all of the specified three metre bounding box for the shooter can be captured from close range.

A lens alters the path of light that passes through it and this causes distortions in the image. As the lenses are circular, distortion is minimal at the centre. The distortions have a pronounced effect at the edges of the image, causing straight lines to appear curved (figure 4.2).

Distortions in the image caused by the lens can be corrected for in software [15]. Firstly, the distortion of the lens must be calculated by viewing a regular mesh of points and collecting an image through the lens [23]. The location of the points in the distorted distorted image can then be adjusted to produce the regular mesh that was expected. The “correction vectors” generated by this process can then be used to correct any future images.

There are other methods for correcting image distortion that rely on an accurate model of the camera being constructed though some calibration process. From this model it is possible to determine the factors leading to the distortion, such as radial lens, barrel or thin prism distortion, which can then be removed.

4.4 Infrared Light and Filters

Infrared light has a longer wavelength than visible light and is in the region of 750 - 1,000,000 nm [1]. Short wavelength (750 - 1,000nm) infrared is known as near infrared and is the region that black and white CCD cameras can detect. It is invisible to the human eye and for this reason care is needed when dealing with powerful infrared sources, as the natural human blink response will not be triggered.

Infrared radiation is emitted from any heat producing object, but more intense sources come in the from of lasers and infrared emitting diodes (IRED’s). IRED’s are inexpensive, readily available and provide an intense infrared point source. Infrared lasers produce a narrow collimated beam of infrared light which can travel great distances. They are useful for measuring direction, but are ex-

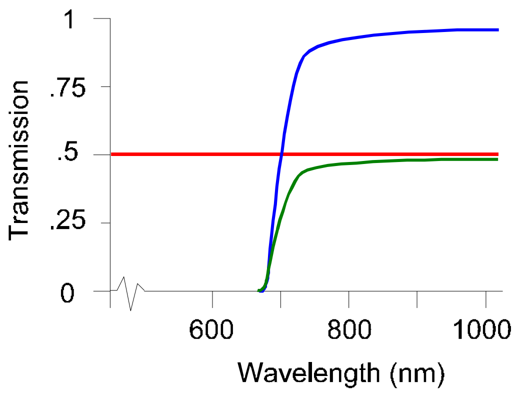


Figure 4.3: Light transmission graphs for Melles-Griot IR filter (blue), Wervatten cutting 2.0 filter (red) and the combination of the two (green).

pensive and run at high temperatures. They also pose a potential health hazard as they are class IIIb lasers [19].

Using glass filters with various metal coatings it is possible to filter out all light except that from the infra-red spectrum. As the IRED's provide such a bright point source, cutting filters can be used to reduce the intensity of light entering the camera - this helps to remove noise effects from sunlight and artificial lighting.

For this implementation, two filters have been used. One of them is an infrared filter from Melles Griot which only transmits light above 700nm, the other is a Wervatten cutting filter which reduces the intensity of all light passing through it. The filters are stacked to produce the desired effect (see figure 4.3).

4.5 Positioning of Cameras

Accurately locating a target depends on the positioning of the cameras. In order to get more accurate results the cameras should ideally be orthogonal to each other, although it is still possible to determine position when the cameras are parallel (as in human eyesight). To make the system is portable the cameras will be attached to a mount. This will ensure that the position of the cameras relative to each other is constant and that they are always well positioned.

In order to be able to determine the position of an object it must be visible from both of the cameras. The region where this occurs is known as the volume of intersection. By changing the position and orientation of the cameras, different volume of intersection can be created (figure 4.4). Angling the cameras towards each other produces a better close range volume of intersection, which is useful for high accuracy positioning of objects close to the sensing equipment moving within a confined space. By arranging the cameras parallel to one another, the

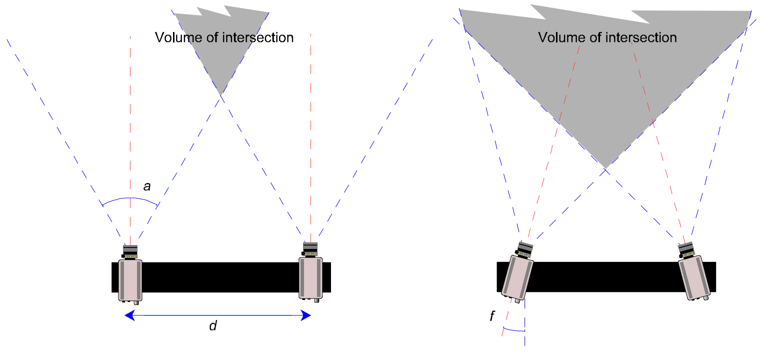


Figure 4.4: Parallel and converging camera layouts. The volume of intersection depends on the viewing angle of the lens *a*, the distance between the cameras *d* and the cameras viewing angle *f*.

volume of interest is much larger, but begins further from the sensing equipment, leading to a loss in accuracy.

In a parallel configuration, to obtain the most accurate results, the spacing of the cameras should be approximately the same as the distance to the target. If the cameras are too far apart, the volume of intersection will be too restrictive and the target may even lie outside, so no inferences on its three dimensional position can be made. If the cameras are too close together then any inaccuracies in the positioning of the cameras will be amplified in the results.

For this implementation, the positioning system will need to be able to locate objects that lie between one and four metres from the rig. The implementation described in this document uses a parallel camera arrangement where the distance *d* between the cameras is fixed at approximately one metre. This ensures that the volume of intersection will begin close to the camera rig. It is expected that the accuracy of the system will decrease over distance.

4.6 Mount for cameras

By fixing the camera to a rigid, the problem of recalibrating the cameras in respect to each other each time the rig is moved is eliminated. For the purposes of this project, a rigid spar mount was designed (figure 4.5). The mount allows a number of different camera configurations depending on the nature of the tracking problem. For close range detailed work the cameras can be angled towards each other. For locating objects at longer rages the cameras can be arranged in a parallel fashion.

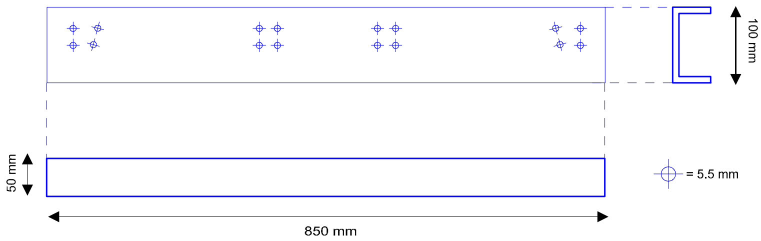


Figure 4.5: The design for the camera mounting spar.

4.7 Summary

By tagging the objects to be tracked with infrared markers and using filters to remove light from the visible spectrum, the task of feature extraction will be made much easier. The cameras will need to be calibrated before they can be used to determine the position of the tagged objects. Without being attached to a mount, the cameras will need to be recalibrated every time they are moved. By using mount, the position of the cameras relative to each other will not change and recalibration is avoided.

Chapter 5

Calibration

> A little inaccuracy sometimes saves tons of explanation.
>
> H.H.Munro (1870 - 1916)

In order to find the location of a target object in three dimensions, it is necessary to know certain properties about each camera, in particular:

- The location of the camera within the world coordinate system,
- The direction in which the camera is facing relative to the world coordinate system,
- The amount of distortion in the images produced by the camera (and how to correct it),
- The field of view of the camera.

Needless to say, these properties are difficult to measure by hand. To derive accurate estimates of these parameters a calibration process can be used.

5.1 Types of Calibration

Camera calibration is an essential step in any three dimensional computer vision task. Without calibration it is impossible to extract metric information from two dimensional images. There are many different techniques for performing camera calibration, but all rely on the fact that the locations of certain points within the received images are known relative to each other. Usually these points are located on a specific calibration object. Depending on the dimension of this object, calibration techniques can be divided up as follows:

Volumetric Calibration is performed by observing a three dimensional object with known feature points. These points are located in the image and paired with their real world co-ordinates. This is the most efficient and accurate method of calibration, but also the most costly in terms of apparatus and time. Making a calibration object of suitable size and precision is often problematic.

Co-Planar Calibration involves observing known points on a planar image, often a regular grid. The plane is observed at several orientations. This method is far easier to implement than volumetric calibration as the planar pattern can be printed off on a standard laser printer.

Linear Calibration uses a number of known points along a line. This technique is a relatively new method and appears to be especially useful for simultaneous calibration of multiple cameras [27].

Self Calibration can be considered as a zero dimensional calibration, in that it does not rely on any particular calibration object. Instead the camera is moved around a static scene. Point correspondences between images are extracted and these are used to determine the properties of the camera. This method involves estimating a number of the fundamental camera parameters and involves much more complex mathematics.

5.2 Camera Models

In order to perform camera calibration, it is first necessary to derive a mathematical model for the camera. There are three major classifications of camera model [14]:

- The Pinhole Model
- The Direct Linear Transform Model
- The Photogrammetry Model

These models are all based on physical camera parameters such as focal length, orientation and lens distortion. The purpose of a camera model is to determine where in the image plane a point in the real world will be projected to.

