# Using Catadioptrics for multidimensional interaction in Computer Graphics

by

James Lane

Submitted in partial fulfillment of the requirements for the degree of Magister Scientiae

in the Faculty of Natural Science

University of Pretoria

Pretoria

Republic of South Africa

November 2001

Dedicated to my Lord Jesus Christ,
Who faithfully stood by me
And saw me through to the end

Thanks to Dad, Mom, John and Lona
Vali for her patience and invaluable help
Andrè Nel
Gerry and Robin for all their help and prayers
Frans, Jaques, Marde, Jeremy, Gernot, Jeurgen, Dr Eckles
and everybody from the
Virtual Environments group at GMD
and the University of Pretoria

# Using Catadioptrics for multidimensional interaction in Computer Graphics

by

James Lane

## Abstract

This thesis introduces the use of catadioptrics for multidimensional interaction in the approach called Reflections.

In computer graphics there is a need for multidimensional interaction that is not restricted by cabling connected to the input device. The use of a camera and computer vision presents a solution to the cabling problem. Unfortunately this solution presents an equally challenging problem: a single camera alone can not accurately calculate depth and is therefore not suitable for multidimensional interaction.

This thesis presents a solution, called reflections to this problem. Reflections makes use of only a single camera and one or more mirrors to accurately calculate 3D, 5D, and 6D information in real time.

Two applications in which this approach is used for natural, non-intrusive and multidimensional interaction are the Virtual Drums Project and Ndebele painting in virtual reality. The interaction in these applications and in particular the Virtual Drums is appropriate and intuitive, e.g. the user plays the drums with a real drumstick. Several computer vision algorithms are described in this thesis, which are used in the implementation of the Virtual Drums Project.

Thesis supervisor: Prof. V. Lalioti

Department of Computer Science

University of Pretoria

Submitted in partial fulfillment of the requirements for the degree of Magister Scientiae

# Die gebruik van Katadioptika vir multidimensionele interaksie in Rekenaar Grafika

deur

James Lane

## Opsomming

Hierdie tesis bied die gebruik aan van katadioptika vir multidimensionele interaksie in die metode wat Refleksie genoem word.

In rekenaar grafika is daar n behoefte aan multidimensionele interaksie wat nie beperk word deur kabels wat gekoppel is aan die invoer meganisme. Die gebruik van 'n kamera en rekenaarvisie is 'n oplossing vir hierdie probleem. Ongelukig presenteer hierdie oplossing 'n soortgelyke uitdagende probleem; 'n enkele kamera kan diepte nie akkuraat bereken nie en dit is daarom nie geskik vir multidimensionele interaksie nie. Daarom is die gebruik van n enkele kamera nie 'n geskikte oplosing vir multidimensionele interaksie nie.

Hierdie tesis stel n oplosing voor vir die probleem wat Refleksies genoem word. Refleksies maak gebruik van 'n enkele kamera en een of meer spieèls om 3D, 5D en 6D inligting akkuraat te bereken.

Twee toepassings waarin hierdie benadering gebruik word vir natuurlike nie indringende en multidimensionele interaksie, word in hierdie tesis aangebied, naamlik die Virtuele Dromme Projek en die Ndebele Skilderkuns in virtuele realiteit (VR). Die interaksie in hierdie toepassings en veral die Virtuele Dromme is gepas en intuitief, byvoorbeeld die gebruiker speel die dromme met 'n regte dromstok. Verskillende rekenaarvisie algoritmes

wat in die implementering van die Virtuele Dromme Projek gebruik word, word in hierdie tesis beskryf.

Studie leier: Prof. V. Lalioti
Department Rekenaarwetenskap
Universiteit van Pretoria

Voorgele ter gedeeltelike vervulling van die vereistes vir die graad Magister Scientiae

# Contents

# List of Figures

.

# List of Tables

# Chapter 1

# Introduction

Interaction is an important aspect of virtual environments in that it affects the sense of presence and immersion these environments aim to achieve. In virtual environments, where the quest is a realistic recreation of reality, it is desirable for interaction to be natural and intuitive. Interaction in virtual environments needs to be multidimensional. The devices commonly used in these environments for interaction and tracking tend to have cables connected to them. These cables restrict the user's freedom of movement and distract the user's attention from the application at hand and thus negatively impact both presence and immersion.

## 1.1 THESIS FOCUS

The focus of this thesis is the creation of a non-intrusive and multidimensional interaction/tracking device that overcomes the above problems. The approach developed in this thesis is called Reflections. It makes use of only a single camera and one or more mirrors for 3D calculation. The approach also calculates 5D or 6D information depending on the number of points the computer vision algorithm tracks on an object.

Because a camera is used to track a user's movement no cabling is necessary. It is an approach well suited for virtual reality and in particular projection-based display systems such as the CyberStage in which the floor projection mirror is used. The approach is also suitable for 3D computer graphics in a desktop environment.

Reflections is implemented in two applications which illustrate its usefulness for supporting natural, non-intrusive and multidimensional interaction in a variety of different virtual reality environments ranging from desktop to large CAVE like projection-based systems.

## 1.2 LAYOUT OF THE WORK

The work presented in this thesis is structured as follows: Chapter two presents a review of relevant literature. This includes an overview of topics in virtual reality, computer vision, interaction and catadioptrics. In the section on virtual reality projection-based systems are highlighted and discussed. Spatial sound is introduced and the different types of tracking devices available in virtual environments are presented. After interaction in virtual environments is discussed, the topic of interaction is dealt with. The section on interaction contains discussions on interaction in general, natural interaction (and the implications thereof) and multidimensional interaction devices. Computer vision is discussed since it is a quintessential component of the Reflections method. The section on computer vision also presents the 2D to 3D problem and solutions to it. Catadioptrics, a field of research that is similar to Reflections, is then presented.

In chapter three an in depth theoretical coverage of Reflections is given. This coverage includes a complete presentation of the different aspects of the approach, the 3D calculation and the 6D calculation. Certain useful computer vision algorithms are discussed in section 3.2 A chroma keying method, SUSAN, image moments and tracking are discussed. These algorithms are all implemented in the Virtual Drums project.

Chapter four presents the design of a tracking device that uses the Reflections approach to implement natural and multidimensional interaction in a virtual drum kit and the virtual environment of Ndebele Painting. In this chapter the practical issues of combining computer vision with Reflections to implement natural interaction in the Virtual Drums project are discussed.

In chapter five test results are given on the Reflections system created for the Virtual Drums project. Results are given for the accuracy of the 3D and 6D calculation and the stability of the system. Results of the system's efficiency and effectiveness for the different algorithms implemented in the computational process are also given. The overall system functions in real time and is therefore able to be used as a tracking device. Results about the tracking system are also presented.

Conclusions are drawn in chapter six and areas for future work are proposed.

# Chapter 2

# Literature Study

This chapter contains an overview of relevant literature and prior work done relating to the creation of the proposed non-intrusive interaction device. This overview will introduce key themes in the work and create the setting for the development of this device. The chapter commences with a look at virtual reality with a focus on projection based display systems. Projection based systems stand to benefit from the proposed interaction device. A discussion on interaction and in particular on natural interaction is then given. Topics in computer vision (CV) and an overview of this subject are presented next as computer vision is a crucial element of a camera tracker. The section ends with an introduction to catadioptrics.

## 2.1 VIRTUAL REALITY

**Figure 2.1 Pictures of Virtual Worlds**
*(Courtesy of GMD)*

At a glance, virtual reality (VR) may be seen as an impressive and technologically advanced form of entertainment as illustrated in figure 2.1. However on closer inspection one will find VR to be far more than just an entertaining experience or fancy collection of technologies. A suitable analogy illustrates this. If VR were a tree with attractive flowers that give way to produce nourishing fruit, then the flowers would represent the attractive technologies and entertaining VR experience while the fruit would represent the noble insights gained from the experience [Brooks]. Although the experience has its benefits, the actual insights gained, discoveries made and things learned from the virtual experience should be considered as the valuable fruit of VR.

If one considers the uses and fields of application for virtual reality, it can be seen that virtual reality has the potential to impact and influence society. Not only this, but it has made and will continue to make significant contributions towards science, education and several other disciplines. Some of the benefits and complex application areas of VR include [Van Dam et al][Macedonia & Rosenblum]:

- the visualization of colossal scientific data sets and models,
- improved understanding and insight into large data sets,
- medical applications such as surgical visualization and planning,

- education, training and art,
- clinical therapy,
- architectural and mechanical design,
- aerial photography and satellites, and
- mechanical, electrical, electronic, civil and mining engineering.

Virtual Reality may be considered as the branch of computer graphics that makes use of realistic imagery and other technologies for developing artificial computer-generated environments that are both responsive and lifelike [Van den Bergh][Van Dam et al]. Virtual reality attempts to make a user's surroundings look, sound, feel and even smell real [Brooks]. But it does not stop here. In the quest for creating a 'virtual reality' it is also necessary that the environment should respond realistically to actions and that the person inside this world should be able to interact with it in an intuitive manner. Finally all of this must all be done in real time [Brooks][Van den Bergh].

The way a person perceives a virtual environment is very important. The sense a person has of actually being in a virtual world and of the virtual world being real is used as a performance measure. This sense a user has of actually being in a virtual world is referred to as presence. Presence is based completely on a user's psychology [Blake et al]. Presence to a large extent depends on a user's involvement and immersion in a virtual environment [Casanueva & Blake]. Immersion is a term that refers to the extent to which a virtual environment (the system) surrounds a user and blocks out external real world sensory data. The immersing or submerging of a user into a virtual world is achieved by replacing the sensory inputs from the real world with those of the virtual world [Blake et al]. Immersion is determined by how well the technologies used to create the virtual world, depict and replace reality.

### 2.1.1 Immersive Technologies

Several different technologies are required to build a system that can effectively immerse a user and give a high degree of presence [Brooks]. The visual display is the key

technology for creating the effect of immersion. This is in most cases a three-dimensional (3D) stereo display. Coupled to this is a graphics rendering system that must display the model of the virtual world at a rate of at least twenty to thirty frames per second. A database system is required to maintain the realistic model of the virtual world. Linked to the display is a head-tracking device. By tracking the position and orientation of the user's head a human centric point of view can be provided. For this reason tracking is considered one of the crucial technologies for immersive virtual reality [Brooks].

Spatial sound and simulated sound fields can also be used to complement the graphics system and assist the virtual environment in achieving an illusion of reality and three-dimensional space. A means for interacting with the system is also needed. This is done by using different interaction techniques, interfaces and devices which allow users to interact with the objects in the virtual world and with the virtual world itself. Other devices such as force feedback devices and olfactory displays are used to further improve the sense of reality.

The visual display is the principle means of achieving effective immersion [Van Dam et al]. The display should be 3D and should substantially occlude the real world. There are several ways to produce realistic 3D imagery. Certain of the methods produce images that physically occupy space while others simply make use of projections onto 2D surfaces and stereo vision. Most of these display systems produce true 3D images [Wartell et al]. True 3D images have a depth cue which causes objects that are nearer to the viewer to appear closer than those further away [Foley et al]. The more conventional methods used to create 3D display systems make use of stereopsis.

### 2.1.2 Stereo Vision

At the front end of the human visual system one finds two eyes located a small distance apart. This separation of the eyes is referred to as binocular disparity. Each eye has a different view of a scene, as illustrated in figure 2.2. The two separate views are known as a stereo pair. The brain fuses the stereo pair to form a 3D image [Foley et al].

Most 3D-display systems take advantage of the way in which the human visual system works, to trick the brain into thinking it sees a 3D image. It does this by presenting each eye with an appropriate synthetic stereo image. A stereoscope works on this principle.

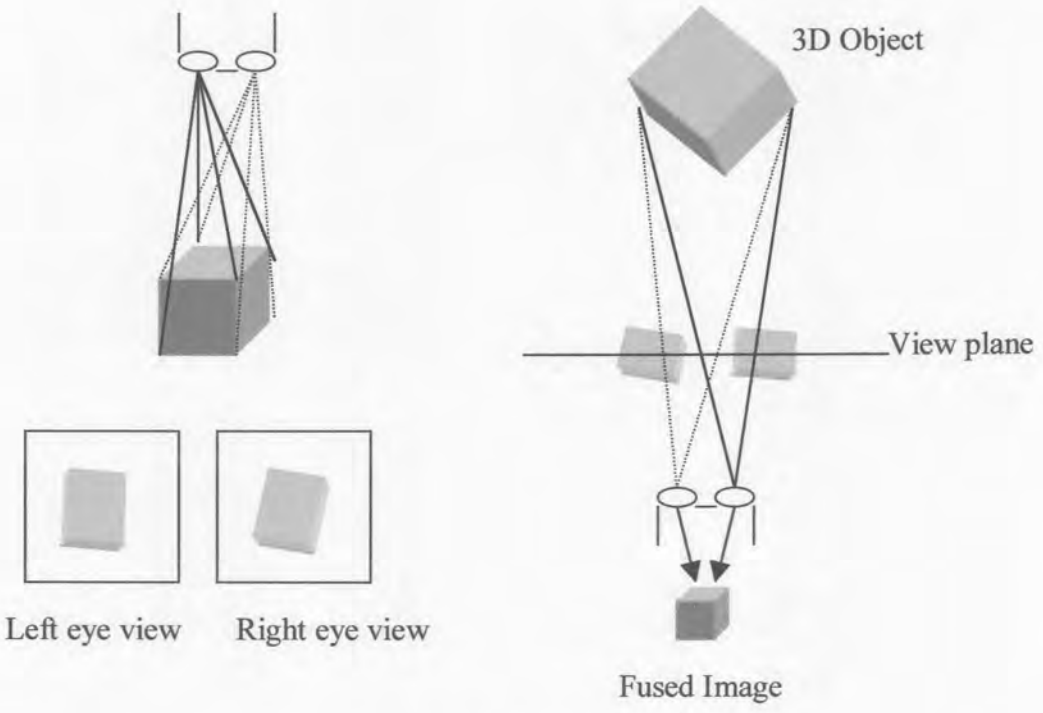



**Figure 2.2 Binocular Disparity.**

Shutter glasses are commonly used to implement stereo vision. Shutter glasses use LCD shutters, one for each eye. These shutters may be opened or closed at a rate of at least 60Hz [Van Den Bergh]. Projections of the left and right eye images are displayed one at a time to the appropriate eye, while the other eye's shutter is closed. In this way the display system provides each eye with its appropriate view of the 3D world. This is done very quickly and hence residual images of the pictures are left for a short period of time in the eye. This results in the eyes seeing the two separate images at the same time. The brain fuses these two synthetic stereo images to form a 3D image. In this way the brain is beguiled into believing that what it sees is 3D. An alternative technology to shutter glasses is the use of polarizing lenses and glasses, which achieve this powerful depth cue in the same way.

The tracking device used in a virtual environment is crucial because it is used to create the appropriate synthetic stereo images of the virtual world from the viewer's perspective. The tracker determines the head's position and direction of gaze. From this information an image generator constructs the appropriate left and right eye images, after rendering the scene once for each of the different eye perspectives [Wartell et al][Van Den Bergh]. These images are then displayed synchronously with the opening and closing of the appropriate shutter.

### 2.1.3 Display Systems

There are several different types of 3D Display systems that make use of stereo vision. The two main different human-centric display systems for virtual environments are head-mounted and projection based displays [Buxton & Fitzmaurice]. Head mounted display (HMD) systems consist of a wide view head mounted stereo display. In this type of system the user's viewing position is tracked. One disadvantage of HMD's is that they are heavy. Projection based systems make use of multiple stereo projection display walls in which stereo shutter glasses are used. The position and orientation of these glasses are tracked to determine what is being looked at and from where, so that the appropriate world-view can be projected. These projection-based systems are sometimes referred to as head-tracked displays (HTD).

In this thesis the focus is on projection based VR because these environments allow the viewer to see the virtual and physical space simultaneously. The user can see his own hands and the tools he is holding as well as other people around him. He can also see the virtual world and objects in it in his physical space.

If there are multiple people in one display system, then a single person's head position is tracked. This means that one person determines the perspective while the others view that perspective from different positions.

One disadvantage encountered in these systems is the shadow effect. The shadow effect occurs in "same location collaboration" caves, when one user stands in front of a projection wall displaying an object to be seen between that user and another, effectively blocking the other user's view of the object [Buxton & Fitzmaurice].

Projection based systems are the most realistic and immersive for the virtual drums application developed in this thesis because the user sees his hands and the tools (drumsticks) he is using for interaction, with the result that interaction looks and feels realistic.

**(a) Stereoscopic Monitors**

Desktop monitors are not considered to be a VR display technology because they do not adequately immerse a user [Brooks]. This is because they are unable to block-out the real world and do not present life sized virtual-world objects. In this paper monitors are discussed because a stereoscopic monitor is a common display system and much of the original system has been developed on a system with such a monitor. In addition examining interaction in 3D with a stereoscopic monitor is an interesting experiment. This is because it presents a test for determining if a sense of spatial awareness can effectively be impressed upon a user by using a stereoscopic monitor in conjunction with natural 3D interaction and spatial sound, rather than relying upon the display system.

Many different types of monitors are available. The most common monitors are liquid crystal displays (LCD) and cathode ray tube (CRT) displays. Large screen display projection CRT's are also available. An example of a large display is General Electric's light valve projection system [Foley]. To make these 2D displays stereoscopic, shutter glasses and stereo vision are needed.

**(b) Workbenches**

The workbench is a projection based display system that consists of one or more large flat projection surfaces. Common workbenches consist of a single surface that may be tilted. By tilting the workbench the height of objects seen on it can be increased. Workbenches cover a large visual angle. Having an angular resolution of about four minutes of arc near the center of the display, these systems can substantially block out the real world. As in all good projection based systems shutter glasses are required and one or more of the viewers can be tracked [Brooks].

The workbench is an example of integrating the computer into the user's virtual world with virtual objects and control tools being displayed as computer generated stereoscopic images. Originally workbenches were designed for human-body models. These models are life-sized when displayed on the workbench [Brooks]. Examples of workbenches include the ImmersaDesk, the Perceptive Workbench, the Responsive Workbench (RWB™) and the Cooperative Responsive Workbench. The latter consists of a smoothly adjacent horizontal and vertical display. The use of the additional horizontal screen enlarges the viewing region and allows virtual objects to be viewed at the user's eye level [Lalioti et al, 1998].

**(c) Caves**

Imagine having an entire room for your display, being completely surrounded by the visual display, you are free to walk around within this room and interact with the graphics you see on the walls. Projection based systems with these characteristics form a fascinating class of VR display technology called a Cave.

CAVEs make use of multiple stereo-projection walls to create a room sized, seamless and omni-directional view of the virtual world. This is ideal for immersion in virtual reality because the user is almost completely blocked off from the real world. Furthermore the user can move around and work within the room [Buxton & Fitzmaurice]. A typical cave

consists of three to six faces of a rectangular cube fitted with rear-projection screens and is driven by a set of coordinated image-generation systems. The projectors used have high resolutions, around 1280x1024 pixels. A ten-foot cave has an angular resolution of four minutes of arc, while a person with twenty-twenty vision has a visual capacity of only one minute of arc. Therefore CAVE systems are considered to be surround projection technology [Brooks]. Caves have certain advantages and disadvantages:

Advantages:

- wide surrounding field of view, and
- shared experience for a small group with at most one head being tracked.

Disadvantages:

- space requirements for rear projection,
- brightness limitations due to large screens, this hinders colour perception,
- corner edge effects,
- reduced contrast colour segmentation due to light scattering,
- shadow effect, and
- cost, since a cave requires multiple image-generation systems.

In a Cave shutter glasses and stereo vision are used to make the display three-dimensional. This allows users to interact directly with virtual objects seen in the Cave. A single users head position and orientation are tracked to give the correct perspective of the user's view. This allows the user to see and look around the virtual world as if it were real. Multiple users can not be head tracked and share the same Cave simultaneously because of the projection constraints. However other people may be inside the Cave and view the user's perspective of the environment. There are certain implications and advantages for interaction arising from the user being physically inside the display system. One such advantage is that a Cave supports body centered human computer interaction.

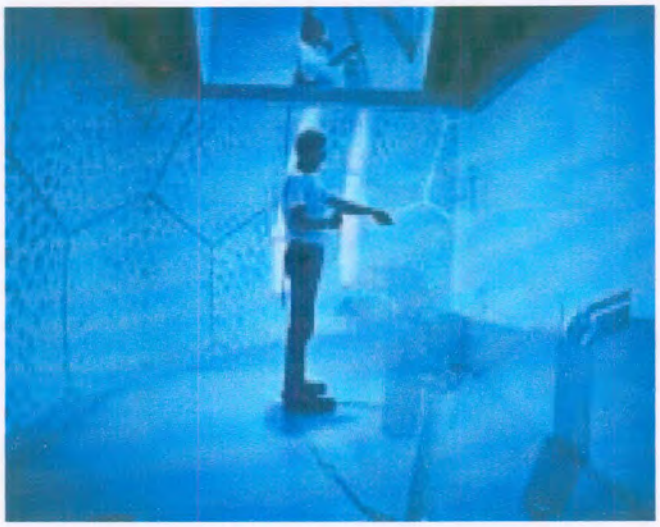


**Figure 2.3 A Picture of CyberStage** *(Courtesy of GMD)*

The first cave was developed at the University of Illinois Chicago [Brooks]. An example of a cave is CyberStage used by GMD (German National Research Center for Information Technology). A picture of the CyberStage is given in figure 2.3. It consists of a room with four stereo projection walls (a floor and 3 walls). It has spatially distributed sound and an acoustic floor. The dimensions of the CyberStage are 3x3x2.4 meters. The projectors used have a resolution of 1024x768 pixels and run at 120Hz. A large mirror situated above the walls is used to reflect projected images onto the floor.

### 2.1.4 Spatial Audio

Spatial sound provides another means for improving immersion. Spatial audio also contributes to a person's 3D spatial awareness and provides information about events that a person can not see or which occur outside his field of view. This is because spatial audio gives the illusion that a sound is coming from a specific position relative to a listener. Consider for instance being in a CAVE and hearing the sound of an explosion behind you. This will draw your attention to an event which occurred beyond what you are currently looking at and gives you an idea of the position of that event.

Spatial audio is simply processed sound that gives a listener a sense of the location of a virtual sound source. The principle behind the technology is that if the sound waves that reach the eardrum from a real audio source at a specific position can be identically recreated and presented to the ear, then that sound from that specific position will be perceived. Digital filters and head related transfer functions are applied to an audio stream to recreate such positional sounds. A requirement for producing true 3D spatial sound is that the position of the user's head needs to be tracked. This is because it is necessary to know the position of the user's head to accurately create realistic spatial sound [Van den Bergh][Spatial Audio].

Earphones are the most common speaker technology used for spatial audio [Spatial Audio]. There are other technologies such as the surround sound system used in CyberStage which are suitable for the playback of spatial sound.

### 2.1.5 Tracking

Trackers are one of the most important technologies for virtual environments because the tracker is used to determine the position of the head and direction of gaze which is needed for both spatial sound and more importantly for the correct rendering of a scene from the user's viewpoint. Tracking is not limited to tracking the position and orientation of the head but is necessary for tracking different objects or devices such as gloves or pointers used for interaction [Lastra]. In certain VR environments interaction is centered around the 3D position and orientation of an object. In these cases an interaction device can be implemented by using a tracking system.

One of the most serious problems that trackers have to overcome is latency. This is because latency is detrimental to presence when tracking head movements and in particular head rotations. This is because a turn of the head can cover a large angle in a fraction of time. Even minimal latency experienced here will affect the way the virtual world is re-rendered. This has a negative impact on presence. This problem is more of a concern for head mounted displays than for projection based systems, since in HMD's the

scene changes and so the generated images of the scene also change when the head is rotated. While in projection based systems head rotation does not change the generated image, it only affects the viewpoint translation, motions of the interaction device and the virtual objects seen [Brooks]. One of the ways to minimize latency is to use a fast interface between the host computer and the tracking device. When tracking data is sent over a network, a fast network should be used.

Another important consideration is that of the tracking region. If the region to be covered by the virtual application is so large that it is beyond the range of the tracker, then other methods for moving through the virtual world, such as flying, are used. However there are trackers that provide coverage for large volumes. An example of such a tracker is Constellation™ [Foxlin et al]. It provides coverage for a building-wide range. In this system tracking is tetherless (wireless). The hardware makes use of ultrasound and a constellation of wireless transponder beacons. The system has no acoustic propagation latency and achieves accuracy somewhere between one and three millimeters. It can be used in both augmented and virtual realities.