5.2.1 The Pinhole Model

In the pinhole model, a camera is modelled by its optical centre C and its image plane (figure 5.1). The optical centre is at a distance f from the image plane and this distance is referred to as the focal length of the camera. A point P in world coordinates is projected to image point p in camera coordinates, where p represents the intersection of the line $\overline{CP}$ with the image plane. Using homogenous coordinates (point p becomes $\widetilde{p}$, the transformation from P to p is modelled using a linear transformation $\widetilde{T}$:

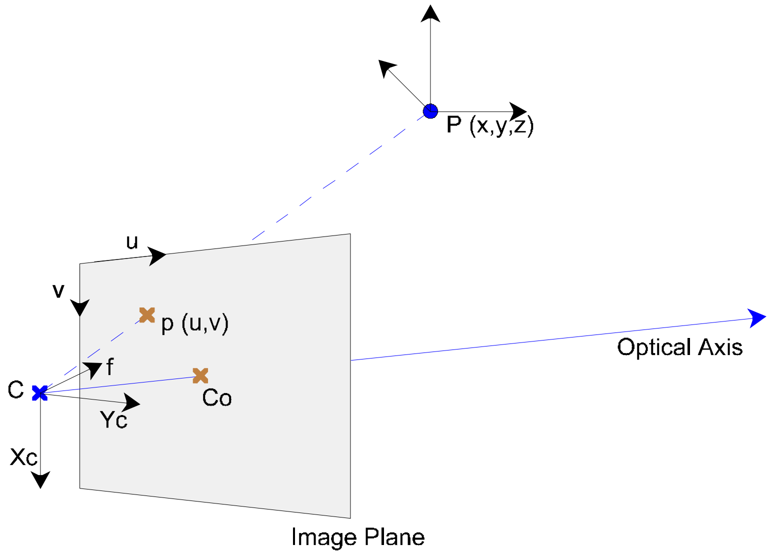


Figure 5.1: The idealised pinhole model of a camera.

$$\widetilde{p} = \widetilde{T}\widetilde{P} \tag{5.1}$$

where

$$\widetilde{p} = \begin{pmatrix} u \\ v \\ s \end{pmatrix} \qquad \widetilde{P} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \qquad p = \begin{pmatrix} u/s \\ v/s \end{pmatrix} \; (if \;\; s \neq 0)$$

The linear transform matrix $\widetilde{T}$ is referred to as the perspective projection matrix. $\widetilde{T}$ can be decomposed into a number of smaller matrices describing the intrinsic and extrinsic properties of the camera. In particular, its position and rotation relative to the world coordinate system, its focal length and the coordinates of the optical centre C_o of the camera.

The pinhole model of a camera is idealised and is impossible to achieve in application as it assumes that the "pinhole" can be reduced to a single point. Most cameras rely on lenses to focus light onto the image plane and these introduce further complications such as radial distortion.

5.2.2 Direct Linear Transform Model

The DLT model is an extension of the pinhole camera model and takes into account non-orthogonal image axes, scaling of image axes and an offset of the optical centre. This model can be extended to take into account radial lens distortion [16]. The project implementation utilises a calibration algorithm developed by Roger Tsai [24] in 1986 that uses this model.

5.3 Tsai's Calibration Model

Tsai's calibration model, is based around the pinhole model of a camera, with 1st order radial lens distortion (see section 5.2.1. Tsai's model is capable of performing volumetric and coplanar calibration. The model has eleven variable parameters. Five of these are the internal or intrinsic parameters:

f effective focal length of the pin-hole camera.

κ_1 1st order radial lens distortion coefficient.

C_x, C_y coordinates of centre of radial lens distortion and the piercing point of the camera coordinate frame's Z axis with the camera's sensor plane.

s_x scale factor to account for any uncertainty in the framegrabber's resampling of the horizontal scanline.

The other six are referred to as the external or extrinsic parameters:

R_x, R_y, R_z rotation angles for the transform between the world and camera coordinate frames.

T_x, T_y, T_z translational components for the transform between the world and camera coordinate frames.

In addition to the 11 variable camera parameters Tsai's model also has six fixed intrinsic camera constants:

N_{cx} number of sensor elements in camera's x direction (in sels).

N_{fx} number of pixels in frame grabber's x direction (in pixels).

Δ_x X dimension of camera's sensor element (in mm/sel).

Δ_y Y dimension of camera's sensor element (in mm/sel).

Δ_{px} effective X dimension of pixel in frame grabber (in mm/pixel).

Δ_{py} effective Y dimension of pixel in frame grabber (in mm/pixel).

5.3.1 Determining the Intrinsic Camera Constants

The values of N_{cx}, Δ_x and Δ_y can be determined by querying the manufacturers of the CCD cameras. The value of N_{fx} is easily determined, but gaining accurate figures for Δ_{px} and Δ_{py} is problematic. The FAQ for Tsai's algorithm [25] suggests taking a straight on image of a rectangular target of length x and height y and then measuring the length and height in pixels of the image of the target.

$$\frac{\text{Width of target in pixels / Actual width of target in mm}}{\text{Height of target in pixels / Height of target in mm}} \tag{5.2}$$

From this ratio, an estimate of Δ_{px} and Δ_{py} can be made which should be close enough for the calculation to converge. If there is any error in the measurement of Δ_{px} and Δ_{py} the calibration algorithm will alter the scaling factor s_x to compensate.

5.3.2 Performing the Calibration

The software implementation uses an optimised version of Tsai's original source code. The calibration library also includes a number of functions that relate image coordinates to world coordinates using the calibrated model. These functions were encapsulated in C++ in an object oriented fashion. The most useful of these functions are listed below:

```
Point3d image_to_world_point(
    Point2d p,
    double z);
// This function takes the location
// of a feature point in pixel
// coordinates and determines the
// real world coordinates of the
// point assuming that it lies on the
// Z plane specified by the z argument.

Point2d undistort_coord(Point2d p);
Point2d distorted_coord(Point2d p);
// These two functions allow for
// correction of radial lens distortion
// in the images. The point p is either
// undistorted, or distorted according
// to the co-efficient of radial
// distortion kappa1.
```

In the software part of the implementation, the first function will be used to construct "lines" in the simulation environment. These lines will pass though the cameras origin (derived by the calibration process) and a second point which represents a possible location for a feature point in the image in the real world. This possible location is determined by giving an arbitrary value for the z argument in the image_to_world_point() function.

5.4 Calibration Data

Calibration data for Tsia's model consists of the three dimensional coordinates of a number of feature points and their corresponding pixel coordinates in the received image. For an accurate calibration, the location of the features in the image needs to be calculated to sub-pixel accuracy (section 6.2.2). An important point to remember is that the algorithm expects the real world coordinates to be entered using a right hand co-ordinate system, not a left-handed system as used in computer graphics.

5.4.1 Distribution of Data Points

To accurately estimate the radial lens distortion and image centre parameters the calibration data should be distributed broadly across the field of view. The distribution of data points should, if possible, span the range of depths the model is to be used over.

5.4.2 Perspective Projection Effects

To be able to separate the effects of f and T_z on the image there needs to be perspective distortion (foreshortening) effects in the calibration data. For useable perspective distortion the distance between the calibration points nearest and farthest from the camera should be on the same scale as the distance between the calibration points and the camera. This applies both to coplanar and non-coplanar calibration.

For co-planar calibration the worst case situation is to have the 3D points lie in a plane parallel to the camera's image plane (all points an equal distance away). Simple geometry tells us we can't separate the effects of f and T_z. A relative angle of 30 degrees or more is recommended to give some effective depth to the data points.

For non-coplanar calibration the worse case situation is to have the 3D points lie in a volume of space that is relatively small compared to the volume's distance to the camera. From a distance the image formation process is closer to orthographic (not perspective) projection and the calibration problem becomes poorly conditioned.

5.5 Summary

The calibration process is fundamental to the successful implementation of the positioning system. Without accurate calibration, a suitable model for the cameras can not be devised and accurate inferences about the real world location of objects detected in the camera images can not be made.

Once a full calibration has been carried out on the stereo rig, it should not need to be performed again as the position of the cameras retaliative to each other is fixed by the mount.

Chapter 6

Software

> When I'm working on a problem, I never think about beauty. I think only how to solve the problem. But when I have finished, if the solution is not beautiful, I know it is wrong
>
> Buckminster Fuller (1895 - 1983)

When the cameras have been calibrated a relationship will be established between the real world and the images produced by the cameras. The software component of the tracking system needs to extract visual features from these images, determine which ones are relevant and use this information together with the calibrated model of the cameras to determine where in the environment the target objects are.

The Matrox video capture board that will be used in this project comes with a development library (MIL lite version 6.0) written in C. This provides low level functionality for grabbing a pixel array from a particular input device connected to the VCB. The library allows simple image overlays (for text and block graphics) and also provides some windowing functionality to display acquired images.

To facilitate the software development, a number of wrapper classes were written that encapsulate the functionality of the library in an object oriented manner. The source code for these wrapper classes is included in the project source directory (see Appendix B for details).

6.1 Image Correction

The images captured by the cameras will inevitably contain some distortion (section 4.3). If this distortion is not taken into account, then the results gen-

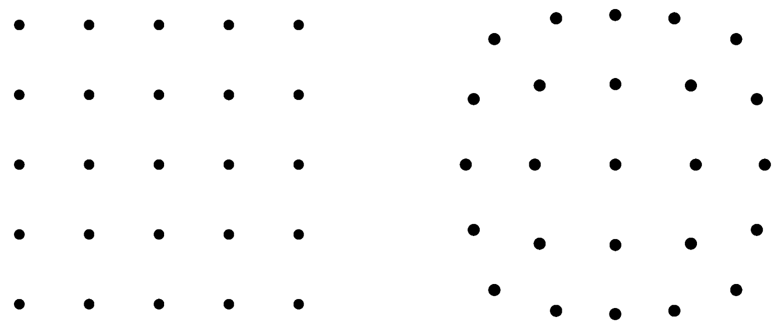


Figure 6.1: Regular grid in a distorted image and the expected image.

erated by the software will be inaccurate. These inaccuracies will became more pronounced as the target markers approach the edges of the image where distortion is highest. The most visible form of distortion in an image taken by a standard camera is radial lens distortion, but there are other forms of distortion that effect the image.

A number of methods exist to correct the distortion in an image. One such method involves taking an image of a regular mesh of points, locating each point in the image and determining where it would appear if the image was undistorted (figure 6.1). An array of correction vectors for various points in the image can then be generated. A correction vector for any point in the image can then be determined by interpolating the nearest vectors.

Another method of removing distortion is to devise a mathematical model that describes the distortion effects in the image. The Tsai camera calibration model (section 5.3) will determine the coefficient of radial lens distortion κ_1 which can be used to correct for radial lens distortion. The implementation uses Tsai's original source code to achieve this.

6.2 Feature Point Extraction

Electro-optical tracking systems rely on the fact that inferences can be made about the real world location of an object from its location in the images received by the cameras. Determining where in the image the object is needs an analysis of the image, which typically involves looking for feature points of the object. For a cube for instance, the corners could be regarded as the feature points.

Feature points within an image can vary from a simple collection of white pixels, to intricate formations such as a human eye. Extracting features from an image is something that human beings are naturally talented at, however performing this process with a computer is a complex issue and the subject of much research. Feature extraction is an intensive process as it often involves analysing a large amount of image data. As this is a real-time system, an efficient feature extraction process is important.

To make the extraction of feature points faster the implementation assumes that all features can be described as "blobs" in an image. A blob is defined as a collection of neighbouring pixels that all have a certain value. A slightly more flexible definition allows the pixels in a blob to lie between certain threshold values (figure 6.2.

6.2.1 Locating Blobs in an Image

The simplest method for locating blobs in a target image is to use an exhaustive search. This involves checking each pixel and determining ifs intensity is between the threshold values. If so, neighbouring pixels are examined to see if they also match until all pixels belonging to the blob have been discovered.

The search for pixels contained in the blob is started at a seed location. An elegant algorithm to locate all the pixels contained in a blob from the seed point is similar to the standard recursive flood fill algorithm:

```
void find_all_pixels_in_blob(int x, int y) {
    if(aleady_visited(x,y)) return;

    if( lower <= pixels[x][y] <= upper ) {
        set_visited(x,y);
        store(x,y);
        find_all_pixels_in_blob(x-1, y);
        find_all_pixels_in_blob(x+1, y);
        find_all_pixels_in_blob(x, y-1);
        find_all_pixels_in_blob(x, y+1);
    }
}
```

6.2.2 Finding the Centre of Gravity

The size and shape of a blob in an image varies greatly. For the purpose of tracking and calibration, a point needs to be defined that specifies the location of the blob. Reducing the blob to a single point location needs careful attention.

The method used in this implementation treats each pixel as if it were a solid portion of an object with mass proportional to its intensity. By calculating the total weight of the blob and finding how the weight is distributed, the imaginary "centre of gravity" (figure 6.2) of the can be found. This is done by evaluating the x and y components separately:

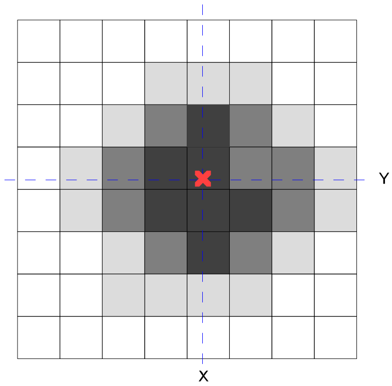


Figure 6.2: A point source as a blob in a pixel array.

$$centre_x = \frac{\sum_{a=X_{min}}^{X_{max}} I_a X_a}{\sum_{a=xMin}^{xMax} I_a} \tag{6.1}$$

$$centre_y = \frac{\sum_{a=Y_{min}}^{Y_{max}} I_a Y_a}{\sum_{a=yMin}^{yMax} I_a} \tag{6.2}$$

The formulas make use of an ordering of the pixels in an blob that is defined as follows. X_{min} refers to the top leftmost pixel in the blob, followed by any pixels in the same column, then the pixels in the subsequent columns ordered from top to bottom until we reach the bottom rightmost pixel, X_{max}. Y_{min} and Y_{max} are defined in a similarly.

This method of blob location provides sub pixel accuracy for determining the centre of feature points within the image and is reliable as long as the effects of foreshortening are not too pronounced (section 6.2.6).

6.2.3 Speeding up the Search

An exhaustive search over a large image involves a great deal of computation. A number of improvements to the search process have been devised to speed up the process. These include:

- Coarse mask classification of multiple pixels,
- Adapting the recursive algorithm to make it non-recursive,

- Subsampling the pixel array.

6.2.4 Coarse mask classification

Each pixel in the image is represented by a single byte. Most modern computers make use of 32 bit processors. When these processors are used to compare single bytes, the byte is read into a 32 bit register and the remaining space is padded out with zeroes. Using a collection of bitmasks it is possible to examine four pixels at a time by treating them as a single 32 bit integer. The technique used in the implementation is to define a single 32 bit mask as follows:

$$\text{mask} = 10000000\ 10000000\ 10000000\ 10000000$$

This mask is combined with the bit values for four pixels using the bitwise **and** operator. If the result of the operation is zero then all of the pixels have values tending towards the black end of the spectrum. If it is non-zero then at least one of the pixels is higher than 128 in value and is therefor near the white end of the spectrum. This technique is especially useful when looking for white features in a predominately dark image, as happens when using infrared point sources and filters.

Non-recursive search

The recursive blob search algorithm defined previously (section 6.2.1) is elegant in design, but suffers from the drawback of many recursive algorithms in that it is processor and memory intensive. A new stack frame must be generated and the function called for every pixel that forms part of the blob. For large blobs (such as an image where all the pixels are the same colour) the recursive approach may even fail as the computer has insufficient resources to complete the calculation. To avoid the overhead associated with recursive calls, the algorithm was adapted to a non recursive version.

Subsampling

Subsampling is a simple technique that can dramatically reduce the time taken to locate blobs in a target image. Subsampling involves examining less pixels than an exhaustive search . For example, by selecting every third pixel from every third row, the search space is reduced to a ninth of the original size. If the features in the image are 3x3 pixels in size or larger, then they will always be located despite the fact that only a fraction of the pixel array has been processed.

It was found during the course of the project that the size of infrared point sources in received images varied depending on their distance from the camera, angle of incidence and the power of the IRED's themselves. In general

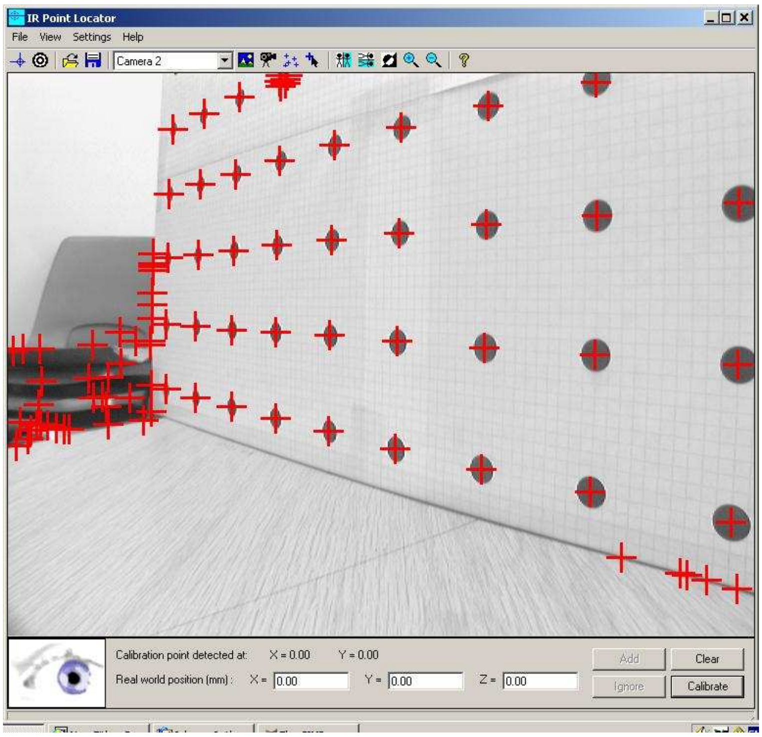


Figure 6.3: Results of the naive feature extraction software.

use, the point sources were always greater than 3x3 pixels in size, enabling the subsampling approach previously described to be used in the implementation.

6.2.5 Removing Outliers

The exhaustive search approach will find all instances of blobs in an image - even if they are only a single pixel in size. In images with a large amount of noise, a huge number of blobs will be detected that do not correspond to feature points. These are known as outliers. Figure 6.3 clearly shows the effect of outliers. The feature points on the calibration pattern have all been accurately located, but there are a large number of outlying points that do not correspond to feature points. To remove them from the results a number of enhancements were added to the search process.

Minimum and maximum size test

Most feature points will occupy more than a single pixel in the image. A large majority of outliers can be removed by ignoring blobs below a particular size. The extension is simple to implement and merely involves incrementing a

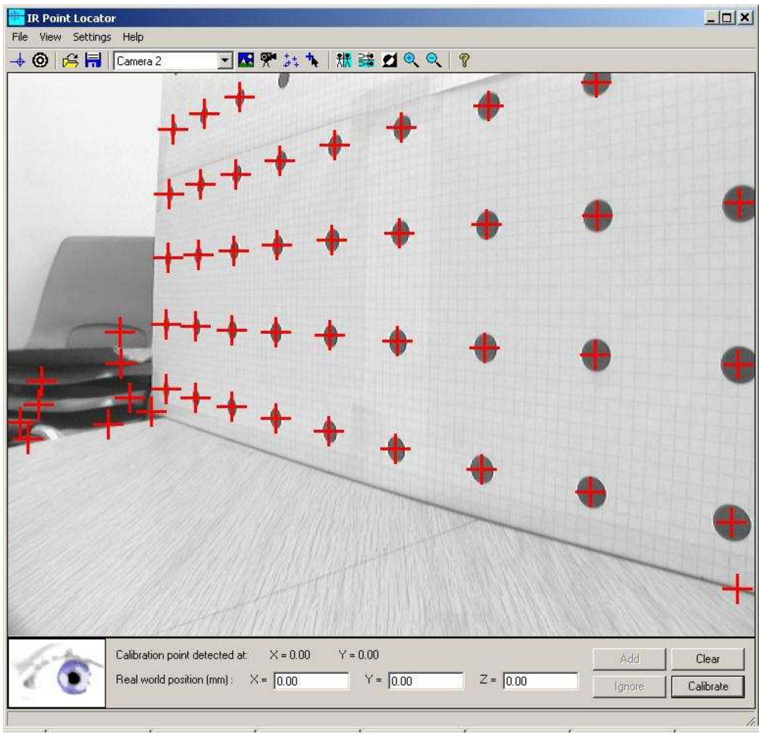


Figure 6.4: Results of the enhanced feature extraction software.

counter for each pixel discovered to be part of the blob. When the entire blob has been uncovered, a check is made to ensure it is larger than the minimum size. If not the blob is regarded as an outlier and ignored. Using the same method a maximum limit for the blob size can also be defined.

The performance of the search algorithm is greatly increased by theses simple measures. Figure 6.4 shows the same image as in figure 6.3 but with the minimum blob size set to 36.

Shapeliness test

Some outliers will still remain after the size based sifting takes place. If the feature points have a regular geometry (for example if they are circular) we can use the relationship between the area of the blob and its perimeter to determine whether its good or not. The implementation allows the user to define a ratio between a blobs area and its perimeter.