### 2.1.6 Sensory Technology for Tracking and Interaction

There are several different technologies commonly used to implement a tracker or input device of some type. These technologies and their advantages and disadvantages are presented below.

**a) Contact Sensing Technologies**

Magnetic trackers are commonly used and have been in use for some time. One of the advantages of this sensing technology is that there is no line of sight restriction and that the sensors are both small and light. Magnetic sensing is however affected by ferrous metals and CRT coils. Magnetic trackers also only cover a small tracking volume [Lastra].

Mechanical tracking is very accurate and has low latency. Unfortunately such trackers have a very restricted range and usually only support one sensor. These systems are commonly found in semi-immersive boom-type display systems [Lastra].

The trackers mentioned above in some way or another require that the sensor be physically connected to the user or object to be tracked. The sensor or tether that is connected to the user can become irksome during use and can even fatigue a user with extended use. Non-contact sensing technologies are reviewed next.

**b) Non-contact Sensing Technologies**

Optical trackers (infrared) provide coverage for large volumes. They are both accurate and fast but have a line of sight restriction. These trackers do not easily support multiple object tracking and are not commercially used in virtual realities [Lastra]. Sunlight or infrared could cause interference. The surface texture and orientation of an object also affect the accuracy and reliability of measurements [Smith et al].

Another interesting optical device is the I/O bulb. It can both emit light and has within it a small video camera for capturing light that passes through the bulb, from outside. This device has been used to achieve interaction in the luminous room, an interactive environment in which the surfaces of certain objects have the ability of both displaying visual information and capturing information. In this system the display is no longer just an output device but an interaction device [Underkoffler et al].

Another sensing technology that is used in an alternative VR environment (fish-tank VR) for tracking is ultrasound. Ultrasonic trackers are simple, accurate and inexpensive. The disadvantages of these systems are that they only cover a small volume and are influenced by occlusions and affected by temperature [Lastra]. Orientation and surface texture affect the accuracy and reliability of measurements taken by ultrasonic sensors. Background noise also constitutes a means of disrupting readings. An example of such a source of interference is mechanical noise [Smith et al].

Electric field sensing presents a clever and powerful way of implementing a non-contact tracking system. By monitoring the electric field generated by a fixed transceiver the position of an object that passes through this field is determined [Underkoffler et al]. Electric field sensing mechanisms are advantageous in that they can track with millimeter accuracy and at millisecond speeds. This is, however, only for ranges up to a meter. Electric field sensing is simple, fast, inexpensive and very accurate. It has been used in desktop applications and in front of the Gesture Wall [Smith et al]. On the downside these techniques are not yet able to adequately distinguish between objects and require that the tracked objects partially conduct an electric field [Underkoffler et al].

Image capture is another non-contact approach to tracking and sensing. It has the following disadvantages [Smith et al]:

- extraneous data is captured,
- an enormous amount of bandwidth is needed,
- constraints are normally made on the lighting and background,
- speed is limited to the cameras frame rate,
- it is also difficult to fish through the sea of information to extract the needed and relevant information at frame rates, and
- there is the line of sight restriction.

In spite of these disadvantages image capture presents a powerful means of sensing, tracking and implementing interaction [Freeman et al]. In the past using image capture for real-time applications was limited by the frame rate and resolution of the camera as well as by slow computer vision algorithms. Today computer vision has advanced significantly and become far more viable for sensing and real-time tracking. Computer speeds have also increased to the extent where they can cope with the large bandwidth and large amount of information that needs to be processed at frame rates. Even slow computer vision algorithms are not so slow when run on powerful and fast machines. Furthermore image capture technology has made large strides forward, e.g. an Artificial Retina Chip [Freeman et al] has been developed. This chip is a powerful image detector for meeting computer vision needs. The chip can capture up to one thousand frames per

second; i.e. it can function at up to 1000Hz. It also has onboard image processing functionality for determining local derivatives and image projections which are used for edge extraction and for calculating image moments. In addition to hardware advances, computer vision algorithms have become faster and even more powerful and clever.

This thesis focuses on implementing sensing, tracking and interaction by means of a computer vision approach.

### 2.1.7 Interaction in VR

Virtual environments, sometimes referred to as 3D interactive environments, require interaction devices that support realistic interaction and which affect multiple degrees of freedom. Furthermore it should be easy for the user to learn how to use the device [Peterson]. 3D interaction devices commonly have one or more buttons for signaling [Van den Bergh]. An example of such a device satisfying these specifications is the Stylus. A stylus is a pen like interaction device that affects six degrees of freedom (DOF), three degrees for position and three for orientation) and which has a button for selecting or triggering certain events.

Some of the most common interaction devices used in virtual environments include the joystick, the stylus and the DataGlove. The DataGlove is a device commonly associated with virtual reality. It monitors finger movements in addition to the position and orientation of a hand. It consists of a lightweight glove with small lightweight sensors and fiber optic cable. The sensors and cable sense the bending of fingers while another type of sensor attached to the glove records the position and orientation of the hand [Foley et al].

The joystick also affects multiple degrees of freedom and can be used to achieve some type of natural interaction (interaction that is intuitive to the virtual application at hand); e.g. a joystick is a natural device for flying an airplane.

Six degrees of freedom devices are not always free moving devices (a device which a person holds and walks around with). They may be desktop based. An example of a desktop six DOF device is the Spaceball™, while the MITS Glove is an example of a six DOF free moving device. Some other devices for 3D user interfaces include the Cubic mouse and the Space Mouse™ [Foley et al][Zhai].

Due to the complex nature of interactions in 3D interactive environments and the need for multi-dimensional control, in addition to engineering challenges, creating multi-degree of freedom interaction devices is rather difficult [Zhai]. There are certain measures which are used to gauge how good an interaction device is. These measures include speed, accuracy, ease of learning, fatigue, coordination and device acquisition [Zhai].

An introduction to interaction, with a focus on natural and non-intrusive interaction, is given in the next section.

## 2.2 INTERACTION

Interaction provides the link between human intentions and computer processes. Simply stated, interaction allows a human to control and communicate with a computer and the programs running on the computer. Good human computer interaction is priceless to humans, because it allows a person to effectively and efficiently use and operate the computer. Good interaction achieves this is by allowing the user to express his intentions easily and meaningfully to the computer.

Interaction is achieved by means of input devices which make use of certain interaction techniques and tasks. The input devices constitute the physical means through which the computer obtains information (input) from the user, or the world around it. Interaction techniques deal with how the input devices are used to input information while the interaction tasks "classify the fundamental types of information entered with the interaction techniques." [Foley et al]. An interface is needed to couple the interaction tasks and techniques. The interface enables the computer to 'understand' the user's

actions. The quality of the user interface not only affects performance but also plays a role in determining whether or not a user enjoys using an application [Foley et al]. Figure 2.4 illustrates these different elements of interaction.

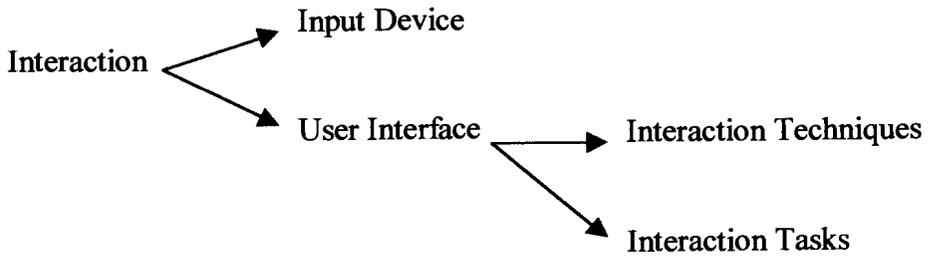



**Figure 2.4 Elements of Interaction**

The design of the input device and interaction techniques should make it easy for the user to convey his intentions to the computer. This will allow the user to focus on the task at hand rather than diverting his attention to focus on how to use the interaction device. To do this the input devices and interface used must reduce or minimize the user's mental workload. In essence the user should not have to worry about translating his intentions into computer commands.

Interaction or performing an interaction task should be easy to learn and once learned should require little further attention from the user. Natural interaction supports these objectives. Before natural interaction is discussed, a summary of interaction metaphors for virtual environments are given.

### 2.2.1 Interaction Metaphors for Virtual Environments

Complex applications in VR often require users to perform a variety of tasks such as viewing, analysis and manipulation [Laviola]. A categorization of interaction metaphors for virtual environments is illustrated in figure 2.5. It provides a foundation for developing new interaction metaphors [Laviola].

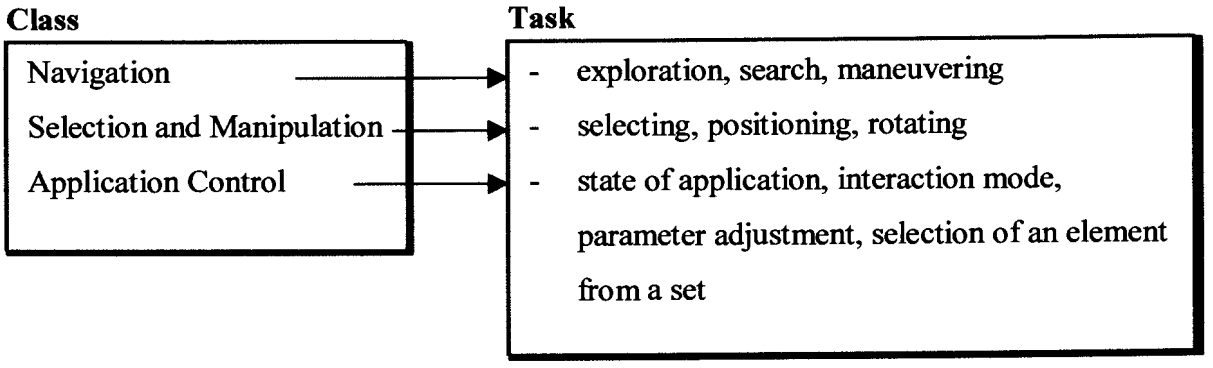



**Figure 2.5 Classes of Interaction for 3D User Interfaces**

Interaction and virtual reality are interdependent and complementary to each other. This is because while realistic interaction enhances the sense of presence in a virtual world, interaction relies on the multi sensory stimulus that a virtual environment provides. This is because without multi sensory input a user is less able to maintain situational awareness [Peterson]. A lack of multi sensory information also leads to a distorted and inaccurate perspective of the conditions in the users virtual environment. For interaction this results in increased task complexity, even for simple tasks because humans naturally perceive their environment through multiple senses and unconsciously combine all sensory input to create a single sensation. Virtual reality is therefore very conducive for aiding interaction as it provides a user with a variety of continuous sensory feedback.

Another science, beneficial for 3D interaction is artificial intelligence (AI). AI can be used to do motion and interaction understanding. Combining AI with VR also seems promising for application in the visualization of large data sets [Laviola].

### 2.2.2 Natural Interaction

Natural interaction in the context of this thesis refers to interaction that is intuitive, effective and appropriate. Natural interaction in this sense is largely determined by the specific task for which it is to be implemented. A natural interaction may stem from a real world interaction that forms an appropriate way of performing a specific task. For

example playing a virtual drum kit with real drumsticks using a tracker to monitor the position of the real drumsticks constitutes a natural interaction, while playing the virtual drum kit with a mouse does not. Natural interaction requires an interaction device, interaction metaphors, interaction tasks and techniques that are natural to the specific task. The interface must also suit the task in a realistic way. Natural interaction for abstract tasks or abstract environments will allow a person to interact with the virtual content at hand in an appropriate, effective and meaningful way.

Natural interaction has many advantages. Firstly natural interaction results in reduced mental workload. This influences presence, which in turn affects immersion. The reduction in mental workload frees ones attention to focus on meaningful interaction with the content at hand. Meaningful interaction and felt presence in turn lead to improved virtual task performance [Peterson]. This domino type effect of the advantages of natural interaction is illustrated in figure 2.6.

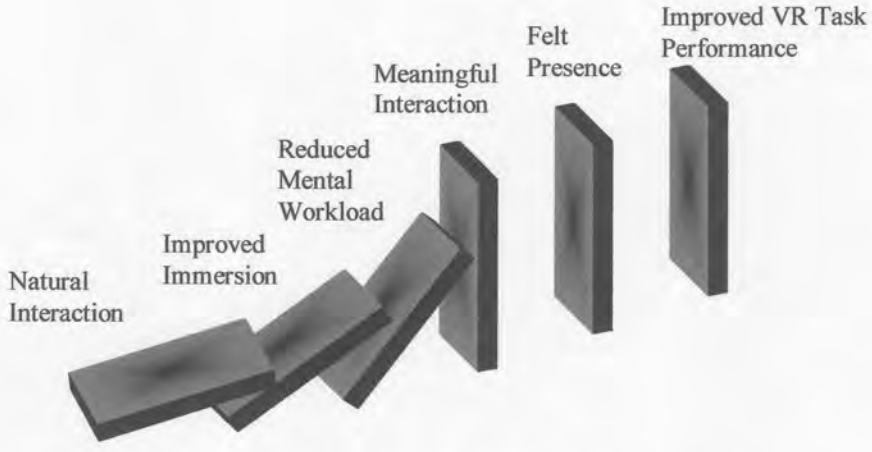



**Figure 2.6 Effects of Natural Interaction**

When interaction is natural the user interface becomes transparent [Peterson]. This transparency is an important benefit because it does not encumber the user with having to determine a command sequence to express his intentions. An interface that is nearly transparent is Jot, a developing 3D modeling system for drawing and erasing. The vision for the Jot interface is that it will be like that of a simple pencil and sheet of paper, where

interaction is simply and intuitively defined by the user picking up a pencil and drawing on the paper. The Jot interface allows one to move seamlessly from tool to tool, with the individual tools physically and perceptually matching the tasks they are to perform [Macedonia & Rosenblum].

There is an old saying that, "One should use the best tool for the job". This saying also applies to the world of human computer interaction, in which a user is only as good as the interaction device he uses. Consider for instance deciding whether to use a mouse or a keyboard for pointing at and selecting objects on the screen. The mouse is clearly the more appropriate tool for this job. The appropriateness of a device depends on the interface and task to be performed [Peterson]. For certain problems better results may be obtained if interaction devices are built and used which draw from a person's task-specific abilities rather than using a general tool such as a mouse [Paley].

Another consideration for interaction is to broaden the toolkit rather than limit interaction to one general tool. This has the advantage that it will allow the user of the interaction devices an ease of expression. One way of creating such a toolkit is to use multimodal interfaces. Because these interfaces combine different types of input they overcome limitations experienced with using only a single form of input. An excellent example of this is combining voice and gesture. Objects can be handled directly using gestures (continuous data), while voice provides a discrete type of input that can be used for signaling. Together voice and gesture provide a more powerful interface than either voice or gesture by itself [Bilinghurst].

An input device is needed that will accommodate natural interactions. Natural interactions are commonly not to be limited to one or two dimensions. They are usually actions that cover some region of 3D space. Thus the input device for natural interaction will have to be a 3D input device. Another requirement for such a device is that it must be able to monitor a variety of interactions unobtrusively. A means will be needed to identify, interpret and translate the actions and movements the input device monitors into meaningful terms or commands for the computer.

### 2.2.3 Non Intrusive Interaction

Many of the input devices discussed to this point are in one way or another physically tethered to the user. There is however a need for non-contact devices and device free interaction. "A device-free interaction system is one which does not require the user to wear any special equipment, be it sensors or specially marked clothing" [Van den Bergh].

In this text, this definition is extended and slightly relaxed with the term non-intrusive interaction. *Non-intrusive interaction is interaction which does not interfere with the users freedom of movement, be it by wires or even wireless objects that need to be connected to the user or the object being used for interaction.*

This definition includes the use of interaction with objects which positively contribute to the interaction. An example of non-intrusive interaction using an object is, playing tennis by using a thetherless real tennis racket (with no sensors or specially marked attachments) in a virtual tennis game. The tennis racket makes the interaction and whole virtual environment seem realistic and because it has no wires attached maintains the definition of non-intrusive interaction even though the user has to hold the tennis racket. The tennis racket itself is appropriate and helpful for interaction in the context of this environment.

Non-contact devices are usually implemented using video. One very interesting implementation of a non-contact device makes use of luminous light and optics in the luminous room using the I/O bulb [Underkoffler et al]. The interaction in this system allows for the direct manipulation of objects. Furthermore the objects used for interaction (props) tend to be well suited to the application. The system allows for a very natural style of teatherless interaction. The applications developed to this point have, however been limited to a flat surface such as a wall or desktop. This means that at this stage these systems do not cater for 3D interaction or tracking.

A multidimensional non-intrusive interaction system would be invaluable for 3D interactive display systems. One of the most promising devices for implementing such a tracking and/or interaction device is the video camera. By using computer vision techniques it is possible to turn a video camera (stereo video camera or multiple cameras) into an interaction or tracking device that can achieve natural, three-dimensional and non-intrusive interaction.

Next a review of systems that use video cameras and computer vision to achieve non-intrusive, 3D and natural interaction are described.

### 2.2.4 Examples of Natural and Non-Intrusive Interaction

Frans van den Bergh developed a real-time vision based locator system that tracks the center of a hand. It identifies different gestures with a high degree of independence. The system operates at 25 frames per second. It currently only tracks the position of the hand in 2D but can easily be extended for 3D tracking by using multiple cameras. Included in his work are several different image processing and computer vision techniques. One of the noteworthy contributions in this work was the development and implementation of a fast chroma-keying technique [Van den Bergh].

Another system that uses CV to accomplish natural and device free interaction is covered in [Segen & Kumar]. In their work they have developed a vision based input interface that aids tasks such as 3D navigation, object manipulation and visualization. The system makes use of two video cameras focused on the desktop. The common field of view of these cameras defines a 3D volume in which 3D interactions take place. The cameras run at 60Hz. The system recognizes three simple gestures and tracks the movement of a user's thumb and index finger (hand). They have, in addition to a virtual fly through, used their interaction system in several applications including a 3D-scene composer. Their scene composer application is of special interest because it uses gesture driven input and allows for 7DOF. Furthermore the interaction in this system is very natural. It allows users to produce and interact with complex 3D-scenes by selecting and manipulating simple

objects in an intuitive way. This application supports multimodality. The different modes it supports include:

- an object mode, which allows for the selection of objects,
- a gripper-mode for the control of a graphic robot hand to manipulate objects,
- a draw mode that lets the user draw in 3D by tracing a curve with the point of the finger, and
- menu control which is done by means of different gestures.

Cameras and computer vision are also used to achieve seamless and non-contact interaction in the virtual environment of the Perceptive Workbench [Leibe et al]. This system's computer vision capabilities support object and hand tracking as well as gesture recognition. It also allows real objects to be used as natural interactors which may be inserted spontaneously into the environment. These objects allow users to control aspects of the virtual environment in a natural way.

The system works in the environment of the perceptive workbench using two cameras for 3D recovery. One camera is placed under the projector with infrared filters and one at the side of the workbench. The filters are used to capture infrared light only. Two infrared illuminators are placed next to the camera and seven of these illuminators are placed above the workbench. These lights are computer controlled and each light casts a shadow onto the surface depending on where the objects are positioned on the table. The second camera is a colour camera and captures a side-view of the scene. This camera is used to evaluate pointing gestures. Infrared is used because it does not interfere with the scene. The system operates at 12-18 frames per second (fps). The calculation of 3D position and orientation of a pointing gesture is done as follows: The camera below the desk determines a plane in which an object lies based on the light source and where it locates the image of the object. From locations of the start and end of the arm seen in the second video camera two lines are determined and their intersections with the plane found. This yields the two 3D positions of the arm from which the orientation is determined. This system also supports 3D reconstruction of objects placed on the surface of the workbench.

In each of the above examples computer vision is essential. In the following section a prelude to computer vision is given since it is the necessary link between the physical device (video camera) and applications that make use of images captured by the camera. Since CV is such a vast field only those aspects of computer vision that contribute to using a video camera to develop a 3D non-intrusive interaction system are covered. One other consideration for using computer vision for interaction is that there should be negligible delay and lag between when a user performs an action and the time of the computer's response; i.e. a fast response time is necessary (real-time). This means that fast computer vision algorithms are needed for interactive computer graphics applications [Freeman et al].

## 2.3 COMPUTER VISION

Computer vision (CV) is the attempt to mimic the human vision system. This endeavor normally entails, on the hardware side, the use of video cameras and a reasonably fast computer. The software side consists of the computer vision process that prepares the images captured by the camera, extracts certain relevant details by analyzing the image and from these features infers certain information about a scene.

Computer vision has many uses such as robot vision, machine vision and interactive computer graphics. Computer vision is necessary to bridge the gap between raw images and meaningful aspects within the images that are needed for the applications. CV algorithms need to be fast. They should also be reliable and robust (work against complex backgrounds and settings). Having to meet all these requirements places serious burdens on the algorithms. Several of today's CV algorithms tend to be both complex and slow. However, by taking into account the context of an application the complexity of interpreting an image can be significantly reduced. In addition to these optimizations, there are several fast and simple algorithms that make real-time computer vision tangible [Freeman et al, 1998]. Another way to simplify CV algorithms is to place certain constraints on the visual environment, e.g. the use of a constant blue background simplifies the vision algorithm and makes image processing easier.

### 2.3.1 Overview

Computer vision is a process in which the raw data captured by a camera is analyzed, interpreted and made sense of. The process consists of four distinctive steps:

i) image acquisition,
ii) image processing,
iii) image analysis, and
iv) image understanding

In the image acquisition step, the camera captures an image (frame) and stores it in memory as an array of pixels. A pixel consists of either colour components for colour images or in the case of black and white camera's, an intensity value. Figure 2.7 illustrates a pixel, the components it consists of and a frame or image which is represented by an array of pixels.

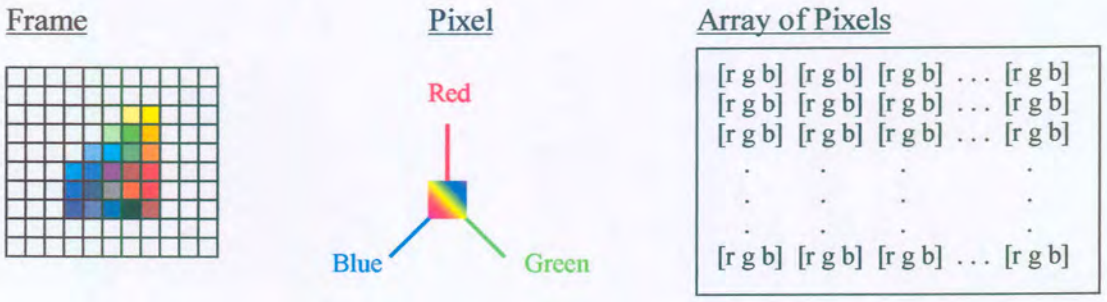



**Figure 2.7 Illustration of a Pixel and a Frame and their Representation in Memory**

An image represented simply by numbers makes little sense and may also be dappled with noise. To deal with noise and prepare the image for analysis the image processing (or image enhancement) phase is needed. This phase may include preprocessing, averaging, contrast improvement and the removal of reflections. After an image has been touched up and its quality improved, image analysis is applied to find features of objects such as edges, corners and boundaries. These key features are illustrated in figure 2.8.

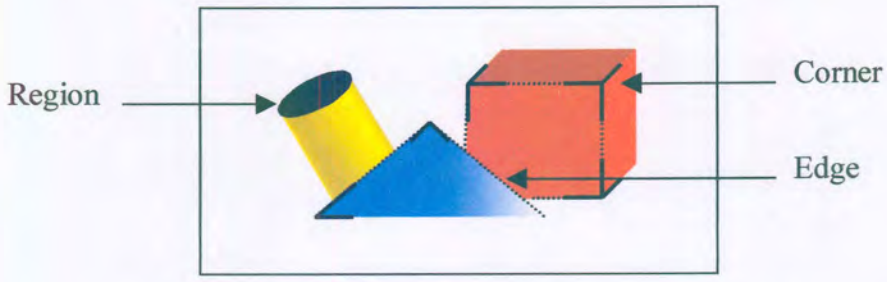



**Figure 2.8 Features Obtained During Image Analysis**

Surface smoothing is a techniques used in the image processing phase to remove irregularities or shading. It can be done before edge detection [Turban]. Edges, corners and region information constitute a more meaningful and higher level of knowledge of an image than the low-level pixel array. The features that are extracted during image analysis are used in the last phase of the process, i.e. the image understanding phase.

In the image understanding phase, AI techniques are used in an attempt to make sense of the analyzed image by identifying shapes and objects and relationships between them.