The perimeter of the blob is defined by the number of pixels that lie along the outside of the blob. This can be measured with a simple adaptation to the blob location algorithm. This technique is useful for removing long, thin outliers from the result set. Such outliers will have a high perimeter to area ratio, whereas

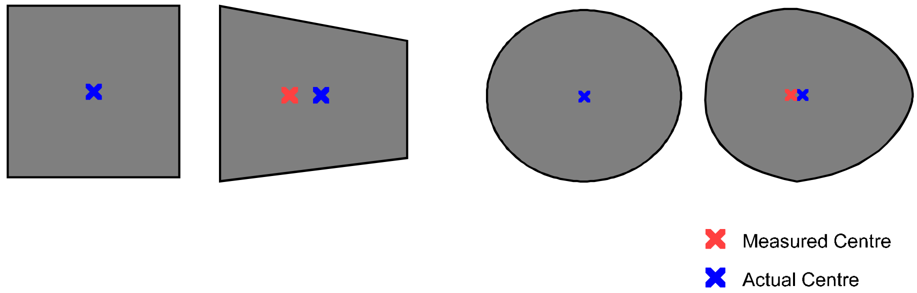


Figure 6.5: The effects of foreshortening on the weighted blob centre.

circular blobs will have a low ratio.

Static analysis

Any outliers remaining can be assumed to be very similar in appearance to the markers. Static analysis involves taking an image of the environment before the markers are placed into it and locating any blobs in the image. Subsequent searches refer back to this information to decide if a blob in the image is new - due to a marker - or was there before and should be ignored.

6.2.6 The Effects of Foreshortening

Foreshortening occurs due to perspective distortion in the captured images. This can have an affect on the the accuracy of the computed centre of the blobs. Figure 6.5 shows how the image of square and circular blobs is altered when they are viewed from an angle (hence introducing perspective distortion). Foreshortening will only occur if the target objects are at an angle to the cameras viewing plane.

The effect of the foreshortening is to "pull" the centre of mass further forward. Foreshortening effects are less noticeable with circular blobs than square ones as most of the circle's mass is along its axes. The implementation does not take the effects of foreshortening into account and this may impact the accuracy of the system. Inaccuracies can be minimised by attempting to avoid foreshortening. This can be done by ensuring the calibration object is parallel to the camera viewing plane and also by making the feature points small.

6.3 Finding the Position of Target Objects

Using image data from two cameras at known positions we can determine the location of a marker in three dimensional space. The two dimensional coor-

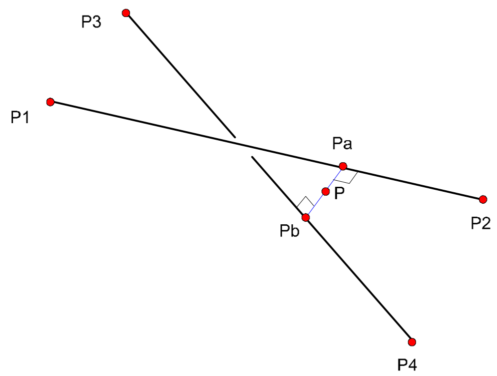


Figure 6.6: Two lines, joined by a line segment that is perpendicular to both.

dinates of the blobs are used to produce vectors from the focal points on the cameras. The three dimensional location of the marker is at the point where the vector from one image intersects the vector from the other image.

6.3.1 Intersection of two lines in 3d space

Once the location of the feature points has been discovered, this information can be used with the calibrated model for each camera to make inferences about the real world location of the feature points. A model for a single camera can determine the line on which the target object lies within the environment.

To find the location of an object in three dimensional space using stereo vision, a line is projected from each camera to the target object. The intersection of these lines gives the position of the object. Small inaccuracies in the calibration of the cameras and the measurement of the feature points mean that these lines will rarely intersect, but pass close to each other at a certain point [5].

If the two lines a and b are not parallel then there is a point P such that P is equidistant to both lines and there is no other point closer to both a and b. This point can be calculated by first finding the shortest line segment between the two lines and then finding the midpoint of this line segment.

To find the shortest line segment that joins two lines, we first define a pair of equations for the two lines. A line is defined as a point p on the line and a vector d along the direction of the line. Any point P on the line can be expressed as the offset of some multiple μ of d from p. The vector d can be easily worked out by using a second point on the line. So for the two lines above, we have:

$$P_a = P_1 + \mu_a(P_2 - P_1) \tag{6.3}$$

$$P_b = P_3 + \mu_b(P_4 - P_3) \tag{6.4}$$

There are two approaches to finding the shortest line segment between lines a and b. The first method is to write down an equation for the line segment and to find the shortest instance of this line. Or more formally, to minimise the following equation:

$$\|P_a - P_b\|^2 \tag{6.5}$$

Substituting the equations for P_a and P_b gives:

$$\|P_1 - P_3 + \mu_a(P_2 - P_1) - \mu_b(P_4 - P_3)\|^2 \tag{6.6}$$

This equation can now be expanded out into the x,y and z components. When the equation is at its minimum, the derivative with respect to μ_a and μ_b will be zero. Solving these two equations gives μ_a and μ_b which can be substituted back into the original equations to give the endpoints of the line segment.

An alternative approach uses the fact that the shortest line segment between the two lines will be orthogonal to both of the lines. Two lines are defined as orthogonal to each other if the dot product of the two lines is zero. As the shortest line segment is orthogonal to both lines, we get a pair of equations:

$$(P_a - Pb).(P_2 - P1) = 0 \tag{6.7}$$

$$(P_a - Pb).(P_4 - P3) = 0 \tag{6.8}$$

By substituting in the equations for P_a and P_b given above, we obtain:

$$(P_1 - P_3 + \mu_a(P_2 - P_1) - \mu_b(P_4 - P_3)).(P_2 - P1) = 0 \tag{6.9}$$

$$(P_1 - P_3 + \mu_a(P_2 - P_1) - \mu_b(P_4 - P_3)).(P_4 - P3) = 0 \tag{6.10}$$

Rearrangement of these formulas allows us to obtain μ_a and μ_b which can be substituted into the original equations to gain the shortest line segment.

Working out the midpoint P of the line segment is a trivial matter once its endpoints have been found. In the implementation, the point P is regarded as the closest point of intersection of the two lines.

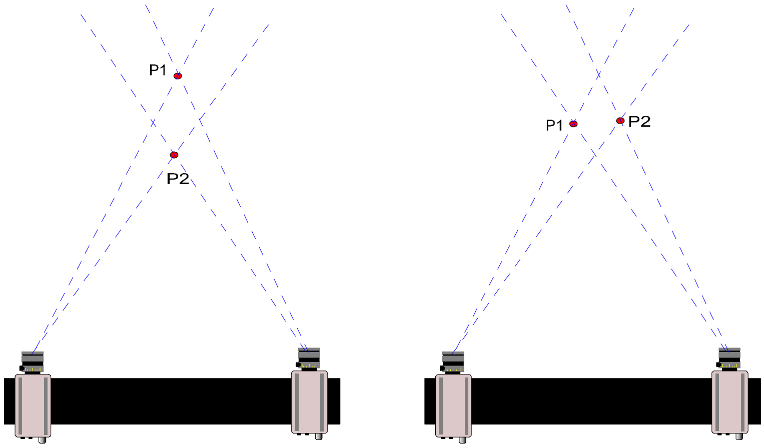


Figure 6.7: The problem of ambiguity when deciding point correlations.

6.4 Resolving Ambiguities

When more than one feature point is located in an image, ambiguities may arise. This occurs when trying to resolve which feature points in one image correspond to the feature points in the second image. With two feature points there are two ways to pair up the points, which result in drastically different positions being calculated (figure 6.7).

6.4.1 Epipolar Geometry

Epipolar geometry [9] [4] is a convenient way of defining the relationship between feature points in images captured by separate cameras. This helps to avoid the problem of ambiguity. Consider a stereo imaging rig composed of two cameras conforming to the pinhole camera model (figure 6.8). The line that joins the optical centres C_1 and C_2 is known as the baseline.

The epipole for a camera is defined as the point of intersection of the baseline with the image plane. Or to put it another way, the epipole is the image in one camera of the optical centre of the other camera (although the cameras themselves may not actually be able to *see* the other camera).

When a point P is located in the environment, a plane defined by P, C_1 and C_2 is formed. This is known as the epipolar plane. The epipolar plane can also be defined by an image point (on one of the viewing planes) and the optical centres.

The line formed by the intersection of the epipolar plane with the image plane of a camera is known as the epipolar line. It is the image in one camera of a ray which passes through the optical centre and image point in the other camera. If a point is located in one image, there is therefore a line (the epipolar line) in

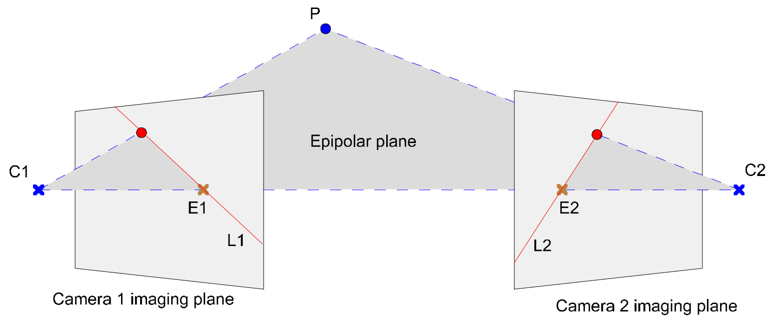


Figure 6.8: Two viewing planes with epipolar lines.

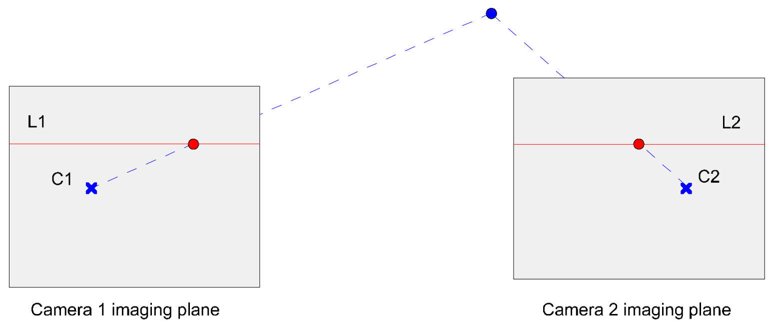


Figure 6.9: Two parallel viewing planes with epipolar lines.

the other image that the point must lie on.

This is very useful feature as it allows correspondences between feature points in images to be determined. Constructing the epipolar line also helps to reduce the search time for the a corresponding point in the second image. As it is known that the point must appear on the epipolar line, only the pixels along that line in the pixel array are examined. If the feature point is not found it can be assumed that it is either out of the second cameras field of view, or is occluded.

6.4.2 Parallel Epipolar Configurations

If the cameras are arranged in a parallel to each other, facing in the same direction the epipolar lines become parallel (figure 6.4.2). This is a very useful feature during the matching process as a point $p = (x,y)$ in one image is guaranteed to lie on the line y in the other image.

The implementation of the project uses the parallel epipolar constraint to determine the correspondence between feature points in the image. The search algorithm also makes use of this constraint when searching for feature points in the second image.

6.5 Motion Tracking and Prediction

Using information from previous measurements an estimate of an objects velocity can be calculated. This estimate can then be used to predict the future location of the object and thus speed up the search process. The accuracy of this method depends upon the motion of the object and the size of the timestep. A large timestep will produce inaccurate predictions for an object that has rapidly changing motion, such as a mosquito in flight. The same timestep may be perfectly acceptable when following slower moving objects such as a slug.

The actual position is related to the predicted position by a normal distribution, with variance dependent on the timestep and dynamics of the object. In the search process. The predicted position of the object is determined using an estimation of velocity v based on the last two measured positions and the time t between the positions being measured.

$$v = \frac{(P_2 - P_1)}{t} \tag{6.11}$$

The estimated position P_3 is then given as:

$$P_3 = P_2 + vt \tag{6.12}$$

The most "probable" location of the object is at this point. In the tracking software, the search begins here. If the blob is not found in this position, it is likely that it will be located near to the estimated position. The search algorithm examines the pixels immediately surrounding the predicted location of the blob and moves further out from this point until the blob is located.

6.6 Summary

The software component of the implementation uses a feature extraction technique that looks for blobs within the stereo images. A number of enhancements were added to the search process to make it more powerful and to reduce the occurrence of outliers. Epipolar geometry is used to determine correspondences between feature points in the two images and also to speed up the search process.

The location of feature points in three dimensional space is calculated by combining information from both of the cameras. The accuracy of this calibration depends on the standard of the hardware calibration.

Chapter 7

Application to Clay Pigeon Shooting

> There is only one admirable form of the imagination: the imagination that is so intense that it creates a new reality, that it makes things happen.
>
> Sean O'Faolain (1900 - 1991)

The hardware and software components described in the previous two chapters combine to form a general purpose positioning and tracking system. To enable the system to provide input to a clay pigeon shooting simulation, a number of extensions to the software were made.

7.1 Finding the Shotgun

Two points are needed to define the location and direction of aim of the shotgun. The shotgun was tagged using two infrared markers, one at either end of the barrel. The tracking system search software was adapted to look for these two points and treat them as two points on a line. The tag at the base of the barrel provides the position of the gun and the tag at the front of the barrel gives the direction of aim.

As only two tags need to be found, the search algorithm was speeded up by using progressive subsampling. This is an extension to subsampling that uses progressively finer distances between samples. This technique is useful when there are a known number of feature points to search for within the image as the search can be stopped when the required number of points has been found.

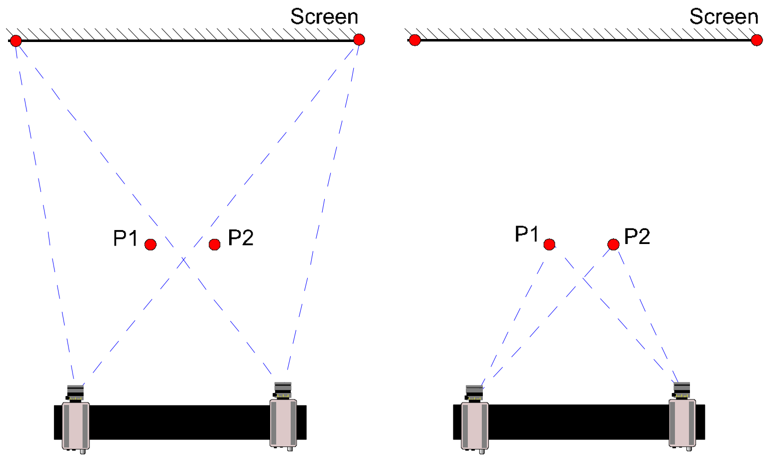


Figure 7.1: Locating the screen corners and reference points.

7.2 Mapping the screen

Once the location and direction of the shotgun have been determined, it is necessary to find out how this relates to the simulation environment. This involves finding the location of the display screen relative to the world coordinate system. This can be done by hand, measuring the distances to the screen corners and entering them into the program. This seams unreasonable however the tracking system itself should be able to provide some assistance here. Two methods of locating the corners of the screen were investigated.

7.2.1 Coordinate transform

This method of locating the screen corners involves placing the tracking system in such a position so that it can see the entire screen. The world coordinates of the corners of the screen are obtained using the positioning system. The positioning system is also used to obtain the coordinates of a number of reference points (figure 7.1). It is important that these reference points will also be visible from the location where the system is intended to be used.

The system is then moved to the location where it shall observe the user and their weapon. The location of the reference points is determined. Based on the information gathered in the first stage, the location of the screen relative to the reference points can be evaluated (figure 7.2).

Once the corner points of the screen have been evaluated the software is able to determine a relationship between the location of the shotgun and the location of the screen (figure 7.3). In practice, this method proved to be too inaccurate to be of any use and an alternative method was found.

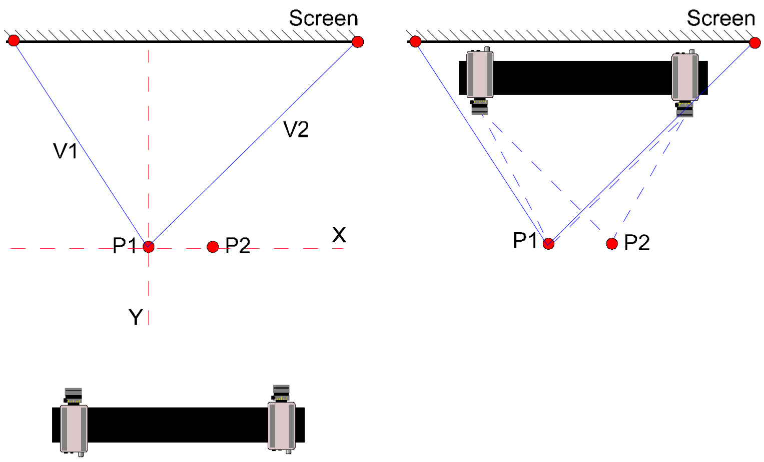


Figure 7.2: Locating reference points in the new position and using this information to locate the screen.

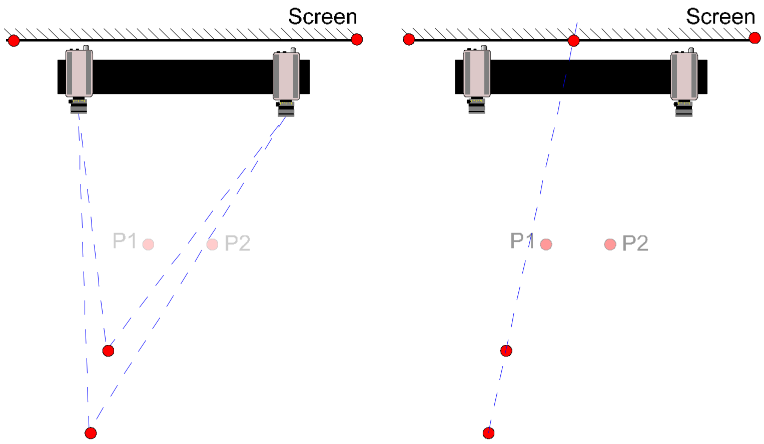


Figure 7.3: Locating the shotgun in relation to the screen.

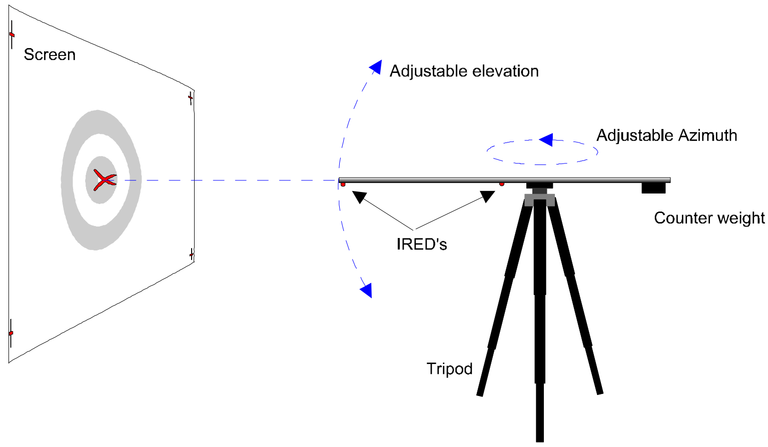


Figure 7.4: The sighting rig for determining vectors to the screen corners.

7.2.2 Sighting of corner points

Calibrating the position of the screen and then moving the system to a new location leads to a build up of inaccuracies. The "sighting" method allows the positioning system to remain in one place. The positioning system is used to determine two or more lines to each corner of the screen from different locations. The intersection (or closest point of intersection will give the location of the corner).

The lines to the corners of the screen were determined using a pipe with 2 infrared tags mounted on it, similar to the markers used to tag the shotgun. This was then mounted firmly on a tripod so that its elevation and azimuth could be accurately set (figure 7.4). The apparatus was placed in various locations and "sighted" at the corners of the screen. The positioning system was used to determine the location and direction of aim of the tube and the required lines were constructed from these measurements.

The same calculation for the intersection of two lines as used in the positioning software (section 6.3.1) can be used to determine the screen corners. Note that for two lines, the estimate of the closest point of intersection is most accurate if the lines are orthogonal. The accuracy of this estimate decreases proportionally with the angle between the lines. In the most extreme case (where the lines are parallel), no estimate of the point of intersection can be given.