A vast amount of processing is done in the computer vision process. For certain applications such as interactive computer graphics fast and simple vision algorithms are necessary to meet the real-time requirements of these applications. Several such algorithms exist, such as image moments, orientation histograms, fast optical flow and normalized correlation [Freeman et al]. Image moments in particular are very useful because they provide global summaries of information such as the position, orientation and dimensions of an image.

In addition to image moments, other image analysis and processing algorithms are necessary for extracting relevant features from a picture.

### 2.3.2 Colour, Light and Infrared

It is more advantageous for the image processing and analysis algorithms to rather use colour images than simple black and white images. This is because colour images contain more detailed edge information and colour presents a means of locating and segmenting regions of interest.

Colour is often used to divide an image into logically associated regions or to discard regions that are not significant. Examples of algorithms that use colour to do this are background subtraction, chroma-keying and vector-keying. Chroma-keying is an algorithm that removes elements of a particular uniform coloured background from an image. Background subtraction removes pixels that correspond to a certain background and vector-keying is used to identify a region in the foreground of a particular colour [Van den Bergh].

Van den Bergh's work provides a good illustration of the use of colour in computer vision applications. By using colour he extracts the silhouette of a hand for gesture recognition in his implementation of a device-free interaction system. Another useful contribution made, is that of a fast software chroma-keying technique. In his work he also implemented and used a vector-keying algorithm. The vector-keyer has the advantage that it is not sensitive to background complexity (at least as long as the background colours are different from the trained "key colour") [Van den Bergh].

[Garcia & Tziritas] present a novel method of detecting human faces in colour images. Their method works for unconstrained scene-conditions; i.e. it works within the presence of complex backgrounds and uncontrolled illumination. The use of colour in this method is clearly evident since they use approximations of the YcbCr and HSV skin colour subspaces to obtain similarly coloured regions by applying colour clustering and filtering. These skin colour regions are then merged to provide potential face areas and wavelet packet decomposition is used to detect human faces.

Taking a slight detour from colour, although it also makes use of the element of colour in images, is the use of light. In their work [Lane & Lalioti] use colour and intensity to locate a luminescent green object in a constrained environment (a room with low light). Their system implemented in CyberStage tracks a small torch. In projection based systems there are insufficient levels of light to track coloured objects. In such environments it is necessary to use luminous objects or infrared.

In their system [Davis & Bobick] use infrared to extract the silhouette of a person in an interactive virtual aerobics system. Their approach overcomes certain difficulties encountered with the more common chroma-keying, background subtraction and rear-light projection methods. Another advantage of using infrared light is that it does not damage the effect of presence because the user can not see it.

Another approach that also makes use of near-infrared light with the same advantages, is the work done by [Leibe et al] in the virtual environment of the perceptive workbench. In their system near infrared light is used to either cast shadows or to illuminate the scene (this does not affect immersion since this light is invisible to the human eye). The computer vision aspect of their work uses a combination of intensity thresholding and background subtraction to capture interesting regions in an image. Once these regions are found they are classified by using a variety of features such as area, eccentricity, perimeter, moments and contour shape. Their work illustrates how knowledge about the context of an application and the environment are used to simplify the vision algorithms, that is they take into account that the user's arm will always cross the front border of the desk. The image analysis phase therefore simply needs to search the portion of the image that corresponds to this border to locate the origin of the arm.

### 2.3.3 Meaningful Features

After an image is segmented and regions determined it is necessary to extract features from these regions for classification, tracking or even stereo matching. Some features of

interest include, edges, corners, points-of-interest, skeletons, orientation and the center of gravity of an object.

Feature-based vision systems rely upon the identification and tracking of good features from frame to frame [Shi & Tomasi]. Points-of-interest (POI) are points that are unique and differ significantly from neighboring pixels. These points are useful for detecting subsequent stereo correspondences. Such points are selected by considering the uniqueness of edge pixels. To define the uniqueness of a pixel a correlation value is used. Using such points of interest works better than using implicit features such as corners as points of interest [Zhang & Gimel'farb].

Edges provide valuable low level information for computer vision algorithms. They are important for subsequent processing steps such as edge-based image segmentation and edge-based stereo matching. There are a myriad of edge detection algorithms and operators. A comparative study on color edge detection is given in [Koschan] and a review of different approaches to edge detection is given in [Smith & Brady].

Corners are also very important features of objects. There are several different algorithms for finding corners; a review of these is given in [Smith & Brady]. Corner detection algorithms are based on the way in which a corner is defined. There are two main categories into which corner finding algorithms are classified, namely low-level and scale based approaches. One approach from each of these camps is considered.

SUSAN (Smallest Univalue Segment Assimilating Nucleus) is a low-level approach that uses non-linear filtering to define those parts of an image that are closely related to each individual pixel. The SUSAN feature-detectors work by minimizing local image regions [Smith & Brady]. SUSAN is used for edge and corner detection as well as for structure preserving noise reduction (thinning). Thinning is a process in which pixels belonging to an object which characterize the shape thereof are identified [Parker et al]. The methods are fast, accurate and noise resistant. However, the corner detector can only find a single corner for each mask position. This results in closely spaced corners being omitted when

a large mask size is used [Zhang & Gimel'farb]. Nevertheless, it must be said that SUSAN is a beautiful feature detector.

A scale-based approach that is of particular interest makes use of Electrostatic Field Theory (EFT). This corner detector is based on the observation that an electrostatic field concentrates around pointed conductor edges. The EFT-based algorithm provides a means of determining not only corners but also skeletons of objects. It provides solutions to many problems that occur in existing skeletonization algorithms such as connectivity and thinness. This algorithm does corner finding, thinning and skeletonization all in one framework [Grigorishin et al]. On the down side for the algorithm to work it needs to calculate the potential distribution inside an object. This is a time consuming process [Abdel-Hamid & Yang, 1994]. A skeleton represents a powerful means of representing a shape and captures both boundary and region information of an object.

After an object is identified, essential features must be determined and points of interest matched.

### 2.3.4 Tracking

Once an object is positively identified, it can be tracked. One implementation of a tracker, tracks the center of gravity (centroid) of a user's hand using a window which follows the centroid [Van den Bergh]. This approach is discussed in more detail in chapter 3. The system performs at 25 frames per second and objects are tracked at speeds of up to 11.7 m/s in the horizontal direction and 8.8m/s in the vertical. Using a tracking window reduces CPU processing time by reducing the region to be processed and searched by the CV process.

[Freeman et al] use image moments to track an object in applications where the size of the object being tracked fills a large region of the image (large object tracking). In one application they implement a system using image moments which tracks a user's hand that fills a large portion of an image. The position and the orientation of the hand are used

to control a robot. They use normalized correlation for small object tracking and illustrate it in an application in which a user's hand and gesture are tracked and used to adjust various television controls [Freeman et al]. They also use optical flow to indicate movements or gestures. Their fast optical flow algorithm is used in a Decathlete application in which a user pantomimes athletic actions in front of an Artificial Retina Chip. The actions are used for interaction with the computer.

One method of note is the work done by [Blake & Isard] in which they have gone beyond simply tracking centroids, key-points and non-polyhedral objects. Their system tracks the curved silhouettes of moving non-polyhedral objects such as hands and lips at 50Hz. They illustrate this tracker in a variety of applications in which they track both rigid and non-rigid motions. One of these applications is a 3D mouse, which uses only a single camera. This is achieved by tracking non-rigid motions. Their approach makes use of deformable models, B-spline curve representation and control theory. They have implemented two algorithms. The first applies a Kalman filter to curves to track rigid and non-rigid motions. The second algorithm is a "system identification algorithm" that uses adaptive control theory and "maximum likelihood estimation". The effectiveness of the algorithms is illustrated in that they track hands even in cluttered backgrounds.

Tracking multiple moving objects at frame-rates is not yet a completely solved problem [Corbett]. There are several algorithms and methods used for multiple object tracking. [Corbett] proposes a system that integrates optical flow and model matching to achieve real-time tracking of multiple objects. This system works for constrained environments and operates at 0.654 seconds/frame in real-time. Occluding objects, objects of similar size and objects with similar colour spectra cause problems with this approach. For many multiple object-tracking systems there is an accuracy verse efficiency tradeoff.

By combining the above techniques computer vision can be used to locate and track an object. However, there is one serious problem that computer vision encounters when trying to implement a 3D locator and tracking device. This problem will be discussed

next. This is followed by an introduction to the solution to this problem which concludes the background for this thesis.

### 2.3.5 The 2D to 3D problem

A single picture taken by a camera does not capture depth information; i.e. the picture is flat or 2D. This means that it is difficult to determine the distance from one object in a picture to another. It is also difficult to determine the distance an object in a picture is from the camera simply by looking at the picture. This inherent property of images taken by a normal camera presents a problem when it is necessary to determine 3D information from a 2D picture [Turban][Lane]. This is illustrated in figure 2.9, which shows a perspective view (left) of two planets, in which the gray moon appears larger than the orange sun. Looks can be deceiving, as depicted in the picture on the right in which the relative sizes and position of the planet and moon are shown. Simply by looking at the 2D picture on the left, it is both difficult to determine the distances the planets are from the camera and the relative sizes of the planets.

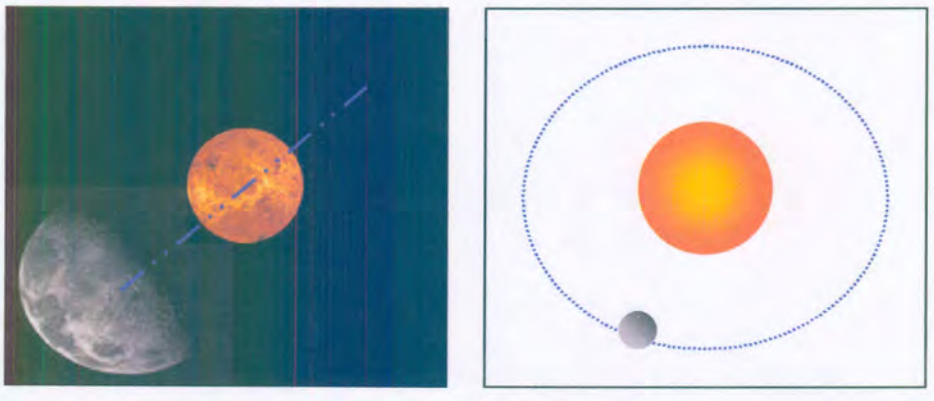


**Figure 2.9 Illustration of the 2D to 3D Problem**

**(a) Single Camera Solutions**

There are ways to estimate depth from a single image. These methods accomplish depth detection by means of analyzing shaded regions or by searching for hints within the image that reveal the necessary depth information, e.g. searching for distortions in lines.

A method implemented at Massachusetts Institute of Technology determines depth by examining a scene under different lighting conditions [Turban].

Depth detecting methods via other schemes (single camera methods) require additional processing to be done to analyze the scene being studied and to extract the depth hints. This extra processing adds to already high image processing costs. This may tend to be a problem in real time applications that require a high frame rate. In addition, single camera techniques do not fully capture three-dimensionality; they just give an idea of depth, i.e. in essence they are only 2.5D (somewhere between 2D and 3D) [Turban].

**(b) Multiple Camera Solutions**

The conventional solution to the 2D to 3D problem is to use a stereo camera or two cameras which are placed at a fixed distance from each other. 3D image reconstruction is done by matching points in these different views of a scene. Other approaches use multiple cameras to capture multiple views of a scene. By combining the different views a depth map is constructed and the relative 3D position of an object calculated.

Stereo cameras and multiple camera methods have several problems and complexities such as the following:

i) multiple camera techniques are expensive. There is additional cost for additional cameras and video equipment. Stereo cameras also tend to be expensive,
ii) these methods are complex,
iii) setting up the system can be tedious and requires additional thought and work, e.g. linking several cameras to a single computer or setting up a network of computers and cameras is complex and requires extra effort,
iv) the pictures from the different cameras need to be synchronized,
v) having multiple cameras is not necessary for applications other than 3D determination,
vi) additional bandwidth and memory are needed for the second camera,

vii) processing costs will be higher for the additional processing and bandwidth of the second video stream, and

viii) special software may be required.

Due to the complexity and expense of these methods they are not commonly used, and for applications such as real-time interaction they are rarely heard of. Stereo cameras and multiple camera solutions tend to be used only in applications where their use is critical and in which the expense and complexity can be justified [Turban].

However, there is an alternative to these methods that is accurate, less costly and less complex. It only requires the use of a single camera (lens) and one or more mirrors.

## 2.4 CATADIOPTRICS

Catadioptrics refers to the optics of a combination of mirrors and lenses, or more scientifically a combination of reflecting and refracting surfaces (dioptrics is the science of refracting surfaces and catoptics is the science of mirrors). A catadioptric sensor consists of some lenses (video cameras) and mirrors. During the design of a virtual drum project in the year 2000, it became evident that 3D interaction can be implemented by using a mirror in conjunction with a video camera.

One of the main uses of catadioptrics is to increase a camera's field of view by using curved mirrors. While panoramic cameras are conventionally used to yield a large field of view, multiple panoramic cameras may be used for 3D computation [Svoboda & Pajdla]. This thesis however focuses on the use of catadioptrics for accurate 3D determination using planar mirrors and a single video camera.

Catadioptrics presents an alternative solution to the 2D to 3D problem. It is a means of using just a single video camera and some mirror(s) to obtain stereo images [Gluckman & Nayar]. Using catadioptrics in this way is referred to as catadioptric stereo [Gluckman & Nayar]. Several researchers have implemented working catadioptric stereo systems, for

instance [Gluckman & Nayar] implemented a real-time catadioptric stereo sensor as an alternative to conventional stereo. Figure 2.10 is a picture of the camera used in their camera-mirror sensor.

**Figure 2.10 Catadioptric Stereo Sensor**
***(Picture from [Gluckman & Nayar] )***

Much of the theoretical work for catadioptric sensors and stereo has already been done including issues pertaining to catadioptric image formation such as the shape of mirrors, the resolution of cameras and focus settings of the cameras [Baker & Nayar]. In addition the complete class of mirrors that can be used with a single camera have been derived. The geometry and calibration of catadioptric stereo using planar mirrors have also been researched and a class of novel stereo sensors (catadioptric stereo) designed that avoid the need for synchronization, rectification and normalization. In their work they have examined how scan-line correspondences of rectified catadioptric stereo sensors can be used to avoid the additional computational cost and image decay associated with rectification (Stereo images are rectified once the scan lines of the images have been aligned with the epipolar lines). Rectification is beneficial in real-time stereo systems [Gluckman & Nayar, 2000].

In the above research it is shown that catadioptric stereo is more advantageous than conventional stereo, in that -

i) the system parameters are identical. Parameters such as gain, lens distortion, focal length and pixel size are identical because only a single camera is used. This is also beneficial to the stereo matching algorithm.

ii) calibration is easier. A direct consequence of using only a single camera is that only a single set of calibration parameters are necessary. Moreover these parameters are constrained by planar motion and therefore only ten parameters are needed instead of the sixteen that are required for traditional stereo cameras.

iii) synchronous data acquisition is not an issue. For conventional two-camera stereo the cameras need to be synchronized. This is not necessary when only a single camera is used.

An application that illustrates the use of catadioptric stereo is seen in the work done by [Delamarre & Faugeras] in which they address the difficult problem of determining the pose of a hand in a sequence of stereo images. Using a stereo correlation algorithm a scene is reconstructed in 3D from the images obtained by a stereo camera. A model of a hand is then fitted to the 3D reconstruction from which the pose of the palm and fingers are determined. In the implementation of their system they use catadioptric stereo (one camera with mirrors).

## 2.5 SUMMARY

In the literature study several relevant topics are introduced that pertain to the creation of a multidimensional non-intrusive interaction device that supports natural interaction. A discussion of the key technologies for immersive virtual reality (IVR) is given in section 2.1 IVR provides an environment that benefits from the use of a non-intrusive interaction device and which provides an environment that is in turn beneficial to interaction. Aspects of interaction are presented in section 2.2 and a discussion on natural interaction and its implications upon users and their effectiveness is given. Non-intrusive interaction is also introduced and the advantage of using computer vision for such interaction

presented. A few existing applications that use computer vision to achieve natural and non-intrusive interaction are discussed. An overview of computer vision is presented in section 2.3 Computer vision is fundamental for using a camera to achieve non-intrusive interaction. Certain common techniques for extracting meaningful features from an object and object tracking are also presented in this section. The 2D-3D problem is then stated and conventional solutions to this problem are discussed. Section 2.4 presented catadioptric stereo as an alternative solution to the 2D to 3D problem.

This concludes the literature study and background review of the work pertaining to this thesis. In the next chapter a deeper look will be taken at the theory of catadioptrics and other aspects such as computer vision algorithms and tracking.

# Chapter 3

# Theory

In this chapter a theoretical overview of key topics introduced in chapter two are presented. The heart of this chapter is concerned with catadioptric stereo (Reflections). This is followed by the theory of those computer vision algorithms used in the practical implementation of the Reflections system. The chapter ends with an overview of two different tracking algorithms for computer vision.

## 3.1 REFLECTIONS

The core focus of this thesis is Reflections (catadioptric stereo) and how it can be used to create a three-dimensional interaction device. For this reason an in-depth coverage of catadioptric stereo is necessary.

Catadioptric stereo works on the premise that mirror reflections can be used to obtain stereo images of a scene from which a depth map of the scene is reconstructed. Stating this more simply, two views of an object or point from different positions and angles can be obtained by using only a single camera and some mirrors. From these views it is possible to calculate the 3D position of that object or point.

The theory of this approach is extensive and for this reason an overview of the approach will first be given. This overview will then be followed by a more detailed discussion of the different aspects covered in the overview.

### 3.1.1 Overview

The Reflections system presented in this thesis consists of the apparatus and the methods for calculating the 3D position of a point and the 5D and 6D information of an object. The method for calculating the 3D information requires a computer vision algorithm. The particulars behind the computer vision algorithm will be discussed in greater detail in section 3.2.

The apparatus required for a Reflections system includes the use of a video camera and one or more mirrors. The apparatus is discussed in greater detail in section 3.1.3.

The way in which this apparatus is used to calculate 3D is as follows: The camera and mirror are placed in such a way that the camera sees an object directly. The camera also indirectly sees the object in the mirror (the reflection of the object). By cleverly using some mathematics the 3D position of a point on an object is calculated. This is illustrated in figure 3.1. The way in which the method works is given more attention in section 3.1.2

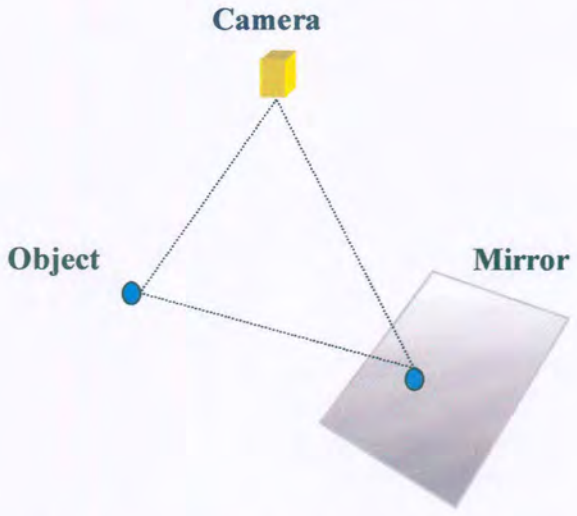



**Figure 3.1 Illustration of a Camera and Mirror Setup**

This approach for calculating 3D will work if the camera sees the two views of the object. The region for which the camera can see the indirect and direct view of the object simultaneously determines the interaction region. The interaction region is a volume in which an object moves and for which the method can calculate the 3D position of the object. This volume is determined by the size of the mirror, the size of the view angle of the camera and the relative positions of the camera and mirror. This volume is limited to that region which lies in the intersection of the camera's field of view and the indirect field of view (determined by the mirror). This is explained in more detail in section 3.1.5. Figure 3.2 illustrates the region of interaction as the intersection of two fields of view.

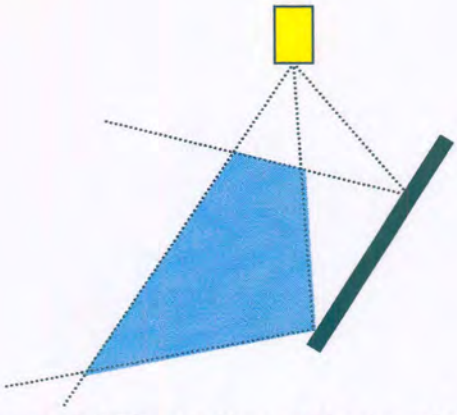


**Figure 3.2 Illustration of Region of Interaction**

The apparatus may be installed in a variety of settings ranging from a desktop to a projection-based system. When integrating the approach in a Cave the floor projection mirror is used. A discussion on different installations for a Reflections set up is given at the end of chapter 4.

The object to be tracked depends on the environment in which the system is installed. In low light environments like the Cave the tracked object used is a luminous object such as a fluorescent tube or torch. In environments with suitable light, a coloured object is tracked. See section 3.1.3 (c) for a more detailed discussion on the object to be tracked.

Once the equipment is in place the 3D information is calculated. This requires three phases:

- a calibration phase,
- an image analysis phase, and
- the actual 3D calculation phase.

The calibration phase involves the installation of the apparatus and taking measurements of the relative positions and orientations of the camera and mirror. This information and the focal length of the camera are required for the 3D calculation. Once the calibration phase is complete the measured information is used in a preprocessing step which is performed only once in the calculation phase. In this preprocessing step the plane equations which mathematically represent the mirrors are determined. Depending on the

approach used for 3D calculation, other information is also calculated in this step. This preprocessing step is discussed in section 3.1.6 (d).

Once the calibration and preprocessing steps are completed, a sequence of video frames is started and an image analysis phase applied to each individual frame. The image analysis phase finds and matches corresponding points in the two views. This information is then used in the 3D calculation phase. Section 3.2 describes in detail the different aspects of this phase. An image taken by the camera in the setup depicted in figure 3.1 is illustrated below in figure 3.3. The positions where the object is found in the image are determined in the image analysis phase.

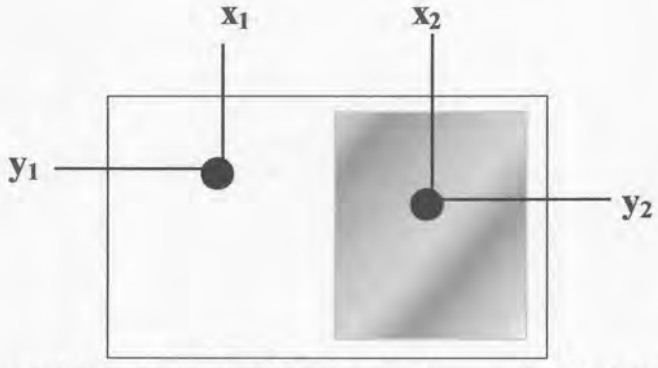



**Figure 3.3 Illustration of Where the Image Analysis Phase Locates the Object and its Reflection in an Image**

The calculation phase consists of the mathematics used to calculate the 3D position of the object. Two methods have been developed. The first is a trigonometric approach which uses geometry and trigonometry to find the 3D information. The second is an algebraic approach which makes use of linear algebra to determine the position of the object. Both the trigonometric approach and algebraic approach are discussed in section 3.1.6.

When calculating 6D information, multiple 3D points on an object are determined and then used to determine the angles of rotation of the object. The method for calculating the 6D information is presented in section 3.1.7. The theory of Reflections is described in detail below.

### 3.1.2 Approach

In section 2.3, the 2D to 3D problem was introduced, namely that pictures taken by conventional cameras lack depth information. This problem is overcome by cleverly using a mirror with a camera. The following simple example illustrates how this method works.

Consider taking a picture of a blue ball in a certain scene. Suppose that there is a mirror in this scene. The mirror is placed in such a way that the camera sees the ball directly as well as the reflection of the ball in the mirror. In a single picture taken by the camera two views of the ball from different angles and positions are seen. From such an image it is possible to calculate the 3D position of the ball. Figure 3.1 illustrates a 3D view of such a camera and mirror setup. Figure 3.4 shows the top, side and front views of the camera mirror setup in figure 3.1. Although this example is simple, it effectively and illustratively highlights how the method works. This example will be used later to illustrate other aspects of the approach.

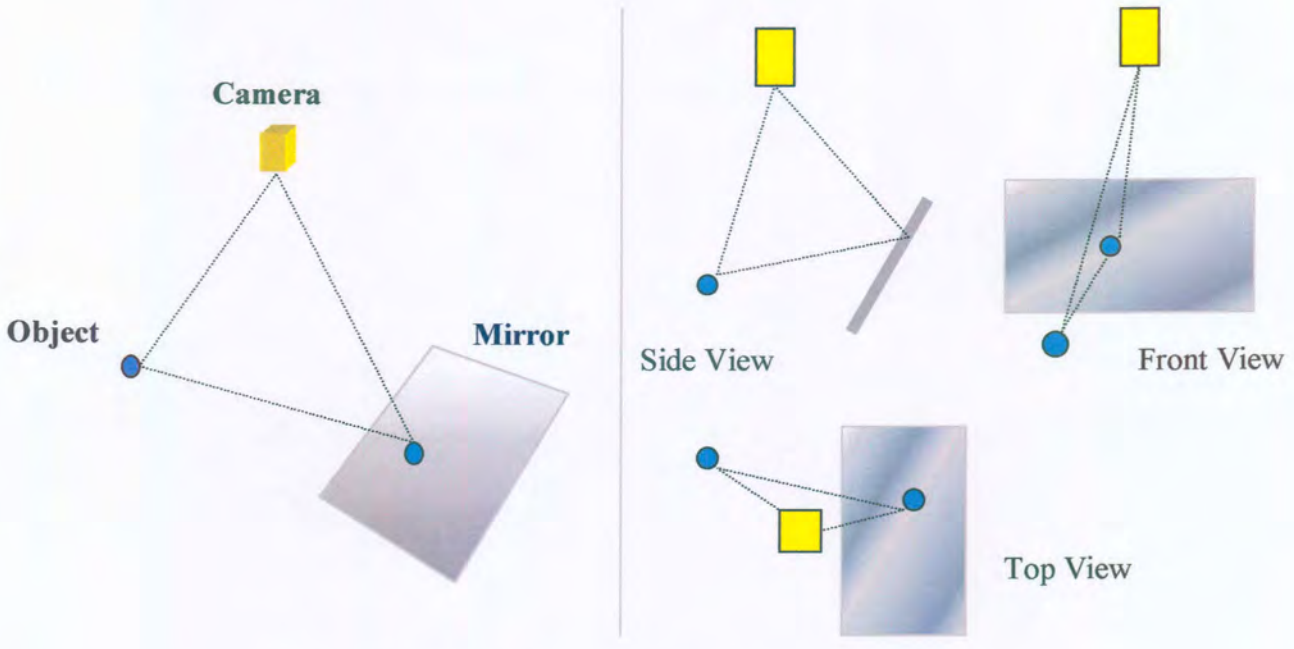



**Figure 3.4 Views of the Camera and Mirror Setup**

The approach taken by conventional catadioptric stereo to describe the geometry of a camera mirror setup provides insight into certain functional aspects of the method. The geometry of a catadioptric stereo system that uses two mirrors is described next.

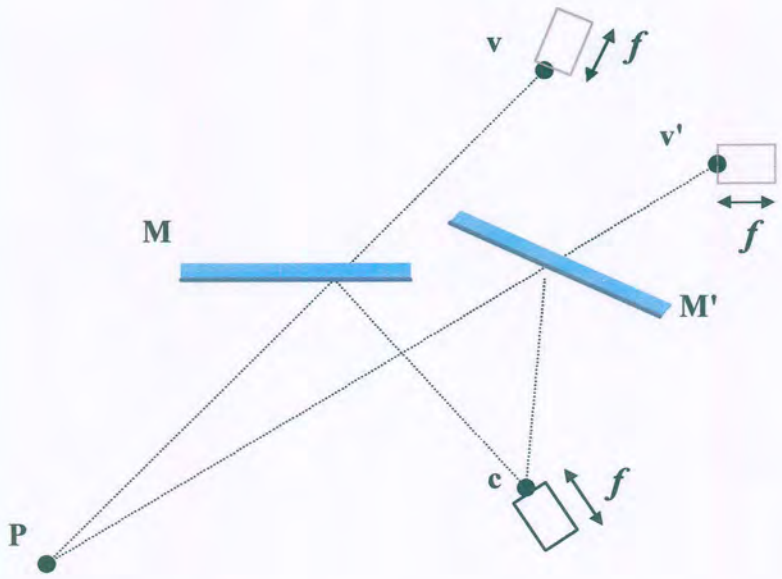



**Figure 3.5 Geometry of Catadioptric Stereo**

Figure 3.5 shows two planar mirrors, placed in such a way that a scene point **P** is imaged as if seen from two different viewpoints, **v** and **v'**. By reflecting the camera pinhole **c** about each of the mirrors, the location of the virtual pinholes of the two viewpoints is found. Reflecting the optical axis about the mirrors yields the orientations (optical axes) of the virtual cameras. The focal length of each of the virtual cameras is $f$, which is the focal length of the real camera. The two virtual cameras are located the same distance the two mirrors are from the pinhole of the camera and are oriented according to the rotations of the mirrors relative to the optical axis of the camera.

Both the Reflections approach taken to describe the method and conventional catadioptric stereo use a ray model of light and a pinhole camera model to represent the camera. Both approaches assume mirrors with perfect reflection. Catadioptric stereo makes use of the idea of virtual cameras, whereas in this thesis the path light travels is used to explain and illustrate how Reflections calculates the 3D information.

### 3.1.3 Relative Orientation

In conventional stereo cameras the orientation of one camera is given relative to the other. This requires specifying three degrees of freedom for rotation and three for translation. In catadioptric stereo constraints exist which limit the relative orientation of one mirror with the other to only five degrees of freedom (three for rotation and two for translation). Irrespective of the placement and orientation of the mirrors, there are only two degrees of freedom for translation because the motion of the translations is restricted to a plane. This constraint exists because the virtual cameras are found by a pure rotation about the screw axis. The screw axis is the intersection of the two planes representing the mirrors. Figure 3.6 illustrates the screw axis and the planar motion constraint.

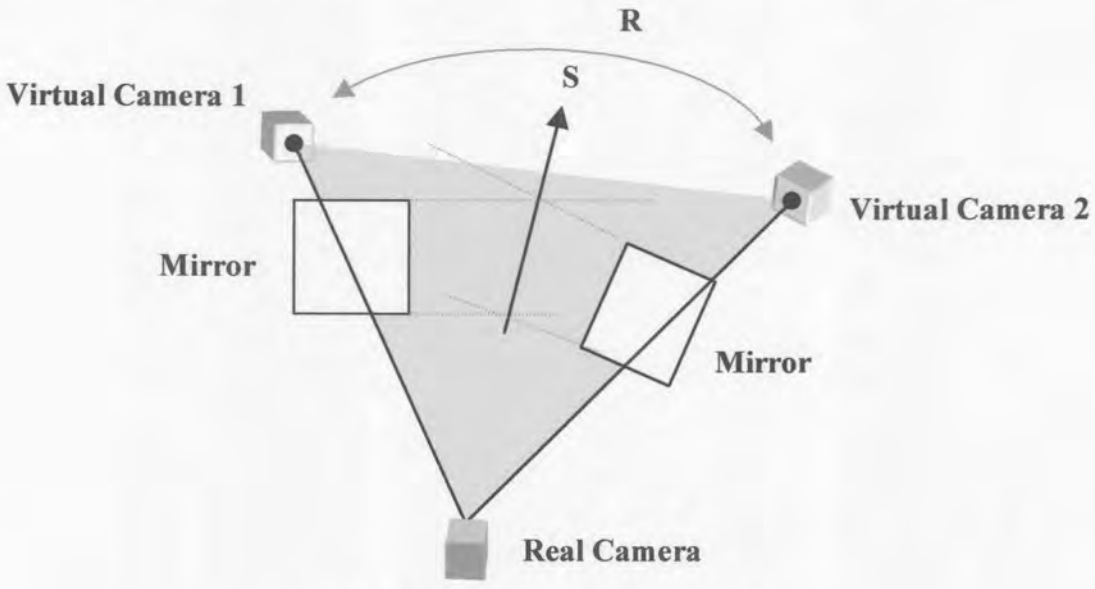



**Figure 3.6 Relative Orientation**

This pure rotation, **R**, about the screw axis, **S**, limits the motion of the virtual cameras to a plane that lies perpendicular to the screw axis. The plane of motion is that plane in which the real and virtual camera centers lie. This plane of motion lies orthogonal to the screw axis. The line where this plane intersects the image plane is called the horizon line.

The intrinsic parameters for catadioptric stereo remain the same for the two virtual cameras. This is because the virtual cameras have the same intrinsic parameters as the real camera. Because there is only one camera for catadioptric stereo the number of intrinsic camera parameters required are half the number of parameters required for traditional stereo. Identical intrinsic camera parameters and the planar motion constraint place restrictions on the epipolar geometry [Gluckman & Nayar].

### 3.1.4 Apparatus

There are different criteria that need to be considered when designing a catadioptric sensor. These criteria include the shape of the mirror, the resolution of the sensor and the focus settings of the camera. In this section aspects of the apparatus that need to be considered are taken into account. The equipment required for a catadioptric stereo system include a single image capture device and one or more reflective surfaces.

**(a) The Image Capture Device**

The image capture device used, must have a sufficiently large resolution and field of view. The field of view needs to be large enough to see the mirror and the object simultaneously. The maximum achievable accuracy of the system depends on the resolution of the camera. For some catadioptric sensors the field of view is increased by using non-planar mirrors. For a catadioptric system using planar mirrors the system's field of view can not be increased. This is because using a planar mirror with a camera is simply the same as moving the camera to a different position; i.e. using a virtual camera in a different position with the same resolution as the real camera [Baker & Nayar].

The camera used must suit the application and application environment. For example when using this method to implement a real-time interaction device the camera needs to have a sufficiently fast frame rate, no less than 25 frames per second. Or if the physical environment has low light conditions then it may be necessary to use an infrared camera. When using an infrared camera it is necessary to use infrared illuminators. If infrared is

not used and a standard camera is used then a stable light source is required that is adequately bright.

**(b) The Reflective Surfaces**

A reflective surface is needed which will reflect a clear and distinguishable image of an object or shape. In this thesis whenever the term reflective surface or mirror is used it refers to such a reflective surface.

In the discussion of the geometry of a catadioptric sensor in section 3.1.1 it was pointed out that the mirror (reflective body) acts as a virtual camera. An alternative way of expressing this is that the reflective body acts as a large picture of a scene which is viewed by the camera.

The size and the shape of the mirror influence the catadioptric sensor. A reflective body that is sufficiently large must be used, with the correct shape. In the context of this paper only planar mirrors are considered. There are several reasons for this choice:

(i) planar mirrors do not experience the image blur that curved mirrors do,

(ii) the focus settings of the camera are not a concern since there is no defocus blur,

(iii) the resolution using planar mirrors is the same as that of the camera,

(iv) a single camera can capture multiple views of a scene using planar mirrors. This is advantageous for applications where multiple viewpoints are necessary, and

(v) calculations for planar mirrors are simple.

While catadioptric stereo is not limited to the use of planar mirrors, the use of planar mirrors remains a good choice.

A catadioptric sensor is not limited to the use of only a single mirror. Using the right number of mirrors has many advantages. For instance, one advantage of using an odd number of mirrors is that rectified stereo images may be obtained without having to perform rectification on an image. Rectification is described in section 3.1.5-(b)

One additional reason for not using curved mirrors is that the finite size of the lens aperture with the curvature of a mirror contributes to image blur [Baker & Nayar]. This is a non-issue in this thesis, since planar mirrors have no curvature and the image blur of a perspective camera is negligible.

**(c) The Computer, the Tracked Object and Lighting**

To use a computer vision approach for tracking in interactive applications requires a fast computer. The computer must also have sufficient memory and a high bandwidth to handle multiple frames per second.

The type of object to be tracked depends on the lighting of the physical environment of the setup. A light or self-luminous object is needed in environments with little or no light, while a coloured object is easily tracked in an environment with sufficient lighting. The ability of the computer vision algorithm to identify an object will determine to a large extent the characteristics of the object to be tracked. For instance it is feasible for a computer vision algorithm to track a coloured object in a well lighted environment and a self-luminous object in low light settings. Thus if colour is used to track the object, then the object needs to be coloured appropriately for the computer vision algorithm to identify it. However the object to be tracked need not be coloured or even self-luminous if a good tracking system and appropriate computer vision algorithm are used. The object may be tracked and identified by its features such as contour, area, eccentricity, perimeter and image moments.

Lighting is a crucial issue. It is light which emanates from some source and illuminates a scene so that it can be filmed by a camera. Without light everything would be dark and

the camera would see nothing. Therefore adequate light of the right colour is essential, especially when the computer vision algorithm tracks a coloured object. The light source must be stable. Flickering lights to a large extent affect the ability of the computer vision algorithm to track the correct object. Consider for example using a camera that captures 60 frames per second (fps) in a scene illuminated by a fluorescent light that flickers at 30 Hz. Even if the camera and light are in sync, every second frame the camera captures will be lighted differently.

### 3.1.5 Image Formation

Before moving on to consider the placement of the camera and mirror, image formation must be considered. This is because it is necessary to determine the type of image the video camera needs to capture in order for the Reflections method to work.

The term catadioptric stereo implies that the images acquired by the camera mirror system resemble stereo images. The Reflections approach, for 3D calculation, requires that the camera obtain images in which two different distinct views of an object are seen. By correctly placing the camera and mirror (or mirrors) the camera effectively captures two such different 2D views of a scene. The view of the single camera is divided in two. Half of the image is dedicated to viewing the mirror and half is given for viewing the object directly (or a second mirror). The image taken by the camera appears to consist of two images placed side by side. These images of the same scene appear to have been taken by two cameras from different positions and angles.

Figure 3.7 illustrates a catadioptric stereo image taken by the camera in the setup in figure 3.1.

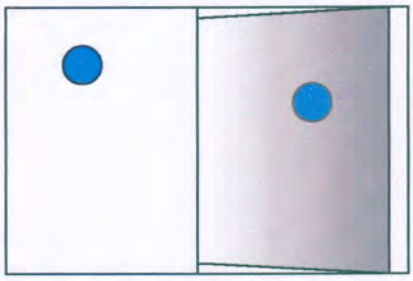


**Figure 3.7 Picture Taken by Camera in a Simple Reflective Setup**

One can see the object distinctly in two different positions in the picture, namely the direct view of the ball as seen by the camera and the reflection of the ball (an indirect view of the ball seen by the camera via the mirror).

After the image is captured, image analysis needs to be preformed on the image to identify, locate and match points in the two different views. That is key points of the object are matched with corresponding key points of its reflection.

In order to obtain pictures similar to the picture in figure 3.7, certain constraints need to be placed on the positioning and orientation of the camera and the reflective surface. These constraints are discussed next.

**(a) Epipolar Geometry**

Epipolar geometry is a means of geometrically describing the relationship between stereo images. Epipolar geometry is often used to match corresponding points in stereo images. Matching corresponding points in the stereo images is crucial for real time 3D reconstruction. The epipolar geometry aids the matching process by reducing the search space from 2D to 1D. This is because it is known that corresponding points lie on the same epipolar lines.

Epipolar lines pass through a point called the epipole. The plane of motion contains the centers of the virtual cameras. The planar motion constraint and the fact that the virtual cameras have identical intrinsic parameters leads to the following results:

- corresponding epipolar lines intersect at the image projection of the screw axis, **m**,
- the epipolar geometry is simply determined from the two epipoles and the line **m**, and
- the horizon line of the plane of motion contains the two epipoles [Nayar & Gluckman].

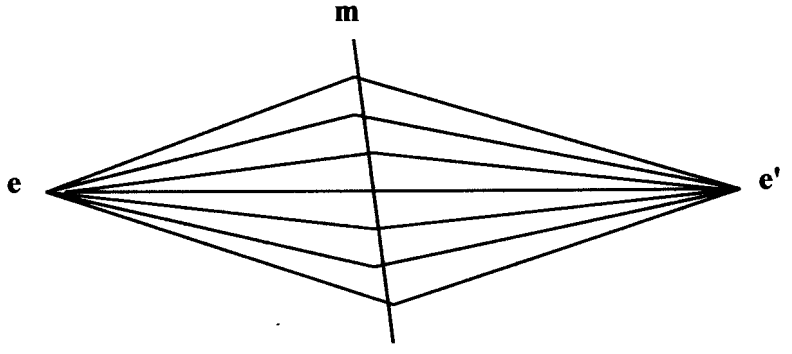



**Figure 3.8 Epipolar Lines Must Meet at the Image Projection of the Screw Axis, m**

Figure 3.8 illustrates how corresponding epipolar lines intersect at **m**, the image projection of the screw axis.

### (b) Rectified Catadioptric Stereo

Rectification is a process in which epipolar lines are aligned with the scan lines of a camera in stereo images. Rectification of stereo images requires additional processing to be done at run time. Rectified images are useful for matching point correspondences in stereo images. This is because in rectified catadioptric stereo images the epipolar lines are aligned with the horizontal scan lines of the image. If point correspondences do not lie on the horizontal scan lines then a complex and computationally expensive process is needed to rectify the epipolar lines. Another disadvantage of rectification is that image degradation occurs as a result of distortions and re-sampling.

Catadioptric systems are able to produce rectified stereo images without additional processing cost and without having to calibrate the internal parameters of multiple cameras. Rectified catadioptric stereo is therefore more advantageous than conventional stereo. The way in which this is achieved is simply by using the right number of mirrors and placing them correctly.

The three requirements for a stereo pair to be rectified are the following:

(i) there must be no relative rotation between the cameras,

(ii) translation must be parallel to the scanlines of the image plane, and

(iii) the internal camera parameters must be identical.

By examining these requirements the benefit of catadioptric stereo for rectification becomes clear. Because catadioptric stereo only uses a single camera, the internal camera parameters are identical (satisfying requirement number (iii)). Meeting the first two requirements is now only a matter of correctly placing the camera and mirrors.

Catadioptric stereo splits the field of view between different systems of mirrors. It is necessary to ensure that the two fields of view overlap (see section 3.2.6). In order to overlap the two half fields of view it is necessary to reflect one of the fields of view relative to the other. To do this requires that an odd number of mirrors be used. By using an odd number of mirrors rectified stereo images can be obtained.

The simplest catadioptric solution for obtaining rectified images is to use a single mirror. The mirror needs to be placed in such a way that its normal lies parallel to the camera's scanlines. Such a rectified catadioptric sensor is shown in figure 3.9. Limitations of the single mirror approach are that the angle of view is limited by the angle formed between the mirror and virtual camera and that the field of view is also limited by the size of the mirror. The single mirror approach is suitable when the object of interest is near to the camera. One other problem encountered with the single mirror approach is that the field

of view is uneven. A larger part of the field of view is given to the direct view of the camera.

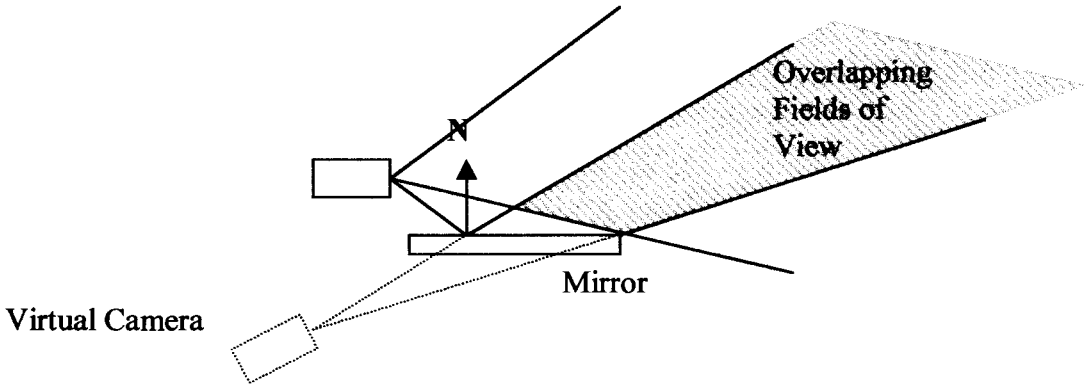



**Figure 3.9 Single Mirror Rectified Catadioptric Stereo Sensor**

Limitations encountered in a single mirror rectified stereo system are overcome in a three mirror system. The immediate advantage of using three mirrors is that the field of view may be divided equally between the different systems of mirrors. Another advantage of using three mirrors is that a large scene is viewable by using small mirrors. Although there are additional constraints placed on the positioning and orientation of the mirrors the three mirror approach seems to be the optimal solution for obtaining rectified catadioptric images. An additional constraint in such a system is that the two rays emanating from the pinhole of the camera must be parallel after being reflected by the mirrors. This constraint requires that the mirrors be angled correctly. It also ensures that there is no rotation between the two virtual cameras.

### 3.1.6 Camera-Mirror Setup

To set up the camera and mirrors to obtain images suitable for 3D calculation, the region of interaction must be taken into account. The region of interaction is that volume in which an object moves and for which the approach calculates the position of the object. This is the region in which images taken by the camera have at least two different views of the object. The camera and reflective surface need to be positioned in such a way that the camera sees both the object and the reflective surface simultaneously. The reflective

surface also needs to be angled in such a way that the camera sees the reflection of the object in the mirror. The two fields of view seen by the camera include the field of view in which an object is seen directly by the camera and the field of view visible to the camera via the mirror (the field of view of the virtual camera). The intersection of these two views form the region of interaction. Note that this region excludes the intersection of the two views directly between the camera and the mirror. Figure 3.10 illustrates the region of interaction.

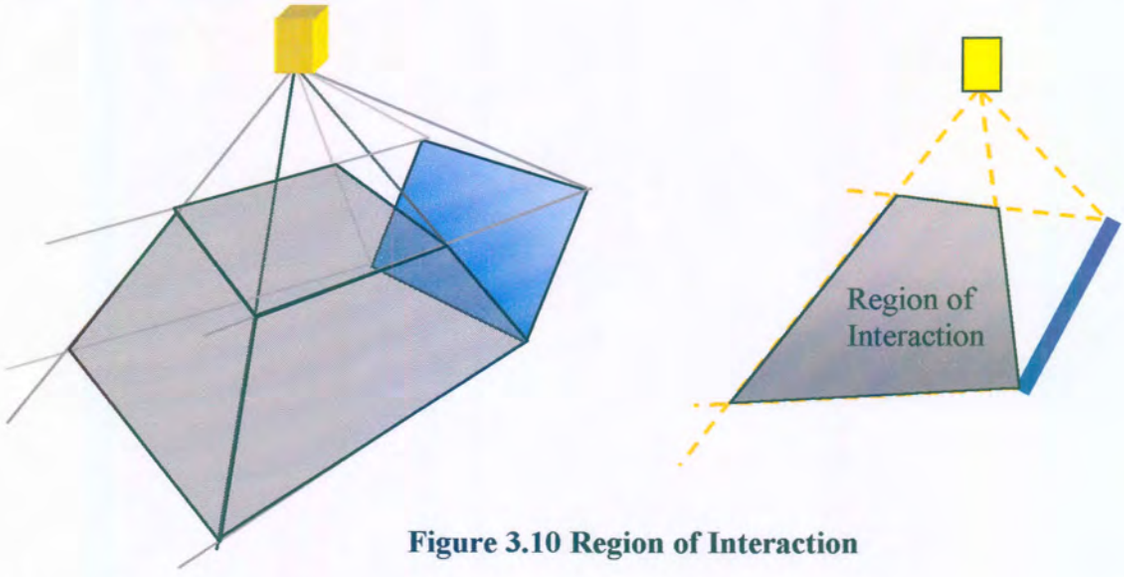



**Figure 3.10 Region of Interaction**

In figure 3.11 the idea of overlapping fields of view is illustrated. These requirements influence both how and where the camera and mirror are placed when installing the apparatus.

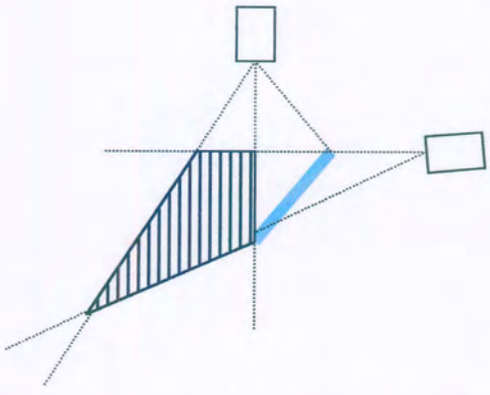


**Figure 3.11 Illustration of Overlapping Fields of View**

Another aspect that should be taken into account when placing the camera and mirror is occlusion. The camera and mirror should be placed so that occlusions are minimized. By wisely selecting the position of the camera and mirror the computer vision algorithm can be simplified or optimized for a specific application. Such optimization and simplifications are beneficial for real-time applications in which the computer vision algorithm needs to keep up with a high frame rate.