If more than two lines are defined for a corner then an average of the resulting closest points between these lines can be found. The averaged position should be weighted towards the most reliable of these points (where the angle between the lines is greatest). Using the properties of dot product it is possible to weight the location depending on the angle between the lines. Recall that $a.b = 1$ if the lines are orthogonal and $a.b = 0$ if they are parallel. The algorithm is as follows:

```
Point3d estimate_screen_corner(
        Line3d *lines,
        int number_of_lines)
{
    double total_weight = 0;
    Point3d p(0,0,0);

    for(int i = 0; i < number_of_lines - 1; i++) {
        for(int j = i + 1; j < number_of_lines; j++) {
            double weight = lines[i].dot(lines[j]);
            p += weight * lines[i].closest_point(lines[j]);
            total_weight += weight;
        }
    }

    return p / total_weight;
}
```

This technique ensures that inaccurate point estimates caused by near parallel lines will not impact on the estimates of the screen corners.

7.2.3 Correction of location of screen corners

In both of the methods described to find the position of the screen corners, there will be a certain amount of error. This means that the four points are unlikely to be planar or form a perfect rectangle. In order to accurately determine where a shot from the shotgun will intersect the screen, we need to adjust the points to ensure that they are planar and do form a perfect rectangle.

Let the four corners of the screen be numbered clockwise from the top left and call the P_0, P_1, $P_2$and P_3. Now define for intermediate points as follows:

$$
\begin{aligned}
P_{01} &= \frac{(P_1 - P_0)}{2} \\
P_{12} &= \frac{(P_2 - P_1)}{2} \\
P_{23} &= \frac{(P_3 - P_2)}{2} \\
P_{03} &= \frac{(P_3 - P_2)}{2}
\end{aligned} \tag{7.1}
$$

Using simple geometry it an be shown that P_{01}, P_{12}, P_{23} and P_{03} will be planar. Suppose that P_{12} and P_{03} define an intermediate x-axis and also that P_{01} and P_{23} form an intermediate y-axis. These two axis are guaranteed to intersect at a point C. The value of this centre point can also be worked out by taking the average of the four estimated screen corner points:

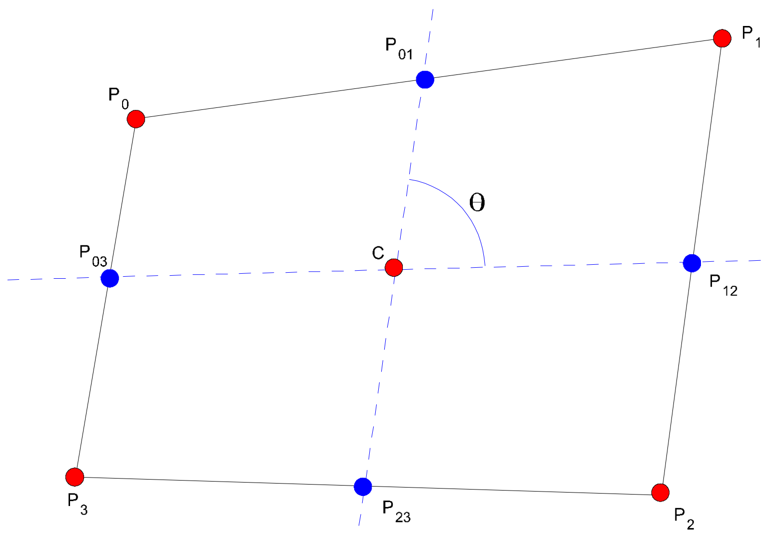


Figure 7.5: Representation of the screen corners and the intermediate points used for the correction process.

$$C = \frac{(P_0 + P_1 + P_2 + P_3)}{4} \tag{7.2}$$

The situation at this stage is depicted in figure 7.2.3 Now the angle θ between the intermediate axes can be found by computing the inverse cosine of their normalised dot product:

$$\theta = \arccos((P_{03} - P_{12}).(P_{23} - P_{01})) \tag{7.3}$$

The axes need to be rotated to make them orthogonal. To do this we define a correction angle ϕ as follows:

$$\phi = \frac{\frac{\pi}{2} - \theta}{2} \tag{7.4}$$

Points P_{12} and P_{03} are then rotated about the normal of the two axes by ϕ radians and points P_{01} and P_{23} are rotated about the normal by $-\phi$ radians. The estimates for the position of the screen corners can now be refined:

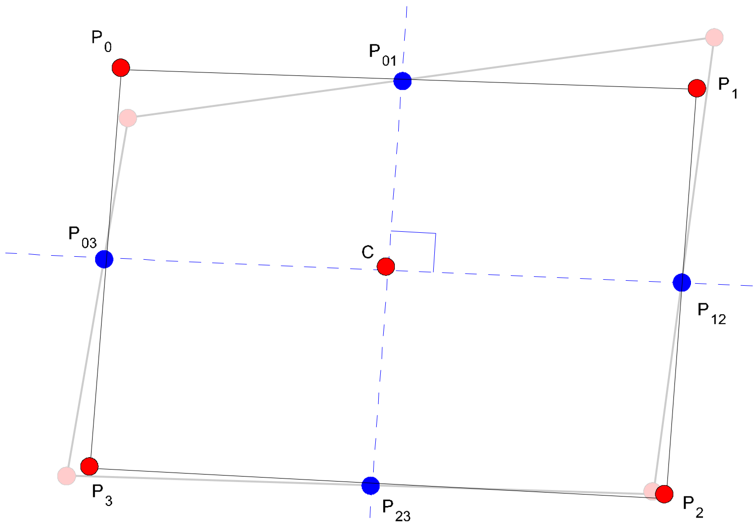


Figure 7.6: The corrected corners after the intermediate points were rotated about the centre.

$$
\begin{aligned}
\Delta_x &= p_{12} - C \\
\Delta_y &= p_{01} - C \\
P_0 &= c - \Delta_x + \Delta_y \\
P_1 &= c + \Delta_x + \Delta_y \\
P_2 &= c + \Delta_x - \Delta_y \\
P_3 &= c - \Delta_x - \Delta_y
\end{aligned} \tag{7.5}
$$

The effect of these transformations can be seen in figure 7.6. These points will be planar and will also form a perfect rectangle, enabling more accurate calculations regarding the location of the screen to be performed.

7.2.4 The shotgun and the screen

With the position of the screen corners known, it is now possible to determine where a shot aimed at the screen will hit. This is done using simple geometry to determine where the line formed by the position and direction of the shotgun intersects the plane defined by the screen corners.

7.3 Summary

The tracking software was adapted to allow estimates to be made for the coordinates of the screen corners. These estimated points are corrected to ensure that they are planar and rectangular. When the user fires the shotgun, the tracking system locates the start and end points of the barrel which are marked with infrared tags.

At present, the location of the shotgun and its direction of aim, as calculated by the system, aim are sent to a simple server that prints out the results. For full simulation the results needs to be transmitted to the simulation software. If the tracking and simulation software are running on the same computer, then shared memory will be the fastest method of communication. If they are running on separate machines the results will have to be sent over a network. Network transmission has inherent overheads and this will increase the latency of the system.

Chapter 8

Evaluation

> Part of the inhumanity of the computer is that, once it is competently programmed and working smoothly, it is completely honest.
>
> Isaac Asimov (1920 - 1992)

The effectiveness of the implementation was evaluated using a number of different tests. These tests and their results are explained in this chapter. A discussion on the limitations of the system that were discovered during the course of the project is also included.

8.1 Spatial Accuracy

The accuracy of the system is one of the most important criteria to assess. The specification suggested a minimum spatial accuracy of ± 10mm for positioning of the target markers. This was evaluated by placing a target marker at a known point in the simulation environment and using the system to determine its position. The result is compared to the actual location of the marker and the error is simply:

$$error = expectedposition - calculatedposition \tag{8.1}$$

This gives the absolute error of the system. A different formula can be used to provide a percentage error in relation to the total distances involved but this kind of analysis is meaningless when assessing spatial accuracy.

After calibration, the stereo rig was placed on a planar surface with an accurate world coordinate system drawn on it. A target object tagged with a single infrared point source was placed on the surface. The position of the point source was calculated using the system and this result was recorded along with the measured position. The target object was moved though a number of translations and results were collected.

Two spatial accuracy tests were carried out. The first test was a close range test (within one metre) where the target object was moved to different locations within a 40 millimetre cube. The second test was conducted at longer range where the target object was translated to different locations within a 2 metre cube. The results for both experiments can be found in Appendix A.

The system performed very well at short range, with an average error of 0.93 millimetres. It is expected that the level of accuracy could be increased further still by performing a more thorough calibration.

In the long range tests, the spatial accuracy was slightly worse, as expected. The average error was 3.4 millimetres. This is well within the 10 millimetre limit laid down by the specification.

8.2 Spatial Accuracy of Shooting Extension

The extended version of the system that monitors the position of a users shotgun was tested using the sighting rig described in section 7.2.2. The rig was set up in six different locations in the defined 3x3 metre simulation space and in each location a shot was fired at five predefined points on the display surface.

The tracking software plots a cross on the screen where it estimates that the shot would intersect. The error between this location and the expected location is calculated as in the previous tests. The results to this test can be found in Appendix A.

The system performed consistently well from all of the locations. And the average error was 9.87 millimetres. Testing was attempted at longer ranges but the IRED's used to tag the barrel were not bright enough to register in the images at these ranges.