Once the equipment is in place measurements are taken and then the process of calculating the 3D information begins.

### 3.1.7 3D Calculation

#### (a) Physics

To construct a physical model of the camera and mirror setup for accurate 3D calculation it is necessary to have an understanding of the physical elements involved in such a system and the properties of these elements. These physical elements include light and mirrors (reflective surfaces). A physical model of a camera and how light interacts with it is also required. The first element to be dealt with is light. Without light, be it infrared or pure sunlight, the camera can see nothing.

#### (i) Light

Light is physically described as an electromagnetic wave. Light has both a wave and particle nature. The wave nature characterizes many properties of light. For instance the wavelength of light determines its colour and the propagation or motion of light is described by using wave fronts. The propagation of light may however also be described by using a ray model. Geometric optics makes use of the ray model of light. A ray represents the path of a light particle and better describes the particle nature of light. Geometric optics deals with and gives an understanding of the propagation and behavior of light at and through different surfaces such as lenses and mirrors. In the ray model

light travels in straight lines (in a homogeneous material). In physics a ray is defined as "*an imaginary line along the direction of travel of the wave*" [Young, 1992]. This definition keeps with the wave nature of light and yet expresses the characteristics of the direction in which particles of light travel. In graphics a ray is a "*semi-infinite line emanating from a point that travels to infinity in a particular direction*" [Angel, 2000]. In both cases a ray is a line with direction.

In geometric optics, light emanates from some source. Rays of light travel in all directions from a source to infinity unless they make contact with some object. At the point where light makes contact with the surface of an object a portion or all of the light may be reflected or refracted. Rays interact with different surfaces in different ways. For instance if a surface is a perfect mirror then all the light of a ray will be reflected.

The ray model of light is useful for simulating and studying the physical effects of light and its interaction with objects such as mirrors and lenses because it gives a close approximation of the movement of light in the physical world. Although it is not well suited for fast computation some simplifications can be made that make computation viable. One such simplification, when considering how rays interact with a camera, is that only those rays that finally pass through the lens of the camera and fall on the film plane need to be considered. This is because these are the only rays that contribute to image formation. The rays which pass through the lens are either direct rays of light emanating from a light source or rays reflected off some object or refracted through some object. The interaction between light, objects and the camera determines how much light is seen by the camera.

### (ii) Mirrors

A mirror is a smooth surface that reflects light. The reflection of light at a planar surface is termed specular reflection. Specular reflection occurs at very smooth surfaces when light is reflected at a definite angle. Reflected light that is scattered by a rough surface is

referred to as diffuse reflection. Figure 3.12 illustrates specular reflection. An example of an object with good specular reflection is a flat mirror.

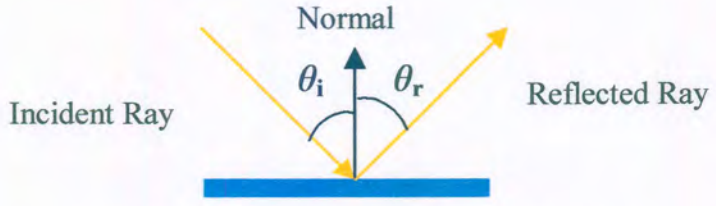



**Figure 3.12 Behavior of Light at a Flat Reflective Surface**

The incoming light ray is termed the incident ray and the outgoing ray is called the reflected ray. These rays are described in terms of the angles they form with the normal of the surface at the point of incidence, as illustrated in figure 3.12.

It is important to take note of the laws governing reflection at a smooth reflective surface. The law of reflection states the following:

(1) the incident, reflected and refracted rays and the normal to the surface all lie in the same plane.

(2) The angle of reflection $\theta_r$ is equal to the angle of incidence $\theta_i$ for all wavelengths, i.e. $\theta_r = \theta_I$

Together (1) and (2) above form the law of reflection, illustrated in figure 3.12.

Planar mirrors have the property that they always produce real (the same size) erect (upright) and reversed images. A useful property of such mirrors is that all images formed by such reflecting surfaces can serve as images for other reflecting surfaces.

### (iii) The Pinhole Camera Model

The pinhole camera provides a simple, yet effective geometric model of a camera. It gives much insight into certain fundamentals of image formation that are required. The pinhole camera consists of a box with a small pinhole on one side. The pinhole allows only a single ray of light to enter the box. This light entering at the pinhole falls on a film plane on the opposite side of the box as illustrated in figure 3.13.

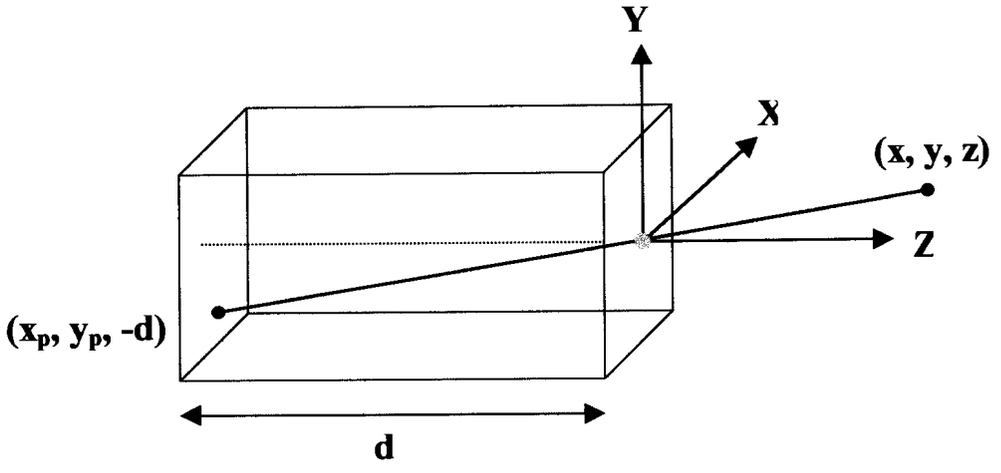



**Figure 3.13 The Pinhole Camera**

It is important to calculate the projection of the point (x,y,z) onto the film plane. The projection point at (xp, yp, -d) is a point on a film plane that is d units away from the pinhole of the camera along the negative Z-axis. One of the aspects of image formation in this model is that the image of a point seen on the film plane coincides with a point in 3D space. The colour on the film plane for this point is the same as the colour of the real 3D point.

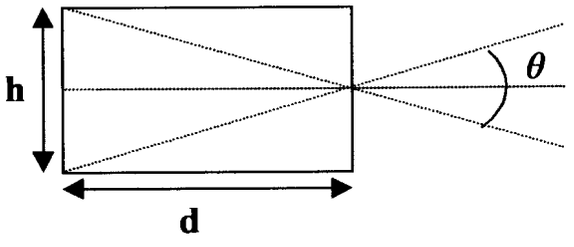



**Figure 3.14 The View Angle of a Pinhole Camera**

Figure 3.14 illustrates how the focal length, the size of the film plane and the angle of view are related. This relationship is expressed in equation 3.1.1.

$$\theta = 2 \tan^{-1} \frac{h}{2d} \qquad (3.1.1)$$

That which a camera sees is determined by the field of view of the camera. The field of view for the pinhole camera is the angle formed by the largest object that can be imaged on the film plane.

The height of the film plane, h, is determined by the size of the largest object that can be viewed by the camera. To extend the above equation for use in a three-dimensional problem, the diagonal length of the film plane is used for h.

The limitations of the pinhole camera are that it admits only a single ray of light (therefore there is only limited light within the camera) and that it has a limited angle of view which can not be adjusted. Most conventional more sophisticated cameras overcome this problem by using a lens in place of the pinhole. The lens is larger than a pinhole and allows more light to enter the camera. By selecting an appropriate lens almost any field of view up to 180 degrees can be achieved. This is equivalent to choosing an appropriate focal length, d in the pinhole camera model. An ideal pinhole camera model has an infinite depth of field. That is every point in the field of view of the camera is in focus. This is not the case with a lens (a lens does not have infinite depth of field). To overcome this problem, a lens can be moved forward or backward to focus on objects at different distances.

The reason a lens does not have an infinite depth of field is because a lens consists of two refracting surfaces. Refracting surfaces are normally transparent and allow light to pass through them. As light passes through a lens it is bent (refracted). Because of this every lens has a specific focal length. The focal length of a lens is determined by the curvature of its surfaces and its index of refraction.

A camera has its own co-ordinate system of axes associated with it. Pictures taken by conventional cameras produce real, inverted and reduced images. The resolution of a camera lens is determined by the maximum number of lines per millimeter in an image that can be distinguished as separate lines [Young].

From the above principles it is possible to construct a mathematical and physically correct model of the setup of a camera and mirrors.

**(b) Calibration**

For conventional stereo cameras calibration includes determining certain intrinsic and extrinsic camera parameters for each of the two cameras used in the stereo rig. This is done by finding the relationship of the perspective projection between 3D points in a scene and different camera images. Calibration is crucial for the vision process because it allows the matching problem to be simplified. Calibration is carried out before accurate 3D calculation is performed.

Conventional stereo methods calibrate the camera by using a predefined grid of known points of which combinations of corresponding points are found and used to compute the perspective projection matrices for the two cameras used in the stereo rig. A weak calibration technique also exists in which only the epipolar geometry of the stereo rig is known but not the intrinsic parameters of the cameras. For this approach only a small number of pixel correspondences between the two stereo images are required.

Calibration for the Reflections approach to 3D calculation requires finding certain external calibration and intrinsic camera parameters. The external parameters required include the position and orientation of the mirror relative to the camera and the orientation of the camera. These can be measured informally by using a protractor, ruler and bevel. The internal camera parameters required include the focal length of the camera, the image skew, the image center and aspect ratio. For most conventional video cameras image skew is assumed to be zero, the image center is taken to be at the center of

the image and the aspect ratio is simply determined by the dimensions of the image. Often the aspect ratio is given in terms of mm. The focal length of the camera is determined by the size of the film (film plane at the back of the pinhole camera) and the angle of view.

**(i) Calculating the Focal Length and Angle of View**

For some cameras the focal length is not known, or perhaps only the aspect ratio of the camera in pixels is known. Before 3D calculation begins it is necessary to determine how a point located in an image, with its position given in pixels, corresponds to a point on the film plane in the mathematical model, with its position given in terms of millimeters. To do this it is necessary to know the focal length and angle of view of the camera. The method below is a procedure for calculating the focal length of a pinhole camera that is equivalent to the video camera being used. This procedure only needs to be implemented once for a specific camera if the focal length and angle of view are unknown. This step precedes the installation of the mirrors and camera.

The focal length **d** of the pinhole camera which has a corresponding 1:1 pixel to millimeter ratio with the images captured by a camera for which only the aspect ratio in pixels is know, is found in the procedure set out below:

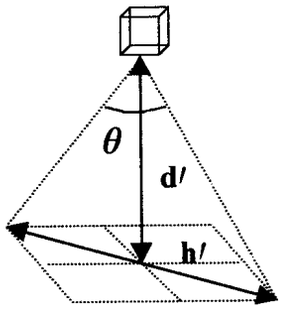



**Figure 3.15 Measurements to be Taken to Find the Angle of View**

(1) Figure 3.15 shows the measurements that need to be taken to determine the focal length of the pinhole camera. The measurements that need to be taken are **d/** and **h/**

where $\mathbf{h}'$ is the largest diagonal length that can be imaged by the camera at a distance $\mathbf{d}'$ from the camera. $\mathbf{h}'$ and $\mathbf{d}'$ are measured in millimeters.

(2) Next the field of view, $\theta$ for this camera is calculated by using the following equation:

$$\theta = 2 \tan^{-1} \frac{\mathbf{h}'}{2\mathbf{d}'} \qquad (3.1.2)$$

(3) The maximum diagonal distance, h, in pixels of the image plane is calculated using the given aspect ratio ($\mathbf{m}$x$\mathbf{n}$ pixels). This is illustrated in figure 3.16.

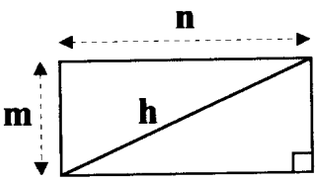



**Figure 3.16 Dimensions of Image**

$$\mathbf{h} = \sqrt{\mathbf{m}^2 + \mathbf{n}^2} \qquad (3.1.3)$$

If the diagonal length of the film is $\mathbf{h}$ pixels long then the corresponding film plane in the mathematical model having a 1:1 pixel to millimeter ratio with this film plane will have a diagonal length of $\mathbf{h}$ millimeters.

(4) The distance from the film plane to the camera in the mathematical model is the focal length of the pinhole camera. This focal length $\mathbf{d}$ is calculated in millimeters as follows:

$$\mathbf{d} = \frac{\mathbf{h}}{2 \tan(\theta/2)} \qquad (3.1.4)$$

where $\theta$ is the field of view and $\mathbf{h}$ is the diagonal length in millimeters of the film plane.

### (ii) Assumptions and Setup

Certain assumptions and constraints are made. These assumptions and the restrictions placed on the setup are done to simplify the 3D calculation procedure.

Assumptions:

(i) A pinhole camera model is assumed.

(ii) The mirror is a perfect flat reflective surface.

(iii) Aberrations and distortions are negligible.

(iv) A perspective camera is assumed.

Setup:

(i) A Cartesian co-ordinate system is assumed for the mathematical model.

(ii) The origin of this co-ordinate system coincides with the center of projection (COP) of the camera (the pinhole of the camera).

(iii) The camera is oriented so that it faces the negative Y direction. Its film plane lies parallel to the XZ plane. The axes of the camera are aligned with the axes of the assumed co-ordinate system; that is the horizontal axis of the film plane coincides with the Z axis and the vertical axis of the film plane lies on the X axis.

(iv) The mirror is positioned **H** units below the origin and $\mathbf{D_1}$ units in the direction of the positive X-axis, i.e. it has a point that lies at $(\mathbf{D_1}, \mathbf{-H}, \mathbf{0})$.

(v) The mirror is oriented so that it's edges run parallel to the Z and Y-axes.

(vi) The mirror forms an angle $\theta$ with the projection plane. This is the angle measured from the projection plane to the mirror, parallel to the X-axis.

(vii) The projection plane lies **H** units below the origin. It lies parallel to the XZ plane.

**(c) The Mathematical Model**

From the above assumptions and set-up information the mathematical model in figure 3.17 is constructed.

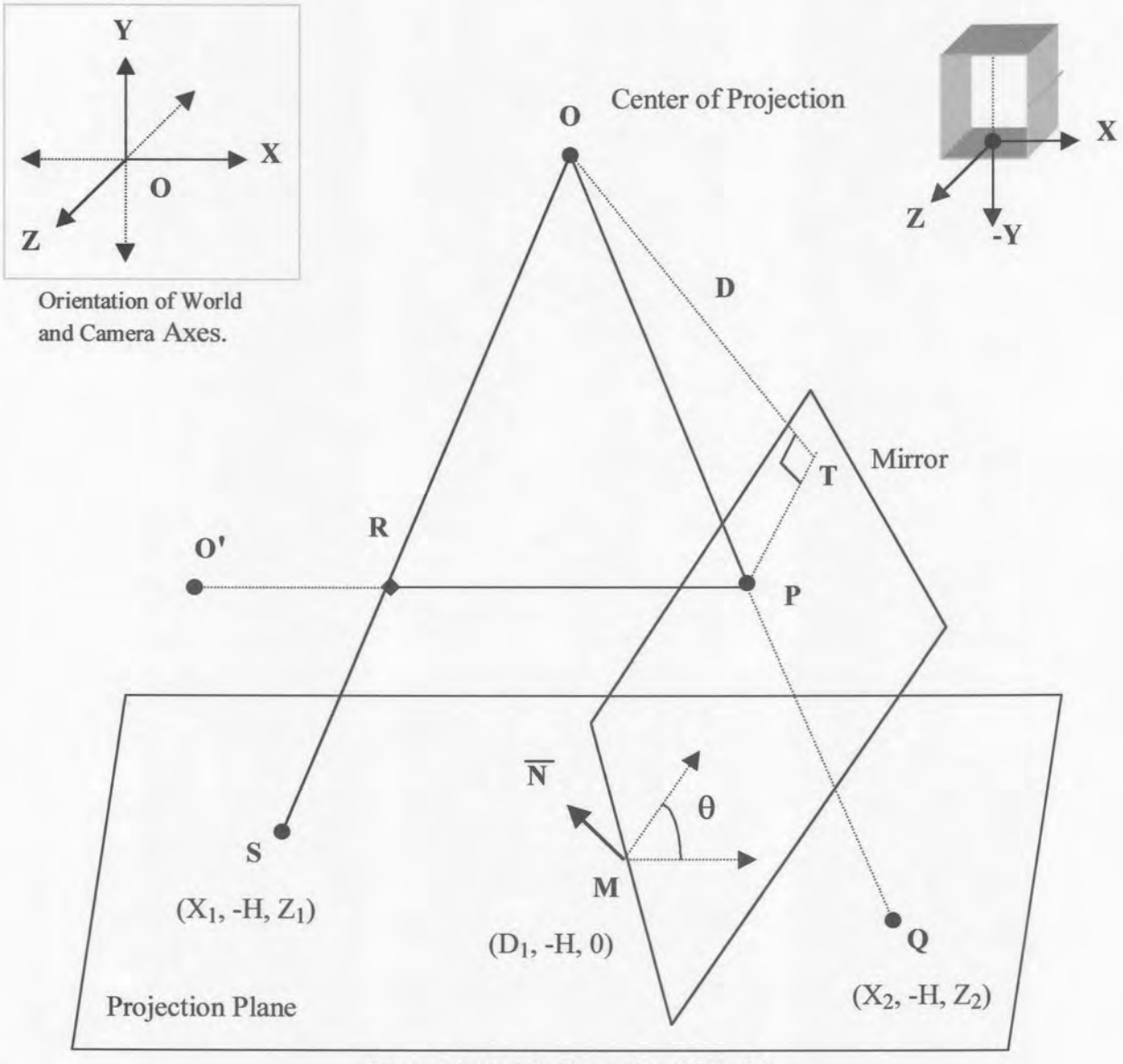



**Figure 3.17 Mathematical Model**

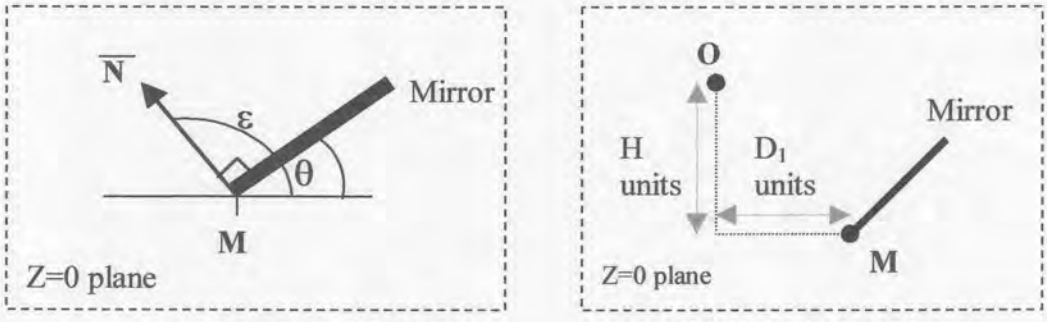
It takes into account the physics of the elements involved, such as light and reflection. This model is used for both the trigonometric approach to calculation and the algebraic method. The camera COP is placed at the origin. It is known that light, reflected off the

object of interest passes through this point (the pinhole of the camera). In this model a projection plane is used in place of a film plane. The mirror is represented mathematically by a plane equation and the rays of light are represented by line equations.

**(d) The Pre-Processing Step**

From the calibration measurements certain necessary elements of the mathematical model are calculated. Some of these elements need only be calculated once for a set up. All information that needs to be calculated only once is done in the pre-processing phase, which is executed before images are captured by the video camera. Doing this saves precious computational time when performing the 3D calculations because the quantities pre-calculated do not need to be recalculated for each frame.

In the pre-processing step the equation of the plane (mirror) is determined. This equation is calculated from a known point **M** on the plane and a normal vector of the plane **N**. The normal **N** is determined from the angle that the mirror subtends with the floor. This angle is measured in the calibration phase. From the assumptions and initial set up it is known that the mirror lies parallel to the X-axis and perpendicular to the Z-axis. If $\theta$ is the angle measured from the floor to the reflecting surface, parallel to the X-axis, then the angle of the normal is $\varepsilon = \theta + \pi/2$. Figure 3.18 (a) illustrates these measurements and calculations.




**(a) Calculating N** **(b) Calculating M**

**Figure 3.18 Illustration of Measurements**

From the angular information of the mirror the normal vector is calculated:

$$\mathbf{N} = \langle \cos \varepsilon, \sin \varepsilon, 0 \rangle$$

The point $\mathbf{M} = (D_1, -H, 0)$ lies on the mirror because the mirror is placed $D_1$ units away from the camera in the direction of the X-axis and $H$ units in front of the camera as illustrated in figure 3.18 (b). From this known point $\mathbf{M}$ and the normal $\mathbf{N}$ the equation of the plane representing the mirror is found to be:

$$\cos \varepsilon \, ( x - D_1 ) + \sin \varepsilon \, ( y + H ) = 0 \tag{3.1.5}$$

If the geometric and trigonometric approach is used in the calculation of the 3D information, then the distance $D$ from the origin $O$ in the mathematical model to the plane (mirror) is calculated. This is done using the following equation:

$$D = \frac{| \mathbf{N} \cdot \overrightarrow{P_0P_1} |}{\| \mathbf{N} \|} \tag{3.1.6}$$

where $P_0$ is a point on the plane and $P_1$ is the origin. This reduces to:

$$D = \frac{| - D_1 \cos \varepsilon + H \sin \varepsilon |}{\sqrt{(\sin \varepsilon)^2 + (\cos \varepsilon)^2}} \tag{3.1.7}$$

$$\overrightarrow{P_0P_1} = \begin{bmatrix} 0 - D_1 \\ 0 - (-H) \\ 0 - 0 \end{bmatrix}^T \tag{3.1.8}$$

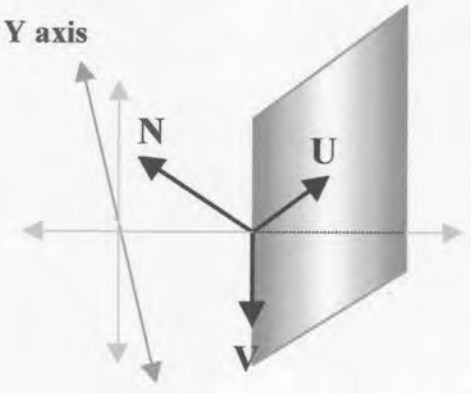
$$| \mathbf{N} \cdot \overrightarrow{P_0P_1} | = | - D_1 \cos \varepsilon + H \sin \varepsilon + 0 | \tag{3.1.9}$$

$\overrightarrow{P_0P_1}$ is the vector from the origin to point $\mathbf{M}$ on the plane. This distance need not be calculated if the algebraic approach is used.

When using the algebraic approach to calculate the 3D information, change of basis matrices and a reflection matrix are calculated in the pre-processing phase. These matrices are used throughout the algebraic calculations, but only need to be calculated once for a specific set up. The use of these matrices will become apparent when the algebraic approach is explained. Next the construction of these matrices is described.

To construct the change of co-ordinate matrices between the original co-ordinate system and the co-ordinate system of the mirror (determined by the orientation of the mirror), the basis vectors for these matrices are needed. The set of basis vectors for the first system *B* is simply the set of standard co-ordinate vectors, for the X, Y and Z-axes. The co-ordinate bases *C* for the mirror's co-ordinate system are the normal, the vector that runs along the mirror in the X direction and the Z-axis (since the mirror is parallel to the Z-axis). The bases of the mirror are illustrated in figure 3.19.

Once the bases have been determined the different change of bases matrices are constructed. These matrices transform the relative co-ordinates of a point in each of the




$\mathbf{N} = \langle \cos\varepsilon, \sin\varepsilon, 0 \rangle$
$\mathbf{U} = \langle \cos\theta, \sin\theta, 0 \rangle$
$\mathbf{V} = \langle 0, 0, 1 \rangle$

**X axis** $= \langle 1, 0, 0 \rangle$
**Y axis** $= \langle 0, 1, 0 \rangle$
**Z axis** $= \langle 0, 0, 1 \rangle$

**Figure 3.19 The Different Co-ordinate Bases**

systems to the other. That is they will allow a point to be swapped between the two co-ordinate systems. These change of bases matrices are constructed in the following way:

$$\underset{B \leftarrow C}{\mathbf{P}} = \begin{bmatrix} \mathbf{N} & \mathbf{U} & \mathbf{V} \end{bmatrix} = \begin{bmatrix} \cos\varepsilon & \sin\varepsilon & 0 \\ \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.1.10)$$

$$\underset{C \leftarrow B}{\mathbf{P}} = \left[\begin{array}{ccc|c} \mathbf{N} & \mathbf{U} & \mathbf{V} & \mathbf{I} \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & 0 & \\ 0 & 1 & 0 & \mathbf{P}^{-1} \\ 0 & 0 & 1 & \underset{B \leftarrow C}{} \end{array}\right] \qquad (3.1.11)$$

It is also necessary to calculate a reflection matrix. The reflection matrix must reflect a point about the X and Z-axes of the standard co-ordinate system. This matrix is simply:

$$\mathbf{R} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \qquad (3.1.12)$$

After the pre-processing step a video sequence is started, in which frames are captured by the video camera.

**(e) From the Film Plane to the Real World**

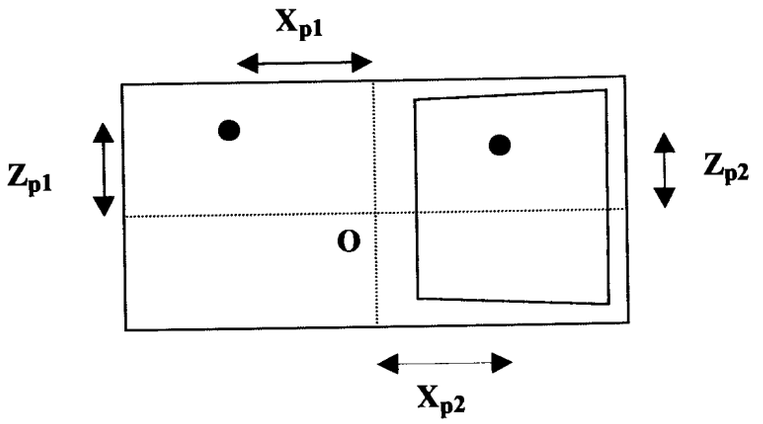



**Figure 3.20 Location of Points in an Image.**

The image analysis phase is applied to each of the frames captured. In this phase corresponding points are located and matched in each image. Figure 3.20 shows the two points located in an image. From the location of these points in an image, corresponding points on the film plane of the pinhole camera model are found. The positions of the points located in the image are given in terms of pixels relative to **O**, the center of the image.

For a film plane with a 1:1 pixel to millimeter ratio with the image, the corresponding points are given:

$(X_{p1}, Z_{p1})$ in pixels, corresponds to $(X_{p1}, -d, Z_{p1})$, in millimeters on the film plane and

$(X_{p2}, Z_{p2})$ in pixels, corresponds to $(X_{p2}, -d, Z_{p2})$, in millimeters on the film plane.

The image of a projection plane, **d** millimeters in front of the camera is equivalent to using an inverted image of the film plane. For this projection plane the pairs of corresponding points become:

$(X_{p1}, Z_{p1})$ in pixels on the film plane, corresponds to $(-X_{p1}, d, -Z_{p1})$ in millimeters on the projection plane and

$(X_{p2}, Z_{p2})$ in pixels on the film plane, corresponds to $(-X_{p2}, d, -Z_{p2})$ in millimeters on the projection plane.

These points on the projection plane are then projected onto a plane lying **H** units below the camera. This gives points **S** and **Q** in the mathematical model. The calculation of these projections is done in the following way using similar triangles as in figure 3.21.

$$X_1 = \frac{H}{d}(-X_{p1})$$

Similarly,

$$Z_1 = \frac{H}{d}(-Z_{p1})$$

and so:

$$Q = (X_2, -H, Z_2)$$

$$S = (X_1, -H, Z_1)$$

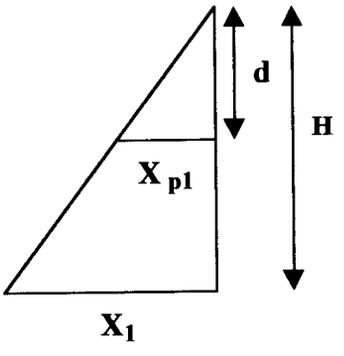



**Figure 3.21 Similar Triangles**

For each image captured by the camera the above steps are the first that must be performed when calculating the 3D information.

### (f) Initial Calculations

Certain initial calculations are performed for both the trigonometric and algebraic approach. These initial calculations determine the line equations of the light rays passing from where the object is seen on the projection plane to the pinhole and from the reflected image of the object to the pinhole. The intersection of the line from the reflection seen on the projection plane to the pinhole with the mirror is also calculated.

From the points found on the projection plane the equations of the lines from **O** to **Q** and from **O** to **S** are calculated:

$\mathbf{OQ = 0 + t_1 V_1}$

$\mathbf{OS = 0 + t_2 V_2}$

where $\mathbf{V_1 = Q - O = < X_2, -H, Z_2 >}$

and $\mathbf{V_2 = S - O = < X_1, -H, Z_1 >}$

Next the point where the reflection is seen in the mirror is found by substituting the line equation **OQ** into the plane equation. This substitution yields the equation:

$$\cos \varepsilon \, ( rX * t_1 - D_1 ) + \sin \varepsilon \, ( -H * t_1 + H ) = 0 \qquad (3.1.13)$$

Solving for $t_1$ gives:

$$t_1 = \frac{\cos \varepsilon * D_1 - \sin \varepsilon * H}{\cos \varepsilon * (rX) + \sin \varepsilon * ( -H)} \qquad (3.1.14)$$

Substituting the value of $t_1$ back into the line equation **OQ** gives the point **P**, the point of intersection of the line **OQ** with the plane:

$\mathbf{P = 0 + t_1 * <X_2, -H, Z_2>}$

$\mathbf{= (X_2 * t_1, -H * t_1, Z_2 * t_1)}$

The length from the origin to point **P** is:

$$| OP | = \sqrt{(X_2 * t_1)^2 + (-H * t_1)^2 + (Z_2 * t_1)^2} \qquad (3.1.15)$$

### (g) Geometric & Trigonometric Calculation

After the initial calculations it is possible to translate the problem from 3D into a simple 2D trigonometry and geometry type problem. This simplification is possible because all the points **O**, **T**, **P**, **R** and **S** lie in the same plane. All the lines between these points also lie in this plane. This is because the lines **OP** and **PR** lie in the same plane. This follows directly from the first part of the law of reflection which states that the incident and reflected rays as well as the normal about which the incident ray is reflected all lie in the same plane. **O** and **R** lie in the same plane and therefore so do all the points along the line **OR**. This includes point **S**. The point **T** also lies in this plane because it is the projection of the point **O** onto the mirror. This means that the line **OT** has the same orientation as the plane. It is also connected to the plane at point **O**. Therefore the line **OT** is part of the plane. Figure 3.22 on the following page illustrates the relevant lines from the mathematical model with angles and lengths that lie in the plane which are used in this approach.

The problem is simplified in that it is now simply a matter of finding the length of the line segment **OR**. By finding this length it is possible to use the parametric equations of the line **OS** to determine the 3D position of point **R**. The problem and given information are stated and solved as follows:

Given: **|D|, |OP|, OTP = 90, UPR =** $\alpha$ (because angle of reflection = angle of incidence)

$$\mathbf{S = (X_1, -H, Z_1),\ P = (X_2 * t_1, -H * t_1, Z_2 * t_1),\ O = (0, 0, 0)}$$

Calculate length: $t_2$

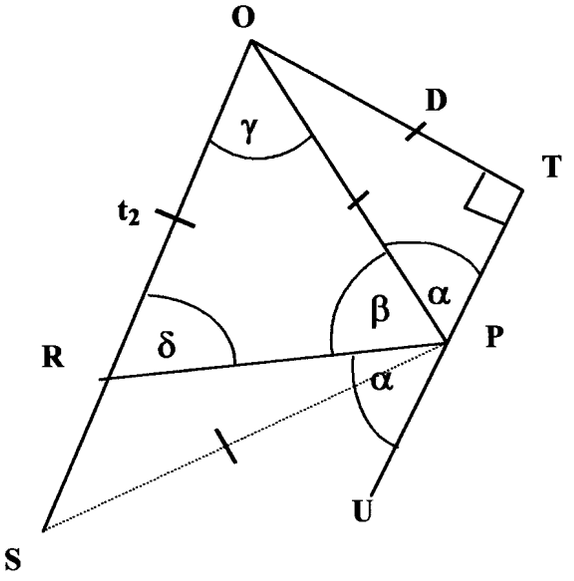



**Figure 3.22 Trig/Geometric View of the Mathematical Model**

| | |
|---|---|
| $\alpha = \sin^{-1}( D / \lvert OP \rvert )$ | $\sin \alpha = D / \lvert OP \rvert$ |
| $\beta = 180 - 2 * \alpha$ | *L's on Straight Line = 180* |
| $\lvert PS \rvert^2 = (X_1 -- (X_2 * t_1))^2 + (-H - (-H * t_1))^2 + (Z_1 - (Z_2 * t_1))^2$ | *distance of PS* |
| $\lvert OS \rvert^2 = (X_1)^2 + (-H)^2 + (Z_1)^2$ | *distance of PS* |
| $\gamma = \dfrac{\cos^{-1}( (PS^2 - OS^2 - OP^2)}{(-2 * OS * OP )}$ | *Triangle Rule* |
| $\delta = 180 - \gamma - \beta$ | *L's of $\Delta$ = 180* |
| $OR = \dfrac{OP \sin \beta}{\sin \delta}$ | $\dfrac{\sin \delta}{OP} = \dfrac{\sin \beta}{OR}$ |
| $t_2 = \dfrac{OR}{OS}$ | *Ratio of Line* |

Since $OS = 0 + t_2 < X_1, -H, Z_1 >$

The final 3D co-ordinate $R = (X, Y, Z) = (t_2 * X_1, t_2 * (-H), t_2 * Z_1)$

**(h) Algebraic Alternative**

Another way to calculate the 3D co-ordinates of an object using Reflections is to use linear algebra. After the initial calculations the following procedure is followed to calculate the 3D information (The first two steps are the same as in the initial calculation phase):

(i) Determine the equations of the lines **OQ**, **OS** and the equation of the plane.

(ii) Determine the point **P** where the line **OQ** intersects the plane.

(iii) Find **O'**, the reflection of point **O** about the normal of the plane at point **P**. This is done using the change of co-ordinate matrices and the reflection matrix.

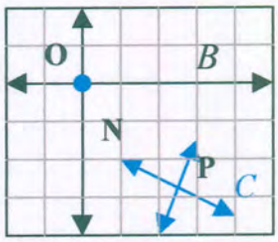
(iv) Determine the equation of the line **O'P**.

(v) Find the intersection of line **O'P** and line **OS**. This point of intersection gives the 3D co-ordinates of the desired point **R**.

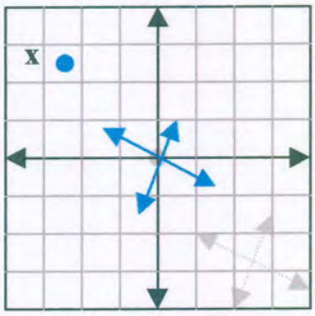
Homogeneous co-ordinates and standard 4x4 matrices commonly used in computer graphics may be used for these calculations.

Next the procedure to reflect the point **O** about the normal **N** located at point **P** is given (step (iii) above). Figure 3.23 illustrates the change of co-ordinates, the reflection and translations performed on a point in 2D that is reflected about a normal **N** at a point **P** on a mirror. This illustration is given in 2D for simplicity. The black axes refer to co-ordinate system *B* while the blue axes represent co-ordinate system *C*. The procedure is as follows:

First translate the system of axes at the point **P** and the point **O** so that the system at point **P** lies at the origin (it is necessary to translate the point **O**, so that it lies in its relative position to the system of axes at point **P**).



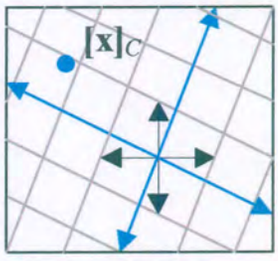

$$\mathbf{x} = \mathbf{O} - \mathbf{P}$$




Point **x** is given in terms of co-ordinate system *B*. **x** must be represented in terms of co-ordinate system *C*.

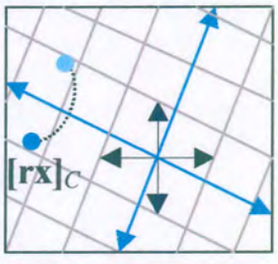
$$[\mathbf{x}]_C = \underset{B \leftarrow C}{\mathbf{P}} [\mathbf{x}]_B$$

This is done by multiplying **x** by the appropriate change of basis matrix.



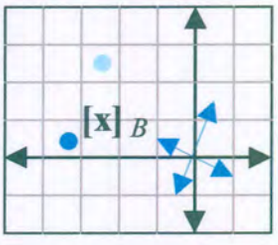

The point, $[\mathbf{x}]_C$ is now reflected about the X and Z-axes, using the reflection matrix **R**.

$$[\mathbf{rx}]_C = \mathbf{R}\,[\mathbf{x}]_C$$




The reflected point is given in terms of co-ordinate system *C*. Using the change of co-ordinate matrix from *B* to *C* this point is changed back to the original co-ordinate system *B*.

$$[\mathbf{x}]_B = \underset{C \leftarrow B}{\mathbf{P}} [\mathbf{rx}]_C$$




Finally, it is necessary to translate the system of axes back to its original position, and therefore the point must also be translated back.

$$\mathbf{O'} = [\mathbf{x}]_B + \mathbf{P}$$

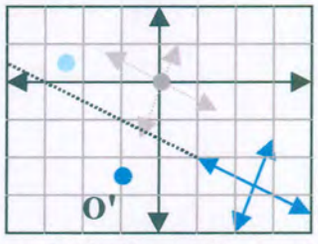



**Figure 3.23 Illustration of Reflecting the Point about the Mirror**

Once **O'** is found the line **O'P** is determined. This line is simply the reflection of line **OP** about **N** at **P**.

$$\mathbf{O'P} = \mathbf{P} + t_3 * \mathbf{V_3}$$

Where $\mathbf{V_3} = < \mathbf{P} - \mathbf{O'} >$.

The algebraic approach may be extended to work for multiple mirrors.

### 3.1.8 5D/6D Calculation

The above steps are used to calculate the 3D position of a point in space. To calculate the complete orientation of an object (6D) at least three such points are needed. To calculate 5D only two 3D points on an object are needed. These two or three points which are used to recover the orientation of an object should be points which convey the shape of the object as they are used to determine a set of axes vectors for the object.

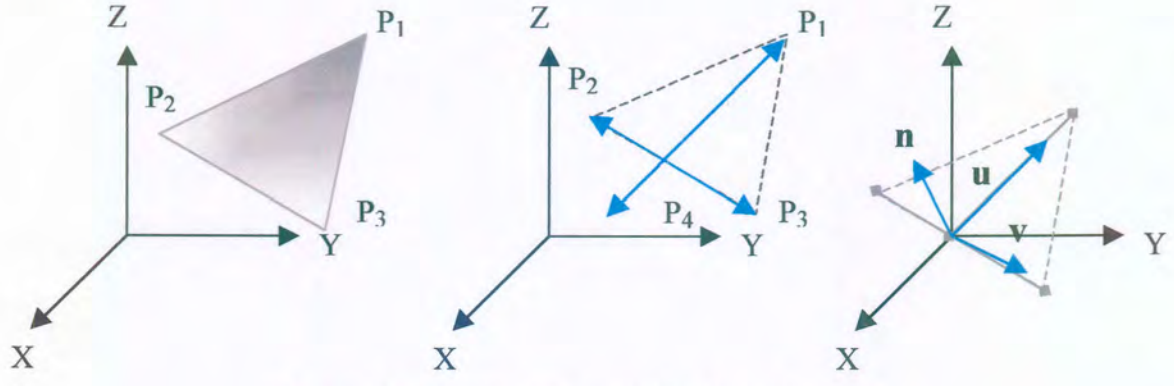



**Figure 3.24 Calculation of 5D/6D Information**

Figure 3.24 shows three figures, the first of which shows three points (in 3D) on a triangle that are used to express the structure and axes of orientation of the triangle. The second and third figure show the axes of the triangle that have been constructed from the original three points. These axes are found from the original points in the following way:

$\mathbf{u} = \mathbf{P_4P_1} = <\mathbf{P_1} - \mathbf{P_4}>$

$\mathbf{v} = \mathbf{P_2P_3} = <\mathbf{P_3} - \mathbf{P_2}>$

$\mathbf{n} = \mathbf{u} \times \mathbf{v}$

where $\mathbf{P_4}$ is a point lying on the line from $\mathbf{P_2}$ to $\mathbf{P_3}$ such that $\mathbf{P_4P_1}$ is perpendicular to the line $\mathbf{P_2P_3}$. It is important to note that **u**, **v** and **n** are orthogonal vectors.

These vectors lie at the origin of a Cartesian co-ordinate system and not at the original position of the object. This is desirable because it is not necessary to translate the system of axes to the origin. To determine the angles of orientation (rotation) of the object amounts to finding the angles of rotation that will rotate the vectors (**n**, **u**, **v**) of the object so that they lie on the axes of the co-ordinate system. Figure 3.25 shows the final position of the axes vectors of the object after these rotations have been applied to them.

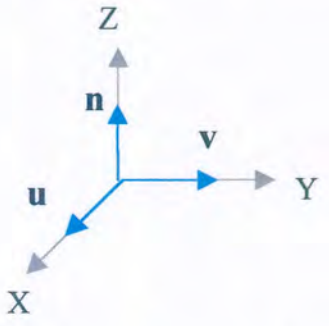



**Figure 3.25 The System of Axes after Applying All the Rotations**

The angles of rotation are found by the following procedure:

(1) Calculate $\theta$, then $\phi$ using the spherical polar co-ordinates of **u**. These are simply the angles **u** forms with the XZ plane and the X axis.

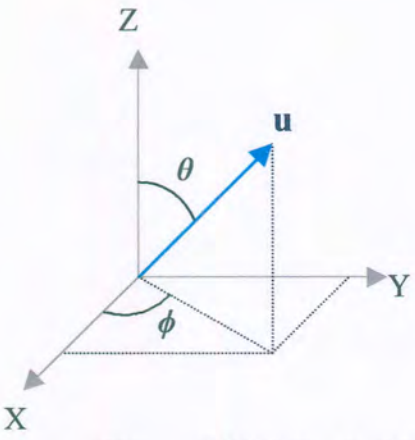
(2) rotate the vectors **u**, **v** and **n**, $-\phi$ about the Z-axis so that **u** lies in the XZ plane.

(3) rotate the vectors **u**, **v** and **n**, $-(90-\theta)$ about the Y-axis so that **u** lies on the X-axis.

(4) Calculate $\gamma$, the angle that the rotated vector **v** makes with the Y-axis.

These different rotation angles give the final degrees of freedom that express the orientation of the triangle.

In figure 3.26 the vector **u** is shown and the angles it makes with the Y and Z-axes. Steps one and two above require **u** to be reflected in such a way that it lies on the X-axis.




$$x = r \sin\theta \cos\phi$$
$$y = r \sin\theta \sin\phi$$
$$z = r \cos\theta$$

**Figure 3.26 Spherical Polar Co-ordinates**

Using spherical polar co-ordinates, a point (x, y, z) in Cartesian co-ordinates is expressed using two angles and a radius, as illustrated in figure 3.26. These angles are determined in the following way:

$$\theta = \cos^{-1}(z / r) \qquad (3.1.16)$$

$$\phi = \sin^{-1}(y / r \sin\theta) \qquad (3.1.17)$$

Because the x, y and z component of vector **u** are known, the first angle of rotation $\theta$ is calculated using equation 3.1.16. Thereafter $\phi$ is found using 3.1.17 Once $\phi$ has been found all the vectors **u**, **v** and **n** are rotated about the Z-axis by $-\phi$. This is equivalent to performing step (1). **u** now lies in the (x, z) plane. It is necessary to rotate **u** and the other vectors $-(90 - \theta)$ about the Y-axis to position **u** on the X-axis. Figure 3.27 shows the system of axes after these rotations have been performed.

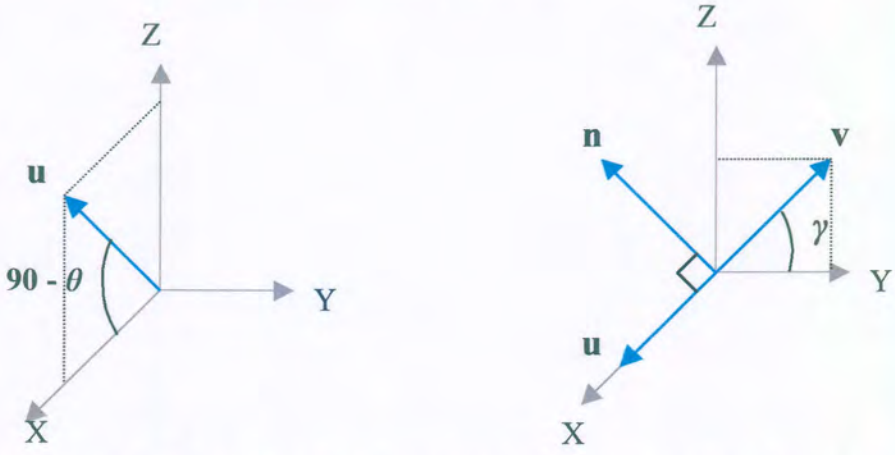



**Figure 3.27 System After Rotating by $-\phi$ and by $-(90 - \theta)$**

The final step is to position **v** on the Y-axis. **v** already lies in the (y, z) plane and **n** also (this is because **u**, **v** and **n** are all mutually perpendicular vectors and **u** lies on the X-axis). To find the angle of rotation from the Y-axis to **v** is simply calculated as follows:

If $\mathbf{v} = \langle 0, y, z \rangle$ then $\gamma = \tan^{-1}(z/y)$

The final rotation by $\gamma$ must result in **v** lying on the Y-axis and **n** lying on the Z-axis.

The final angles of rotation that express the orientation of the object are thus: $(\theta, \phi, \gamma)$.

The complete 6DOF that can be determined for an object include: three degrees for the objects position or relative translation from the origin of the camera and three for rotation expressing the orientation of an object relative to the axes at the origin:

$(x, y, z)$ - position

$(\theta, \phi, \gamma)$. - orientation

In the above explanation the orientation of a triangle is found using three known points on an object. The above approach assumes that a right-handed co-ordinate system is obtained for the object. For the above triangle there is no difference between the top view and bottom view of the triangle. For this reason it is not necessary to determine an up vector for the triangle. For some objects the above procedure to calculate the 6D information is sufficient. However for more complicated shapes it is necessary to distinguish between up and down. To do this it may be necessary to determine another point on the object, for the correct determination of the vectors **u**, **v** and **n**.

## 3.2 COMPUTER VISION

Computer vision is needed to analyze raw frames captured by the camera. It is the process in which meaningful and important information is extracted from the vast amount of data contained in an image. For the method of Reflections the computer vision algorithm has to identify and locate an object and the reflection of the object in the two different views of each image. Once it has located the object and it's reflection it needs to find and match certain feature points on the object and on the reflection of the object. Once a correlated point pair has been found the 3D information is calculated. Multiple degrees of freedom are calculated by finding and matching multiple correlated point pairs. The feature points found must reflect the geometric structure of the object. Meaningful point features that represent the structural features of an object include the center of gravity, edge points and corner points. The computer vision algorithm for Reflections detects an object, extracts and finds meaningful features of the object and then matches these features in the different views. Following are certain computer vision techniques which were found to be very effective and useful for the above tasks.

### 3.2.1 Colour & Inverse Chroma Keying

Colour provides a powerful and effective means of finding key regions of interest within an image, e.g. if a human face is to be found in a picture then only those skin-coloured regions of the picture need to be examined. Image segmentation provides a means of lessening the burden of searching through and processing an entire image. It does this by finding and exploring only those regions that are of interest. In colour images the use of colour presents an effective and simple way of segmenting an image. There are two different algorithms for doing this:

- vector keying, which segments those parts of an image with a colour that is approximately the same as a certain key vector (Colour may be represented as a vector), and
- chroma keying which segments regions of a specified colour.