8.3 Update Rate

The theoretical maximum update rate of the system is limited to the frame rate of the cameras used (25 Hertz). The update rate can be tested by running the system for a specified time and counting the number of updates received:

$$updaterate = runtime/updatesrecieved \tag{8.2}$$

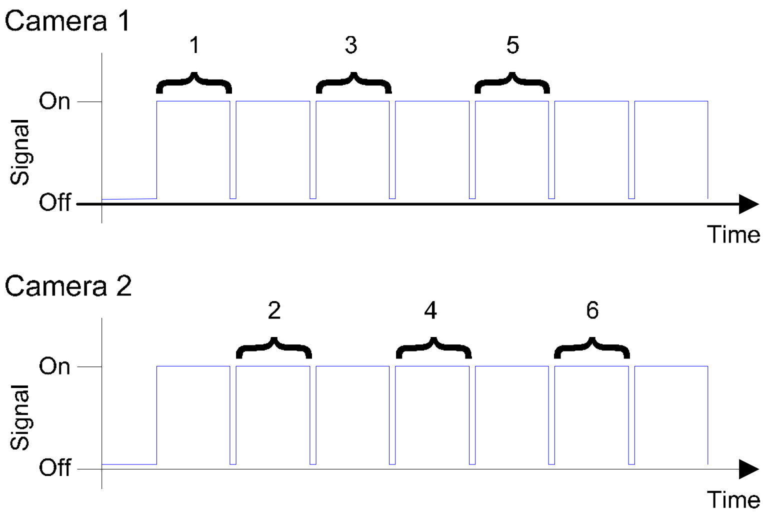


Figure 8.1: The sighting rig for determining vectors to the screen corners.

It was found that the VCB was not able to capture frames simultaneously from the two cameras. Instead, one frame had to be grabbed from one camera and then the next available frame grabbed from the other camera. The effect that this hardware limitation has is dramatic.

Firstly it means that the two images in the stereo pair come from different moments in time - thus reducing the accuracy of the system when it is used to position moving targets. Secondly the update rate is reduced. In the best case, if both cameras are synchronised when a grab is performed then the effective stereo frame rate is halved (figure 8.1). Unfortunately, without linking the cameras to specific synchronisation hardware there is no guarantee that this will occur.

With unsynchronised cameras, the frame rate will be reduced even further(figure 8.2). The VCB grabs a frame from camera one and then switches its input to camera two. Camera two is already halfway through a frame and the VCB has to wait for the next full frame. The result is that the update rate is reduced to a third of the theoretical maximum, or approximately 8 updates per second.

8.4 Safety

One of the criteria given in the specification was that the system should be safe for general use. As the infrared sources used to tag target objects are relatively low intensity, they will not damage human eyes. If (as suggested in section 9.1.1) infrared lasers are used then the issue of safety will have to be reexamined.

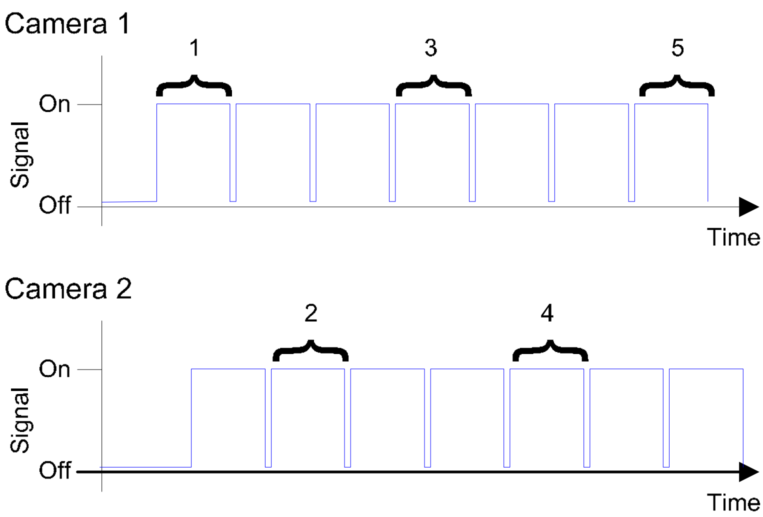


Figure 8.2: The sighting rig for determining vectors to the screen corners.

Chapter 9

Conclusion

Experience is the name everyone gives to their mistakes.

Lady Windermere's Fan, 1892
Oscar Wilde (1854 - 1900)

The aim of this project was to produce a tracking system capable of monitoring the location of a user and their shotgun in a 3x3x3 metre simulation environment. A general purpose tracking system was implemented which was then extended to meet this goal.

A thorough investigation into common positioning technologies was conducted and electro-optical was selected as the most suitable method. The tracking system that was developed made use of accurate calibration and stereo vision to determine the location of target objects within the environment.

The system performed well in accuracy tests and was found to be as accurate as many commercially available systems. Over large distances (in excess of two metres) the accuracy of the system decreased. It is expected that with a more careful calibration, the accuracy of the system can be further improved.

An extension to the system was implemented to enable it to be used in the monitoring of a clay pigeon shooter's shotgun. The accuracy of the system was once again high.

Hardware limitations imposed a limit on the maximum update rate that the system could achieve. It is hoped that the hardware component of the system can be altered to allow a faster update rate.

The tracking system and its extensions demonstrate the accuracy it is possible to achieve using an electro-optical approach to tracking and a robust calibration process. The system has yet to be integrated with the simulation software and it is hoped that this will be achieved once the hardware issues have been addressed.

9.1 Further Work

9.1.1 Using Lasers to Determine Direction

The approach of placing two infrared tags on the barrel to determine its direction of aim works reasonable well. There are however, a two major problems with this method. Firstly, the tags can be occluded from view all too easily. Different users hold the gun in different ways, so no assumptions can be made about whether or not the tags will always be visible. Secondly, without precision adjustable mounts for the tags it is difficult to get them perfectly aligned with the barrel.

A better method of determining the direction of aim may be to use an infrared laser. This laser could be housed inside the barrel of the gun. Not only will this hide the laser from view, but if the fit is snug the laser should become closely aligned to the direction of the barrel.

A glass bead placed at the end of the barrel in the path of the laser will scatter sufficient light to form a very intense point source, without blocking the majority of the laser beam. This beam will continue and hit the screen. The positioning rig can the n be used to locate the point source at the end of the barrel and the point where the laser hits the screen. These two points can then be used to describe the position and direction of ai of the shotgun as before.

9.1.2 Improving the Update Rate

The limitations of the VCB used in the project resulted in a poor update rate. This meant that the effectiveness of the system for tracking fast moving objects was reduced. By using two separate VCB's it will be possible to grab frames at 25 Hz, dramatically improving the update rate.

9.1.3 Calibration

As mentioned frequently throughout this report, the accuracy of the system depends greatly on the quality of the calibration performed. A careful volumetric calibration of the system was carried out, however this was conducted over a small spatial range due the complications involved with accurately positioning calibration objects over large distances. A more extensive and accurate calibration will enable the system to achieve its true potential.

Whilst working on the project, a number of alternative methods of camera calibration have come to light, such as Enhanced Direct Linear Transform and the Haralick model. According to analysis by Jian, Malcolm and Zhongping the Tsai method is the least accurate of these [26]. Using an alternate calibration technique may produce better results.

9.1.4 Completing the Interface

A radio operated trigger was designed that would send a signal to the tracking system when the trigger on the gun was pulled. This was not fully integrated into the system and so trailing wires were used to send the trigger pulse instead. These detract from the interface in that they place restrictions on the movement of the user.

With an improved update rate, accurate calibration and more natural interface, the tracking system will be more than capable of fulfilling the role of monitoring a clay pigeon shooter in a simulation.

Appendix A

Test Results

Test Number	*Expected Position*			*Measured Position*			*Total Error*
	pos_x	pos_y	pos_z	pos_x	pos_y	pos_z	
1	300	20	300	300.23	19.87	300.45	0.52
2	320	20	300	320.48	20.01	300.11	0.49
3	320	20	320	319.19	20.02	319.21	1.13
4	300	20	320	299.13	19.39	319.50	1.17
5	340	20	300	340.41	19.81	301.04	1.13
6	340	20	340	339.47	20.24	340.66	0.88
7	300	20	340	300.28	19.12	340.05	0.92
8	300	40	300	298.99	39.72	300.12	1.05
9	320	40	300	320.43	39.69	300.81	0.97
10	320	40	320	320.87	40.08	320.10	0.88
11	300	40	320	300.15	40.79	320.42	0.91
12	340	40	300	340.23	40.20	299.38	0.69
13	340	40	340	339.39	40.61	340.46	0.86
14	300	40	340	300.52	39.85	340.78	0.58
15	300	60	300	299.97	61.20	300.91	1.51
16	320	60	300	319.89	60.83	300.81	1.65
17	320	60	320	320.09	60.22	320.49	0.54
18	300	60	320	300.91	60.90	320.56	1.40
19	340	60	300	339.86	60.23	300.47	0.53
20	340	60	340	340.00	60.74	340.97	1.22
21	300	60	340	299.88	60.51	339.85	0.55

Table A.1: Moving a target within a small volume at 1 metre range

Test Number	*Expected Position*			*Measured Position*			*Total Error*
	pos_x	pos_y	pos_z	pos_x	pos_y	pos_z	
1	2500	20	2500	2501.73	19.86	2501.71	2.43
2	4500	20	2500	4501.98	18.91	2501.10	2.56
3	2500	20	4500	2500.89	19.82	4498.82	1.49
4	4500	20	4500	4502.64	19.03	4501.65	3.26
5	2500	2020	2500	2502.31	2022.41	2502.23	4.01
6	4500	2020	2500	4501.43	2023.83	2498.84	4.24
7	2500	2020	4500	2503.22	2022.49	4502.29	4.67
8	4500	2020	4500	4501.05	2023.48	4502.76	4.58

Table A.2: Moving a target in a large volume at ¿2 metre range

Test Number	*Expected Position*	
	pos_x	pos_y
1	38.94	89.81
2	37.42	89.38
3	38.93	91.27
4	42.25	88.19
5	39.59	88.69
6	37.40	90.01
7	40.86	90.66
8	39.75	89.77
9	37.29	91.13
10	36.37	90.58
11	40.36	88.08
12	37.38	89.52
13	41.20	88.11
14	38.57	89.79
15	36.88	89.98
16	37.11	91.39
17	41.71	87.36
18	36.46	92.15
19	38.27	88.16
20	36.24	89.36

Table A.3: Static stability of point of intersection calculations

Test Number	*Expected Position*		*Measured Position*	
	pos_x	pos_y	pos_x	pos_y
1	1160	515	1163.37	518.56
2	1210	560	1214.29	565.02
3	1725	750	1728.99	654.28
4	730	750	733.20	754.13
5	940	1970	933.46	1976.66
6	2105	1815	2109.73	1821.97
7	2410	850	2415.73	855.65
8	0	940	-2.78	945.02
9	0	0	-3.94	3.81
10	200	200	204.94	205.12
11	400	400	403.63	405.86
12	600	600	605.84	604.38
13	800	800	802.66	806.43
14	1000	1000	1003.01	1003.43
15	1200	1200	1206.34	1202.89
16	1400	1400	1405.73	1405.08
17	1600	1600	1604.32	1603.18
18	1800	1800	1805.65	1802.35
19	2000	2000	2007.50	2003.72

Table A.4: Arbitrary point choice and aiming results translation method

Appendix B

Software Compilation

B.1 Acquiring and Compiling the Source Code

B.1.1 Downloading the software

The implementation was written entirely in C++ and the source code can be downloaded from
`http://www.doc.ic.ac.uk/~sac99/project/source`. This link should remain active until late 2003, but will be eventually become inactive. When this occurs, if you desperately need the source code (or have any questions) you can e-mail me on s.coulson@hushmail.com.