Vector keying is the approach commonly used to find skin coloured regions in pictures. Vector keying algorithms typically learn the key colour by training the algorithm on a set of images. Chroma keying on the other hand is used to mask regions of a specified colour. It differs from vector keying in that the key colour need not be learned. For instance a colour such as blue is easily specified without having to learn a vector for this colour. A distance measure is used to classify shades of blue around a specific colour. Chroma keying is commonly used to mask out blue pixels belonging to a uniform blue background in the bluescreen technique of the film industry.

Inverse chroma keying is the use of chroma keying to either select specific regions of a certain colour or to mask out regions which are not of that colour. Because of its simplicity and effectiveness for segmenting an image and finding regions of interest, inverse chroma keying is discussed in this thesis. For the chroma keying and vector keying methods to work colour information is necessary. In gray scale images a different approach is required to segment regions in an image.

To better understand chroma keying an understanding of colour is needed. Colour is a broad topic that encompasses several fields of study. A detailed description thereof is beyond the scope of this thesis. However a brief discussion of the colour model used in CRT and colour raster graphics is necessary.

**a) Colour**

A colour image is considered to be a two-dimensional vector field, $f(x, y) = [\ R, G, B\ ]$, consisting of three different components. These components are commonly red, green and blue. These are the primaries of the RGB colour model.

The RGB colour model is a hardware-oriented model used in CRT displays to represent colour. It provides a means of specifying a colour within a certain colour gamut (colour range). Red, green and blue constitute the primaries of this model and are additive, (i.e. red + green = yellow). The RGB colour model forms a 3D co-ordinate system of colour,

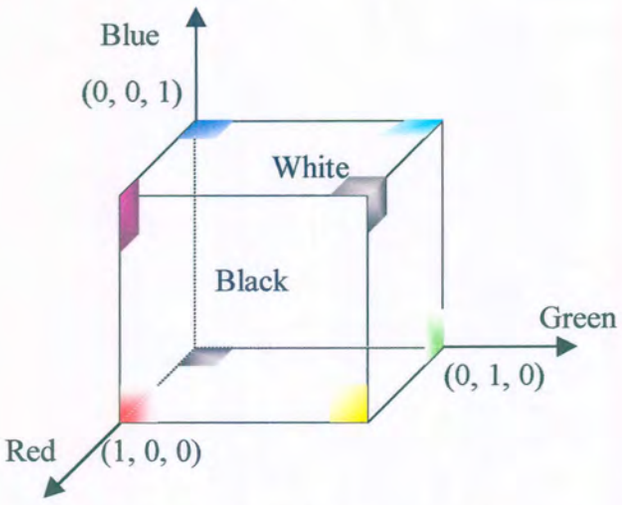



**Figure 3.28 RGB Colour Space**

with the primaries lying on the principle axes. The RGB colour space represents all the colours of a specific colour gamut that lie in the unit cube of this co-ordinate system. The RGB colour model although effective is not very user friendly when it comes to identifying colours. Figure 3.28 illustrates the RGB colour space.

A more intuitive approach for representing colour is the hue, saturation and lightness (HLS) colour model. Hue specifies a colour for example blue or red. Lightness specifies the intensity. Saturation gives an indication of how far a colour is from gray. For instance navy blue is highly saturated, while light blue is unsaturated.

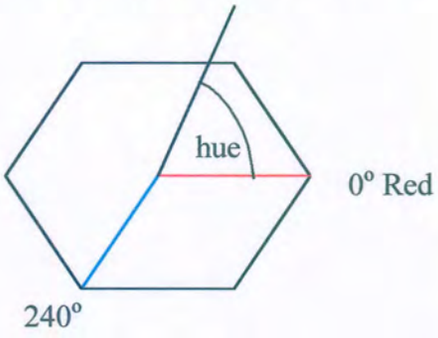



**Figure 3.29 HLS Color Model Viewed From Above**

The HLS colour space is geometrically represented by a hexcone. When viewed from above, as in figure 3.29, a hexagon is seen. The angle from the center to an edge represents the hue. Red lies at zero degrees and blue at two hundred and forty degrees. To select a specific colour a sector around a colour is targeted. Unfortunately images returned by an image capture device are not in this colour space. Images returned by a digital camera tend to be in the RGB colour space. These images can be converted to the HLS space or some other colour space but this requires that each pixel be translated from one space to another. This requires additional processing which is not necessary because there is a fast chroma keying method for the RGB model.

**b) A Fast Chroma Keying Algorithm**

[Van den Bergh] developed a fast software chroma-keying algorithm for images in the RGB colour space. The algorithm requires a maximum of five operations per pixel. The algorithm essentially consists of criteria that determine when a pixel is blue. A pixel is blue when its blue value is dominant. This means that:

$$B > R \text{ and } B > G \tag{3.2.1}$$

This criteria does not prevent impure shades of gray from being considered blue. For instance a pixel with values [ R, G, B ] = [ 0.99, 0.99, 1 ] which is almost white, will be considered blue by the above test. To exclude impure shades of gray and to overcome this problem a distance measure is needed. This distance measure simply states that if a pixel is above a certain threshold, $\mathbf{d_{max}}$, then the pixel will appear blue. This constraint (distance measure) is expressed as follows:

$$d = \sqrt{(B - R)^2 + (B - G)^2} > d_{max} \tag{3.2.2}$$

This distance measure is computationally expensive because of the square root and multiplications. A computationally inexpensive distance measure that can be used is:

$$d = 2B - R - G \tag{3.2.3}$$

Although the measure expressed by equation 3.2.3 is not a true distance measure it is a suitable, effective and efficient method of expressing the distance constraint. The first constraints are illustrated in figure 3.30, which shows the RGB cube with a skew pyramid. The volume of the skew pyramid represents those colours that satisfy the first constraint, **B > R** and **B > G**.

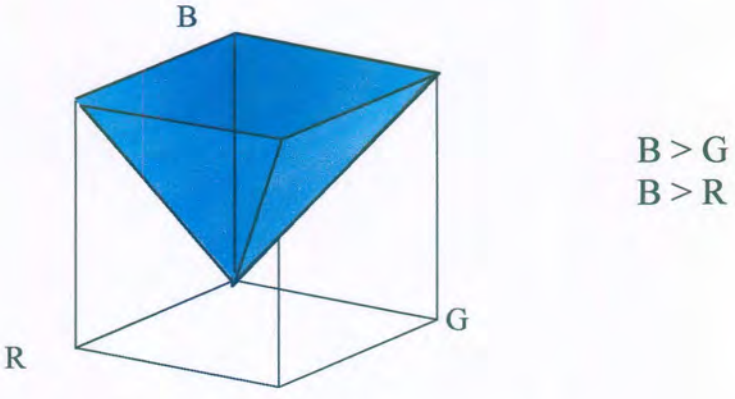



**Figure 3.30 First Set of Constraints**

The distance measure simply selects a portion of this pyramid which is a sufficient distance away from gray.

To this point the algorithm presents a means of determining if a pixel in the RGB colour space is blue or not and is suitable for image segmentation. However because only simple thresholding is used an image segmented by this method will have sharp edges. The algorithm can be extended for blending by using masks and an alpha value to make the edges smooth [Van den Bergh].

## 3.2.2 Image Moments

Image moments present a fast and robust means of calculating useful summaries of global image information or averages. Moments are sums over many pixels and are therefore not susceptible to small pixel value changes. Image moments give a measure of various useful features of a rectangle with the same moments as an image or shape; this is illustrated in figure 3.31.

**Image of Object** | **Rectangle with Same Moments as Image**

**Figure 3.31 Illustration of Image Moments**

The useful features of a shape (image of an object), determined by using image moments include:

- position, $x_c$, $y_c$,
- orientation, $\theta$, and
- the dimensions, $L_1$, $L_2$.

These features are illustrated in figure 3.32.

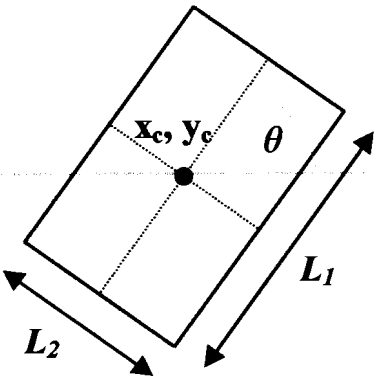



**Figure 3.32 Features Calculated by Image Moments**

Image moments are calculated as follows:

$$M_{00} = \sum_x \sum_y I(x,y) \qquad M_{11} = \sum_x \sum_y xy\,I(x,y)$$

$$M_{10} = \sum_x \sum_y x\,I(x,y) \qquad M_{01} = \sum_x \sum_y y\,I(x,y) \qquad (3.2.4)$$

$$M_{20} = \sum_x \sum_y x^2\,I(x,y) \qquad M_{02} = \sum_x \sum_y y^2 I(x,y)$$

where $I(x, y)$ is the intensity at pixel x, y. From the moments above the features of the object are calculated as follows:

$$x_c = \frac{M_{10}}{M_{00}} \qquad y_c = \frac{M_{01}}{M_{00}} \tag{3.2.5}$$

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \qquad \theta = \frac{\arctan(b/(a-c))}{2} \tag{3.2.6}$$

$$b = 2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right) \qquad L_1 = \sqrt{6\left(a + c + \sqrt{b^2 + (a-c)^2}\right)} \tag{3.2.7}$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2 \qquad L_2 = \sqrt{6\left(a + c - \sqrt{b^2 + (a-c)^2}\right)} \tag{3.2.8}$$

### 3.2.3 Stereo Matching

Many computer vision algorithms require some form of matching to be done. Stereo matching entails identifying point correspondences in stereo images. Matching algorithms match tokens in one image with tokens in another by using certain features (attributes) [Yang & Gillies]. An example of a token is a pixel, while an attribute could be the intensity of the pixel. A token can be anything from a single pixel or edge to a curve or image region. The type of token or attribute to be matched determines the type of matching algorithm to use.

The main difficulty in stereo matching is to find matching point correspondences. A brute force approach would be to match all possible tokens in one image with all the tokens in another. However, this is far too computationally expensive. Therefore constraints are used to reduce the search space for matching corresponding points. There are several types of constraints that are used to reduce the search space. These include the following:

- geometric constraints that occur due to the image system, e.g. the epipolar constraint used to reduce a 2D search space to a 1D space,

- object geometric constraints which state that one can assume that the distance an object is from the imaging system varies slowly, and
- physical constraints, e.g. the way in which light affects a model.

Matching of moments or extracted features of an object are done at the geometric level in two dimensions. This type of matching may be extended to 3D by using distances between vertices and angles between adjacent surface normals. The epipolar constraint is very advantageous for matching because it reduces the search time and improves accuracy.

Four categories of stereo matching algorithms exist. These are:

- correlation-based in which intensity is used to correlate areas,
- relaxation-based, in which a guessing scheme associates corresponding pixels by making use of some information. These associations are then shuffled by propagating the different constraints,
- dynamic programming, in which a cost function with a large number of variables is minimized, and
- prediction and verification. Tokens which have more meaning and carry more information than pixels are used for matching.

Some correlation-based algorithms find and match points of interest for only a few points in an image. This is an effective approach in cases where limited computing resources are available. When suitable hardware is used, correlation is normally performed on all the points in an image by using constraints. Of all these points only valid matches are retained.

It is important for the computer vision algorithm to extract or find key features of an object or shape. These features are used as tokens for stereo matching. Good features to use, especially for the Reflections method, are those that convey aspects of the image's shape. If stereo views of an object's skeleton are matched then five or even six degrees of

freedom are obtainable by using the Reflections method. Using an object's corners or endpoints is an effective way to match a correlated pair of points in stereo images.

To round off the CV theory an edge and corner finding algorithm which is robust, fast, accurate and noise resistant is discussed.

### 3.2.4 SUSAN

SUSAN (Smallest Univalve Segment Assimilating Nucleus) [Smith & Brady] is a low level approach to image processing with the ability of detecting features such as edges and corners (including junctions). It is also used to do structure preserving noise reduction. It makes use of non-linear filtering to determine those parts of an image which are closely related to each individual pixel.

The way in which SUSAN works is by associating a local area of similar brightness with each pixel in an image. This area of similar brightness is found by placing a circular mask with a center pixel (the nucleus) at every pixel in an image. The brightness of the nucleus

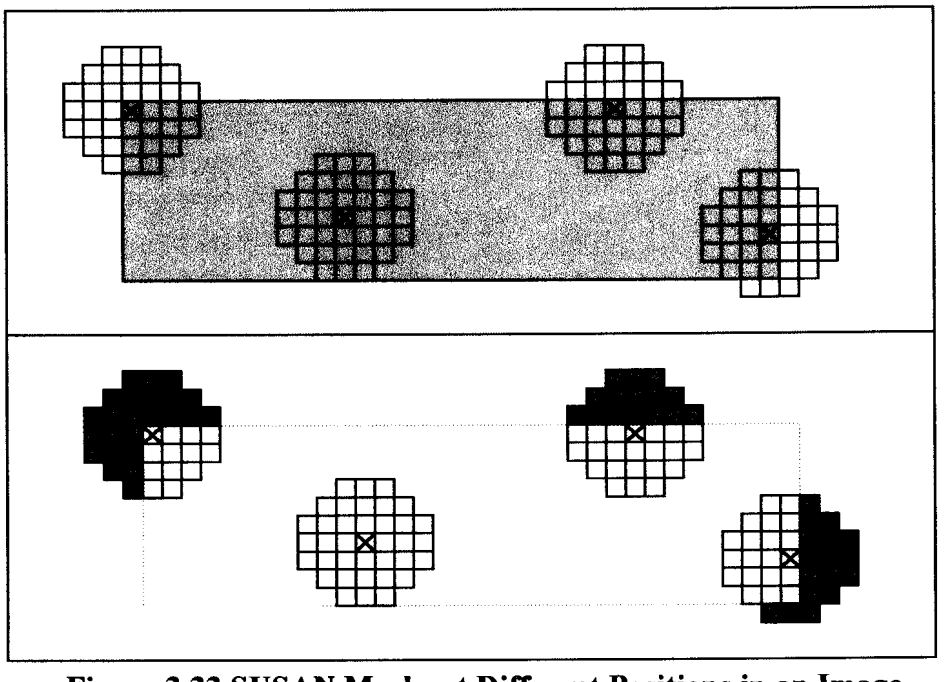


**Figure 3.33 SUSAN Masks at Different Positions in an Image**

is compared with the brightness of each surrounding pixel in the mask and those pixels with similar brightness are added together to give the total area of similar brightness. This area is known as the USAN (Univalue Segment Assimilating Nucleus).

The USAN contains information about the structure of an image. Figure 3.33 illustrates four masks placed at four different positions in an image. The mask lying in the flat region has a maximum USAN area. Those masks lying on a straight edge have USAN's that are half the area of the mask whereas those masks lying near corners have USAN areas that are below half the maximum USAN area. As a mask approaches an edge the area of the USAN decreases. As it approaches a corner the area becomes even smaller. By finding local minima the exact positions of image corners are found.

The SUSAN principle states that:

> "an image processed to give as output inverted USAN area has edges and two dimensional features strongly enhanced, with the two dimensional features more strongly enhanced than edges." [Smith & Brady, 1995].

Two-dimensional features being corners and junctions while one-dimensional features refer to edges. This algorithm differs from conventional approaches in that:

- no image derivatives are used (This is the reason why performance remains good, even in the presence of noise), and
- no noise reduction is necessary.

Non-linear filtering and the summing (integration effect) of those pixels with similar brightness produce strong noise rejection. If there are not multiple features within a mask, then SUSAN is not influenced by the uncertainty principle which states that, "the better the detection quality the worse the localization of the detected features". The localization of features using SUSAN is independent of the mask size as long as there are no other features within the mask region.

**(a) Edge Detection**

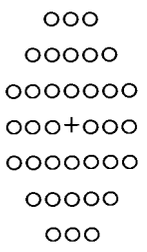


**Figure 3.34 A Typical SUSAN Mask**

To implement an edge detector using SUSAN a circular mask is placed at every point in an image. Commonly a mask with radius equal to 3.4 pixels is used. Such a mask has an area of 36 pixels around the nucleus, as illustrated in figure 3.34.

By applying the mask to a pixel an edge response for that pixel is obtained. The process to determine an edge response for each pixel is as follows:

(1) Compare the brightness of each pixel within the mask to the nucleus. A simple equation for doing this comparison is:

$$c(\mathbf{r}, \mathbf{r}_0) = \begin{cases} 1 \text{ if } |I(\mathbf{r}) - I(\mathbf{r}_0)| <= t \\ 0 \text{ if } |I(\mathbf{r}) - I(\mathbf{r}_0)| > t \end{cases} \quad (3.2.9)$$

where $\mathbf{r}_0$ is the position of the nucleus (center pixel) and $\mathbf{r}$ is the position of any other point in the mask. $I(\mathbf{x})$ is the intensity of the pixel at position $\mathbf{x}$.

An alternative and smoother equation to use is:

$$c(\mathbf{r}, \mathbf{r}_0) = e^{-\left(\frac{I(\mathbf{r}) - I(\mathbf{r}_0)}{t}\right)^6} \quad (3.2.10)$$

This function allows for a slight variation in brightness without affecting the result too much. It gives good stability about the threshold. This function should be implemented using a lookup table to improve speed.

The threshold t, determines the minimum contrast of features (edges) and the maximum noise that is ignored. Performance is not dependent on fine-tuning t.

(2) Next these values are summed giving the USAN (the area within the mask of pixels with similar brightness).

$$n(\mathbf{r}_0) = \Sigma\, c(\mathbf{r}, \mathbf{r}_0) \qquad (3.2.11)$$

(3) n is now compared with the geometric threshold, $g = 3n_{max}/4$ where $n_{max}$ is the maximum value n can be (n is equal to the number of pixels in the mask surrounding the nucleus).

The initial edge response then becomes:

$$R(\mathbf{r}_0) = \begin{cases} g - n(\mathbf{r}_0) & \text{if } n(\mathbf{r}_0) < g \qquad (3.2.12) \\ 0 & \text{Otherwise} \qquad (3.2.11) \end{cases}$$

(4) The edges in an image are found after non-maximum suppression is done.

It is necessary to find the direction of an edge. There are three reasons for this:

(i) it is necessary for non maximum suppression,

(ii) to localize an edge to subpixel accuracy, and

(iii) applications may use attributes of an edge pixel such as edge direction, position and strength for matching purposes.

By using the USAN the direction of an edge is found in the following way: The first step in finding the direction of an edge is to decide what type of edge is being dealt with. In figure 3.35 two different types of edge points are shown.

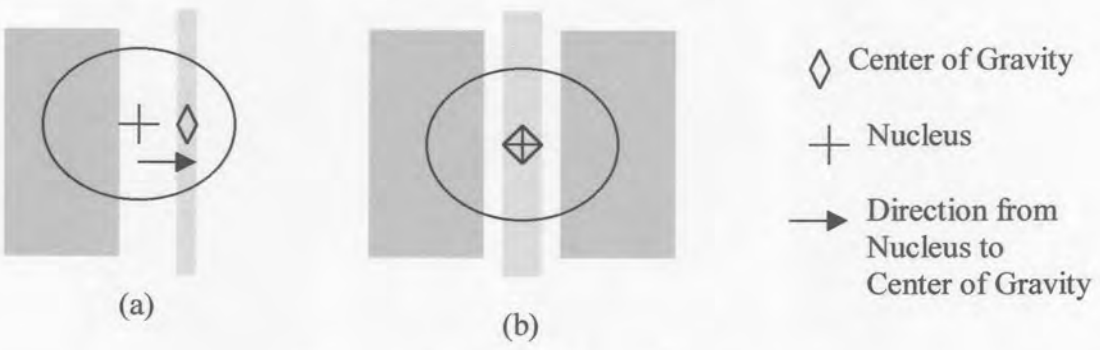



**Figure 3.35 Two Different Edge Types**

The first type of edge points are those edge points that lie on one side of an edge (ideal step edges). This first case is referred to as the inter-pixel edge case which occurs when the vector between the nucleus and center of gravity of the USAN is perpendicular to the local edge direction. Figure 3.35, (a) illustrates edge points that fall into this case. To find the direction of such an edge, the center of gravity and a vector **v** from the nucleus to the center of gravity are found. The direction perpendicular to **v** gives the direction of the edge. The center of gravity is calculated as follows:

$$\mathbf{CoG}(\mathbf{r}_0) = \frac{\sum \mathbf{r}\, c(\mathbf{r}, \mathbf{r}_0)}{\sum c(\mathbf{r}, \mathbf{r}_0)} \qquad (3.2.13)$$

For cases where edge contrast is high or the real edge projects very close to the center of a pixel rather than in between pixels (for non-sharp step edges) the intra-pixel edge case is encountered. It exists for points that lie on a thin line for which the brightness is half way between the brightness of two regions that form the edge. In these cases the USAN is a thin line in the direction of the edge. This is illustrated in figure 3.35 (b). The direction of such an edge is calculated by finding the longest axis of symmetry. The following second moment calculations are used to estimate the direction of an edge:

$$\overline{(x - x_0)^2}(\mathbf{r}_0) = \sum (x - x_0)^2\, c(\mathbf{r}, \mathbf{r}_0) \qquad (3.2.14)$$

$$\overline{(y - y_0)^2}(\mathbf{r}_0) = \sum (y - y_0)^2\, c(\mathbf{r}, \mathbf{r}_0) \qquad (3.2.15)$$

$$\overline{(x - x_0)(y - y_0)}(\mathbf{r}_0) = \sum (x - x_0)\,(y - y_0)\, c(\mathbf{r}, \mathbf{r}_0) \qquad (3.2.16)$$

The direction of an edge is estimated by the ratio of *(3.2.14)* to *(3.2.15)* while the sign of the gradient is determined by *(3.2.16)*.

One question remains, "How does one decide which case to use?". If the USAN area is smaller than the diameter of the mask (in pixels) then the intra-pixel edge case holds true. This should be clear from figure 3.35. If the USAN is greater than or equal to the diameter then the inter-pixel edge case applies and the center of gravity is found and the direction of the edge is determined. However, there is one exception. If the center of

gravity is less than one pixel away from the nucleus then the intra-pixel edge case applies since this case is a better fit for this situation. This occurs when the intermediate brightness band is more than one pixel wide and a large mask is used.

If an edge is blurred or smoothed the area at the center of the edge will decrease causing edge responses to increase. This means that for blurred or smoothed images edge responses improve. Once again the size of a mask does not affect accuracy. The USAN area occurs on top of the edge regardless of the mask size. After applying non-maxima suppression and thinning, sub-pixel estimation is performed if necessary. The final step of the edge finding process is to suppress non-maxima responses in the direction perpendicular to an edge.

**(a) Corner Detection**

The corner finder is similar to the edge finder. The first few steps are identical for both the corner and edge detector. The procedure for finding corners is as follows:

(1) Place a circular mask around a specific pixel.

(2) Use equations 3.2.10 and 3.2.11 to calculate the area in the mask with similar brightness to the nucleus (find the USAN).

The next step for the corner detector is similar to the edge finder but uses a lower value for the geometric threshold.

(3) Use equation (3.2.12) to subtract the USAN size from the geometric threshold. The geometric threshold is set to half $n_{max}$, i.e. $g = 1/2\ n_{max}$.

For the edge detector it is only necessary to use g when there is noise. If the mask lies at a corner then n will be less than half of $n_{max}$. Using this value for the threshold is suitable because of the nature of a quantized straight edge [Smith & Brady].

(4) Find the USAN's centroid and contiguity and test for false positives. This step removes falsely reported corners such as edges or noise.

False positives occur in real data when boundaries between regions are blurred. Finding the center of gravity and its distance from the nucleus can be used to identify and

eliminate false responses. This is because the USAN of a corner will have a center of gravity that is not near to the nucleus. In the intra pixel edge case the USAN will form a thin line that passes through the nucleus. In this case the distance from the center of gravity to the nucleus will be small. In real images or images scattered with noise or fine complicated structure enforcing contiguity in the USAN aids the removal of false positives. A true corner will have all the pixels within its mask lying in a straight line pointing outwards from the nucleus in the direction of the center of gravity as part of its USAN. This effectively forces the USAN to have a degree of uniformity and reduces false positives in reported corners. This is illustrated in figure 3.36.


**Figure 3.36 Testing Contiguity**

(5) The final step of the process is to suppress non-maxima corner responses in the set of corners reported.