B.1.2 Development Environment

The software was developed using Microsoft Visual C++ version 6.0. The project file is called Locator.dsw and is in the root of the source code directory. When compiling, ensure that the release or debug build is to set to multi-threaded in the Project Settings menu.

B.1.3 Third Party Libraries

In order for the program to compile, you will need to link to the Mil Lite 6.0 libraries which are in the MIL folder in the source code directory. The GUI was developed using wxWindows and the wxWindows libraries are in the wxWindows folder.

Appendix C

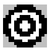
User Manual

The information in this Appendix is designed to allow a user of the positioning system to quickly get to a stage where they can begin determining the location of target objects. A much more comprehensive guide can be found in the *README* file in the source code directory.

C.1 Quick Command Reference

In calibration mode, the function of the buttons on the toolbar is as follows:

 Switches to positioning mode. You can not enter positioning mode until both cameras have been calibrated.

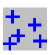
 Displays the target on the screen for calibration of the screen corners.

 Displays a dialog to let you load either an image in .tif format, a camera calibration file in .cal format or a screen calibration file in .scr format.

 Displays a dialog to let you save an image, camera calibration or screen calibration file.

 Grabs a single image from the currently selected camera.

 Begins a continuous grab from the currently selected camera that continues until the button is pressed again. This is particularly useful for focusing the cameras and setting the black and white reference levels.

 Runs the blob search on the current image and marks all located blobs.

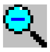
 Allows the user to pick a seed pixel for a blob search using the mouse.

 Flips the images received from the current camera horizontally.

 Flips the images received from the current camera vertically.

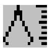
 Displays a control window that allows the user to alter the black and white reference levels of the camera.

 Zooms in on the image.

 Zooms out from the image.

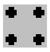
 Displays a (very) brief description of the program.

In positioning mode, the function of the buttons on the toolbar is as follows:

 Return to calibration mode.

 Grab a single pair of stereo images and perform analysis on them depending on the currently selected options.

 Begin grabbing stereo pairs continuously and perform analysis on them depending on the currently selected options. Press again to stop grabbing.

 Select corner calibration mode. The current corner to be calibrated can be changed in the settings menu.

 Change to trigger mode. You can not enter this mode unless both cameras have been calibrated **and** the corners of the screen have been calibrated. When activated, the target is displayed (if it is not already displayed) and the system then waits for the trigger on the gun to be pressed. When this occurs the system determines where the gun was aiming and if the target was hit a marker is drawn in the appropriate place.

 Disable automatic matching of feature points and allow the user to match them by hand.

 Displays a (very) brief description of the program.

C.2 Setup

Before starting to use the tracking system, ensure that the Video Capture Board (VCB) is installed in the computer. Next check that the cameras are linked into the correct ports on the VCB and that they are connected to a 12V DC power supply.

The cameras do not have to be fixed to a mount and can be located anywhere within the simulation environment (with certain provisos), but better results will be achieved if you keep the following in mind:

- Both cameras need to be able to see all of the area where location and positioning will be performed.
- A greater distance between the cameras will provide higher accuracy at long ranges, but will be difficult to calibrate.
- It is important that once calibrated the cameras are not moved relative to each other, otherwise they will have to be recalibrated.
- If using infrared point sources to tag target objects, try to reduce noise effects from sunlight and artificial light as much as possible.

From this point on, the manual will assume that the cameras are firmly attached to a mount (so that their position relative to each other is fixed). This ensures that the cameras can be moved to a new environment without the need for recalibration.

C.3 Calibration

If using the system for the first time, a full calibration will be needed. The system supports two types of calibration, Planar and Volumetric. The type of calibration can be selected from the *Settings* menu in the main window. You will need a calibration object which has a number of clearly visible feature points at known locations. The easiest way to create this object is to print out a pattern of regular dots and stick it to a planar surface such as a hardback book.

For both types of calibration you will need to enter at least twelve calibration points, although accurate calibration needs many more than this. Bear in mind the following:

- For coplanar calibration, all the data points must lie on the plane $Z = 0$;
- If possible, the points should be spread over the entire field of view of the camera.
- For coplanar calibration the plane should be at an optimal angle of 30 to the camera.
- The origin of the real world coordinate system should not be near the optical centre of either camera.

C.3.1 Overview of a typical coplanar calibration

Firstly the intrinsic camera parameters need to be set. This can be done by bringing up the *intrinsic parameters* panel via the settings menu. Change each parameter to suit the camera to be calibrated. For more information see the Tsai calibration FAQ [25].

Check that the camera is working correctly by starting a continuous grab. During the continuous grab you should set the focus and black and white reference levels of the camera (its easier to do this when you get instant feedback). Once you are satisfied with the settings and position the calibration object in the environment. Ensure that it covers a broad field of view of the camera.

Clicking on the *find all* button in the control bar will display all blobs detected in the image. If the number of outliers is high, tweak the search parameters via the settings menu. Once the number of outliers has been reduced to an acceptable level you can begin the calibration (ensure calibration mode is set to coplanar). Clicking on the eye in the bottom left corner of the window will automatically take you though the calibration process. Each blob will be located in turn and you need to enter its x and y co-ordinates.

You can choose to either *add* the blob to the calibration or *ignore* it. Once you have entered all of the calibration points, click on the *calibrate* button. All going well, the calibration process will evaluate the extrinsic parameters of the camera. You can view these from the view menu.

C.3.2 Volumetric Calibration

Volumetric calibration is more accurate, but more time consuming and difficult to set up. One suggested method is to use a planar calibration surface, take an image of it and enter all the calibration points. Then translate the plane by a known distance, take another image and enter the calibration data.

Once both cameras have been calibrated it is recommended that you save the calibration for each camera. You can now enter tracking mode by clicking on the *track* icon.

C.3.3 Tracking and Positioning

Details on the use of the system for tracking and positioning can be found in the README file in the source code directory.

Bibliography

[1] Professional Thermographers Association. What is infrared. 2003. http://www.prothermographer.com/what_is_infrared.htm.

[2] Unknown Author. Correction of image distortion and perspective. 2003. http://philohome.free.fr/barrelpers/barrelpers.htm.

[3] Trojan Aviation. A shooting simulator for sport and competition. 2003. http://www.trojansim.com/.

[4] Sylvian Bougnoux. Learning epipolar geometry. 1999. http://www-sop.inria.fr/robotvis/personnel/sbougnou/Meta3DViewer/Epip%olarGeo.html.

[5] Paul Bourke. The shortest line between two lines in 3d. 1998. http://astronomy.swin.edu.au/~pbourke/geometry/lineline3d/.

[6] Winchestre Rifle Company. The total recoil shooting simulator. 2003. http://www.kramerintl.com/recoil.htm.

[7] The DryFire Corporation. Training aids for clay and game shooters. 2003. http://www.dryfire.fsnet.co.uk.

[8] Asa Denton. Virtual environments for simulated clay pigeon shooting.

[9] Andrea Fusiello. Epipolar geometry. 1998. http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/FUSIELLO/node4.html.

[10] R.A. Giblin and D.J.Compton. Measurement system for the external ballistics and pattern analysis of shot clouds. 1996.

[11] R.A. Giblin and D.J.Compton. Measurement system to assist the development and evaluation of non-toxic shot. 1996.

[12] CURT Inc. Shooting simulators. 2003. http://www.sfab.fsrskytte.se/curt/.

[13] LaserShot Inc. The next generation of shooting simulation. 2003. http://www.lasershot.com.

[14] Sergei Kovalenko. Overview of camera models. 2002. http://vis-www.cs.umass.edu/~kovalenk/thesis/ch2_OverviewOfCameraMode%ls.pdf.

[15] Yung-Chei Lin and Chiou-Shann Fuh. Correcting distortion for digital cameras. 1999.
http://nr.stic.gov.tw/ejournal/ProceedingA/v24n2/115-119.pdf.

[16] T Melen. Geometric modelling and calibration of video cameras for underwater navigation. 1994.

[17] Axel Mulder. Human movement tracking technology: resources. 1998.
http://www.cs.sfu.ca/~amulder/personal/vmi/HMTT.add.html.

[18] NDI. Measurement you can trust. 2003.
http://www.ndigital.com/optotrak.html.

[19] University of Waterloo. Laser classification. 2003.
http://www.adm.uwaterloo.ca/infohs/lasermanual/documents/section8.htm%l.

[20] C.M. Oman. Motion sickness: A synthesis and evaluation of the sensory conflict theory. 1990.

[21] Polhemus. The fast and easy digital tracker. 2003.
http://www.polhemus.com/fastrak.htm.

[22] Ivan Sutherland. The ultimate display. 1965.

[23] Tsuyoshi Yamamura Toru Tamaki and Noboru Ohnishi. Correcting distortion of image by image registration. 2002.
http://www.vision.ie.niigata-u.ac.jp/~tamaki/study/accv2002_dist.pdf.

[24] Roger Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. 1986.

[25] Reginald Willson. Tsai camera calibration faq. 1995.
http://vasc.ri.cmu.edu/IUS/usrp2/rgw/www/faq.txt.

[26] Fang Zhongping Xu Jian, Andrew Malcolm. Camera calibration with micron level accuracy. 2001.
http://www.simtech.a-star.edu.sg/research/TechnicalReports/TR0090.pdf%.

[27] Zhengyou Zhang. Camera calibration with one dimensional objects. 2000.
http://research.microsoft.com/~zhang/Papers/TR01-120.pdf.