SUSAN is very advantageous for stereo matching, because SUSAN calculates an USAN for each pixel. Thus each pixel has several attributes that characterize it. These attributes are used for identifying and matching corresponding pixels in stereo images. Some of the attributes reported corners have, in addition to position, are:

- the value of image brightness at the nucleus, and
- the position of the center of gravity of the USAN with respect to the center pixel (nucleus).

It is important to realize that these attributes often remain fairly constant between images.

## 3.3 TRACKING

Tracking is crucial for interactive applications that must function at frame rate speeds, or for other applications which simply must follow the location of an object from frame to frame. Implementations of trackers vary from very simple trackers that use windows which track a single point of an object such as the center of gravity, to highly complex predictive trackers that track the entire shape, attitude and position of an object. A simple window based tracker and a predictive Kalman filtering tracker are discussed below.
A good tracker first finds relevant information. Before an object is tracked the shape of the object must be verified, i.e. the object to be located must correspond to the desired object to be tracked. If this is not done the wrong object may accidentally be tracked.

### 3.3.1 A Window Based Tracker

A simple tracker can be implemented by using a window, where a window is a sub-image of a frame and a frame is the entire image captured by the camera. By using image moments the center of gravity (centroid) of an object or shape in a frame or window is found. Once the centroid is calculated the window is placed around the shape, centered at the centroid. The window is moved from frame to frame so that the centroid found in the last frame becomes the position at which the window is centered in the current frame. This is illustrated in figure 3.37.

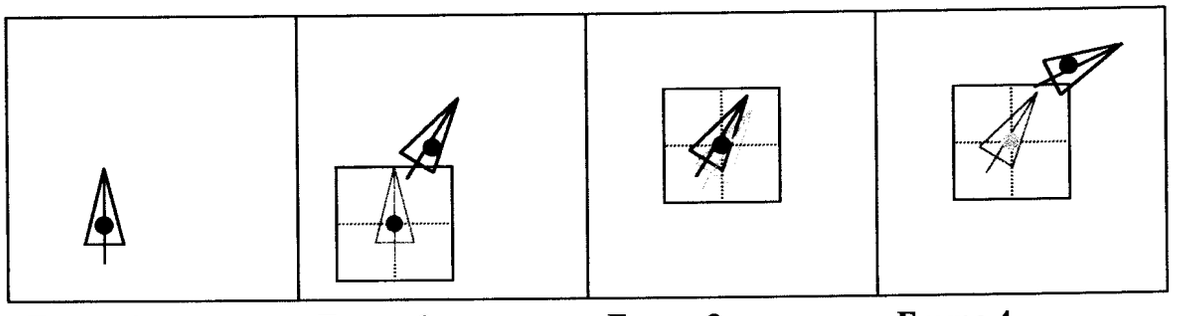



**Figure 3.37 A simple Window Tracking System**

In this simple tracking system the window is always one step (or frame) behind the object, unless of course the object does not move from frame to frame, as illustrated in figure 3.37 in frame three and frame four. When an object does not move, the centroid remains the same for two consecutive frames and so the window position also remains the same. It is possible to use information from previous frames to predict where the object will be in the next frame. This allows the window to be positioned more accurately with the result that there is less chance of the tracker losing the object. Hence, a more robust tracker can be implemented. Implementing prediction tracking is not always necessary. For instance if a large window is used and the cameras frame rate is high then an object will have to move extremely fast to escape the tracking window [Van den Bergh]. For the object to escape the window it will have to move more than half of the windows length from one frame to the next.

### 3.3.2 A Predictive Curve Tracker

Although there are several trackers that track points or even polyhedral shapes, there are not many trackers that track the entire curve of a non-polyhedral object. Yet [Blake & Isard] are able to track the curved silhouettes of moving non-polyhedral objects at full frame rate (50 Hz). They successfully track the position, attitude and shape of non-polyhedral objects such as hands and lips.

The theory that lies behind such a tracker includes deformable models, B-spline curve representation and control theory. In their work a tracker is an estimator for a moving, piecewise-smooth image plane curve. The problem is to estimate the motion of the curve. A curve is represented using B-splines of some predefined form with control points varying over time. The tracker estimates the control points in such a way that the control points represent a curve, at each time step, which closely matches some underlying curve. The tracker makes use of Kalman filtering. A Kalman filter has two parts. The first is a prediction model which uses information from past frames and information about the dynamics of an object to make predictions about the current position of the curve. The second aspect of the Kalman filter is measurement assimilation. This combines

information from a frame with the latest prediction. The tracker consists of a system model, which specifies the dynamics of a curve over time relative to some average shape. The measurement model specifies the positions along a curve at which measurements are made and the reliability of these measurements. The underlying control points that represent a curve are obtained using an interactive drawing tool for a specific frame.

The above methods only track a single object. Tracking multiple objects is more challenging and needs to be considered.

### 3.3.3 Multiple Object Tracking

The difficulties encountered when tracking multiple objects include noise, aperture (which occurs when considering only the local motion of an object) and occlusion. Occlusion is the most problematic of these. When one object moves between the camera and another object either partial or complete occlusion occurs. Prior information or knowledge about the objects seen and how they occlude each other presents one means of overcoming occlusion. This information gives insight as to how occlusions are to be handled and how boundaries of an object are to be segmented. Often a more advanced segmentation algorithm is needed to deal with occluding objects [Corbett].

To track multiple moving objects in real-time two features are commonly needed:

- motion estimation. Motion fields are examined to gain insight into the real 3D motion fields of an object
- object association. Features should be stored for objects between frames which can be used to identify and associate appropriate objects from frame to frame.

A simple way of dealing with multiple object tracking for the simple window based tracking system proposed above is to consider two scenarios that can occur:

(1) the overlapping of windows which suggests the possible overlapping and occlusion of objects, and

(2) windows which do not overlap, indicating that there are no occlusions.

In the second case each object in each of the different windows is found and removed from the image. In the first case, in which objects may occlude one another, a smarter segmentation algorithm is needed. Perhaps combining contour and region information will allow the algorithm to ascertain what an image would look like if an occluded object or occluding object were segmented. Advances in CV allow for such image segmentation to be done where objects that occlude other objects are successfully removed and the objects in the background are left in the image. Figure 3.38 illustrates an example of such a dexterous computer vision algorithm, courtesy of the Huji CV lab. Such algorithms will be invaluable especially if the identification and feature extraction algorithms are not able to handle partially occluded shapes.

**Figure 3.38 Segmenting Occluded Objects.**
*(Pictures Courtesy of Shmuel Peleg & Hebrew University of Jerusalem)*

## 3.4 SUMMARY

In this chapter the theory of catadioptric stereo is discussed in detail in section 3.1. From this discussion it is clear that there is far more to catadioptric stereo than meets the eye. For instance lighting, the type of object to be tracked and how to install the apparatus require considerable thought and planning. The essential apparatus in such a system includes a single camera and one or more mirrors. This apparatus must be installed with precision for accurate 3D calculation. The 3D calculation requires a calibration phase, an image analysis phase and a 3D calculation phase. The 3D calculation phase includes a pre-processing step which is performed only once for a Reflections set up. If sufficient

points on an object are matched then the 5D and 6D information of an object are obtainable.

A suitable computer vision algorithm is required. This algorithm must locate and track an object, find structural features of this object in the stereo views and match these features. Several useful computer vision algorithms for performing these tasks are described in section 3.2. These algorithms include image moments, a fast chroma keying algorithm and SUSAN. Image moments are useful for calculating features such as the centroid, orientation and dimensions of an object. The chroma keying algorithm is an effective means for identifying blue pixels in an image and is useful for image segmentation. SUSAN is an edge and corner detector. Both a window tracker and predictive contour tracker are introduced in section 3.3. The chapter ends with a discussion on multiple object tracking.

# Chapter 4

# Reflections in Interactive Applications

In this chapter the implementation and design of the Reflections method are discussed. The main focus of this chapter is on two applications in which this method is used to implement natural and non-intrusive interaction. These applications are the Virtual Drums project and Ndebele Painting. Many aspects of creating a Reflections system are discussed in the theory section, several of which are implemented in the development of the above systems. This chapter concludes with a discussion on two implementations of Reflections in different virtual environments.

## 4.1 DESIGN AND IMPLEMENTATION

There are two main processes involved in a Reflections system. The first deals with the physical installation of the camera and mirrors. The second part encompasses the software used to calculate the necessary information. The physical installation includes setting up the equipment and reporting measurements of the installation and equipment. The software aspect consists of those algorithms and processes that are necessary to capture and process individual frames from the video camera and which calculate the 3D, 5D or 6D information. Figure 4.1 shows these processes and their different components.

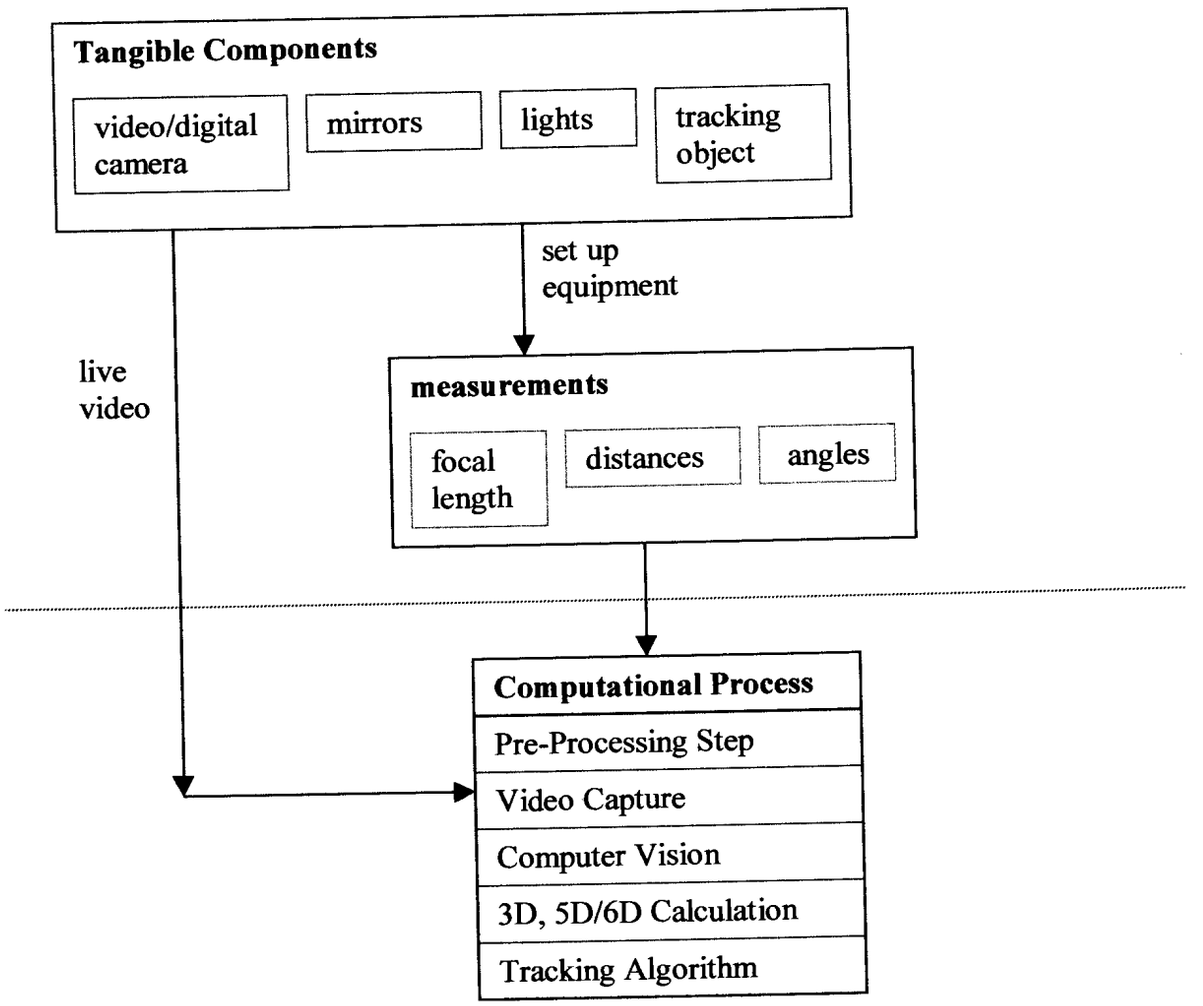



**Figure 4.1 Flowchart of a General Reflections System**

The tangible components of the s rors, computer, lighting and the object to be tracked. Installing these elements and taking relevant measurements of the equipment and the placement thereof constitute the physical installation process of the tangible components. Once the equipment is in place, a sequence of frames are captured by the video camera and processed. From each processed image the 3D and 5D or 6D information is calculated. This part of the system forms the computational process. The computational process requires the implementation of different software components. These include:

- a video capture component,
- an appropriate computer vision algorithm,
- a matching algorithm,
- the calculation component, and
- a tracking algorithm.

A closer look at what the two main processes entail is given below.

### 4.1.1 The Physical Installation Process

Before any computation is performed and the 3D information calculated, the camera and mirrors need to be installed. The following guidelines can be used to assist one when installing a Reflections system:

(1) position the image capture device so that it views the required mirrors and region of interaction,

(2) place the reflective surfaces in suitable positions with suitable orientations so that the region of interaction is viewable by the camera,

(3) install the lighting and necessary filters, and

(4) measure the position and rotations of the mirrors relative to the camera.

When orienting the mirrors and installing the equipment it is necessary to see the view of the camera. This is important for correctly and optimally viewing the region of interaction. For this reason the installation process contains a video capture software component for simply displaying live video. In the systems implemented in this thesis the

computational process has the option of capturing and displaying frames without performing any calculation.

The setup and placement of the camera and mirrors need to satisfy the constraints of a Reflections system and must at the same time still meet the needs of the interaction that is to take place. The constraints and requirements that need to be met when setting up the equipment are the following:

- the camera and mirror must be placed in such a way that the required region of interaction is clearly visible in the two different views,
- the camera and mirror should be placed in such a way as to minimize occlusions,
- the camera and mirror should be placed so that a clear view of the object is seen in the two different views, and
- the lighting should be suitable. Stable lighting should be used and placed in such a way that the details of an object that need to be seen by the camera are distinct and clear.

Choices concerning the number of mirrors to be used in a set up and where to place them and the camera, to a large degree depend upon the specific application and the size of the desired region of interaction.

The application environment determines what type of object is to be tracked. A coloured object is tracked when suitable lighting is available. In settings with low light levels an (self-luminous) object or a torch is used as the tracked object. In some environments it makes sense to use infrared lighting.

Certain environments and interactions may require more sophisticated computer vision algorithms and also place certain restrictions on the positioning of the equipment. All of these aspects need to be taken into consideration when installing the equipment.

Finding the optimal position for the equipment requires careful planning and thought. This is because there is a tradeoff between accuracy and the size of the interaction

volume. In some environments it is difficult to find a suitable position to place the camera and mirrors while maintaining the required accuracy and still covering the entire region of interaction.

Certain measurements are taken when the camera and mirrors are in place. These measurements include the relative positions and orientations of the mirrors and camera. The focal length of the camera is another crucial measurement that is needed. The approach in section 3.1.6 is used to calculate the focal length if it is not already known. The installation phase is completed when the equipment is in place and the necessary measurements are taken.

### 4.1.2 The Computational Process

The components of the computational process are introduced in figure 4.1. A flow chart outlining the steps followed in the computational process is given in figure 4.2.

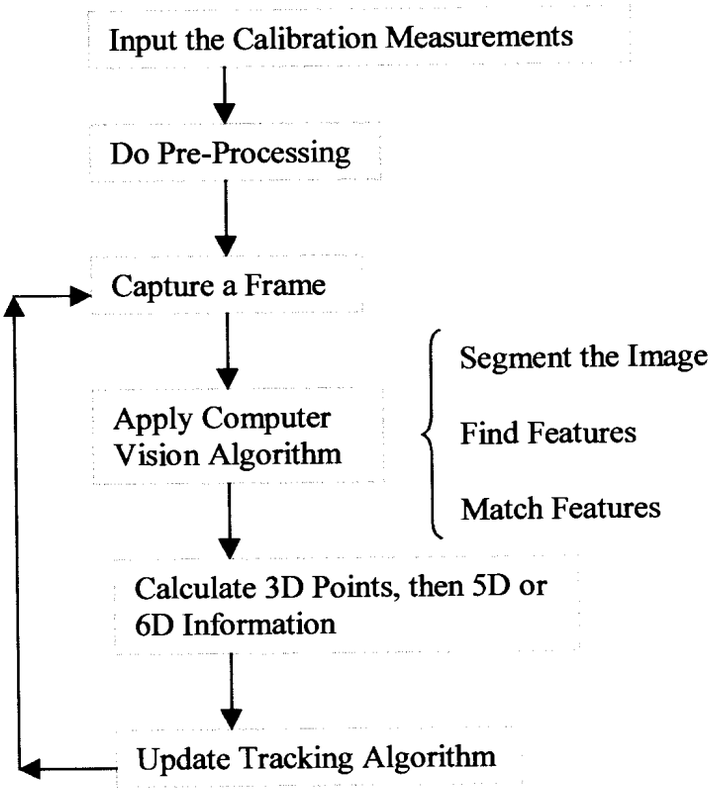



**Figure 4.2 Flow Chart of the Computational Process**

The computational process receives as input the measurements taken in the physical installation. These measurements are used in a pre-processing phase in which all the initial calculations are performed. The initial calculations include determining the distance of the projection of the camera onto the mirror when using the trigonometric approach to 3D calculation. For both the algebraic and trigonometric approaches the plane equations that mathematically represent the mirrors are calculated. For the algebraic approach the inverse change of co-ordinate system matrices are calculated. The preprocessing phase is very important since it calculates information at the outset of the computational process that only needs to be calculated once and which will be used in future calculations. Performing these calculations once at the start reduces the amount of processing that is done during the computer vision process while frames are being captured. Capturing frames and processing them is computationally intensive. Therefore as little as possible additional processing should be done during this stage of the process.

After calibration and preprocessing the video capture algorithm is started. Frames are captured by the video camera and stored in memory (a frame is stored in memory until it is replaced by a subsequent frame). Once a frame is in memory the computer vision algorithm processes and analyzes this frame. The algorithm segments the image, extracts meaningful features and matches corresponding features in the two different views. Once a set of corresponding points matched by the computer vision algorithm are found the 3D calculation is performed. If multiple sets of points on an object are found then the orientation of that object is determined by using the 5D/6D-calculation process described in section 3.1.7.

The tracking algorithm tracks objects of interest from frame to frame. In this way the video camera is used as an input sensor and tracking device for interactive interaction.
The steps of capturing frames, performing the computer vision algorithm on those frames, calculating the multidimensional information and tracking objects of interest, are performed repeatedly while the program is running.

This concludes the description of the design of the system and the procedure to be followed when implementing it. The Reflections system is implemented in both the applications of the Virtual Drums project and Ndebele Painting in VR. These real-world implementations and a detailed description of the different computer vision techniques used and the practical solutions implemented are discussed next.

## 4.2 VIRTUAL DRUMS

The Virtual Drums project aims to create a simulator of a real drum kit in virtual reality, which a person can play in an identical manner to playing a real drum kit. It was in the creation of this virtual drum kit that the usefulness of the Reflections method was realized.

### 4.2.1 The Application

The virtual drum kit has three main components. These components are illustrated in figure 4.3.


**Figure 4.3 Components of a Virtual Drum Kit**

**(a) Visualization**

The first aspect of the virtual drum kit is the visual component. This includes using 3D computer graphics to create a realistic model of a drum kit. A picture of a real drum kit is

shown in figure 4.4 This real drum kit was used as the model for the creation of the AVANGO version of the virtual drum kit.


**Figure 4.4 A Yamaha Drum Kit**

To make the virtual drum kit realistic, the visualization of the drum kit includes cymbals which swing back and forth realistically and pounding drums with some visual effect that illustrates when a drum is struck. The visual effect chosen for the drums is a blue ripple which moves outwards from the center of the drum. This visual cue is representative of the sound wave produced when a drum is played. Along with these visual effects it is important to visualize the drumstick. The position and orientation of the drumstick are graphically represented to provide the drummer with visual feedback for his movement of the real drumstick. A wood textured cylinder is used to visualize the drumstick.

**(b) Sound**

It is essential that this virtual drum kit have a realistic audio component. To do this each cymbal and drum has an appropriate drum sound associated with it. Appropriate sounds for each drum and cymbal were obtained and connected to the appropriate pieces in the visual model. Additionally spatial sound is used which gives an indication of the source and location of a sound. This enhances the apparent realness of the virtual drum kit. Volume may be used to indicate how hard a drum or cymbal is struck.

**(c) Interaction**

The final part of this drum kit is the interaction aspect. Playing the virtual drum kit should be identical to playing a real drum kit, or as close as possible to it. Allowing the drummer to play the drums and cymbals with real drumsticks accomplishes this to some degree. This requires the interaction aspect of the drum kit to monitor the position and orientation of the drumsticks. To do this a six degrees of freedom tracker or even a five DOF tracker is essential. Furthermore the tracker should not restrict the user's movement in any way. The tracker must therefore be wireless. Because the Reflections method provides a means for satisfying these requirements and demands, it is used to implement the interaction aspect of the Virtual Drum kit.

It is necessary that the position and movement of the real drumsticks be related to the virtual world. The audio and visual effects are triggered as the sticks move into positions where the drums are located. This is accomplished by monitoring the position and motion of the real drumsticks. It is also possible to track the velocity and acceleration of the drumsticks.

Finally there is the issue of tactile and force feedback. Allowing the user to use real drumsticks already provides the drummer with some tactile (touch) feedback. However, the issue is to simulate the reverberation that is felt when a real drum is struck. Perhaps one simple solution to this problem is to use drum-putty (a substance that makes almost no sound when struck, but which allows a drumstick to bounce of it as if bouncing off a real drum). All that is needed then is to project the image of the drums onto the surface of this reverberative material and to position this material appropriately. However, force feedback is beyond the scope of this thesis and consequently it is not considered here.

### 4.2.2 The Prototype

The original implementation of the Virtual Drums project has the following features:

| | | |
|---|---|---|
| Graphics | - | OpenGL |
| Sound | - | Bergen Sound Server - Simple playback of sound files |
| DOF | - | 3D position |
| Calculation | - | Trigonometric/Geometric 3D Calculation |
| Environment | - | Desktop |
| Tracked Object | - | Light Point |
| Lighting | - | Low Light Environment |
| CV Complexity | - | Elementary Computer Vision Algorithm |

This prototype of the virtual drum kit illustrates the use of the Reflections method in a desktop environment. The graphics and visualization of the project are implemented using OpenGl and C++. The drum kit has animated cymbals and an animated blue ripple that is used as the visual effect which is triggered when a drum is played. Figure 4.5 is a screen snap shot illustrating the graphics of the original Virtual Drum kit.

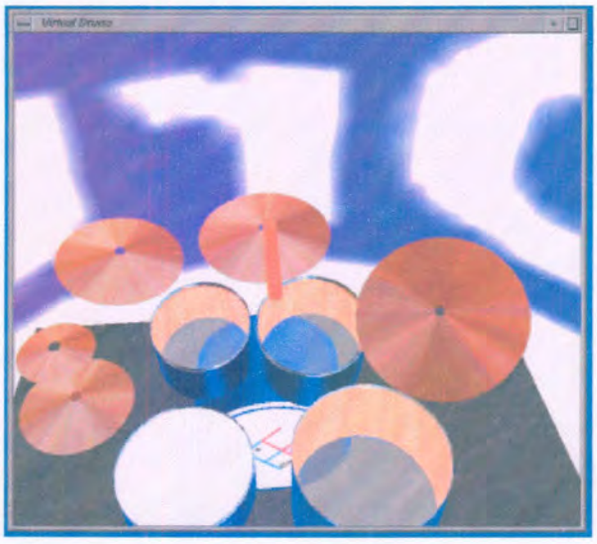


**Figure 4.5 Visualization of the Original Virtual Drum Kit**

The sound component is implemented using the Bergen sound server. This sound server plays the ".wav" sound files for the appropriate drums and cymbals. Spatial sound and volume are not catered for in this first application.

Interaction is implemented using the Reflections method. The application tracks a light source (a small green luminous object) placed on the tip of a drumstick. Only one drumstick is tracked in this first implementation. A point on this light source is located in both of the views seen by the camera via the mirrors. To locate these points the computer vision algorithm searches for a point in the image with an above threshold green value. This requires that there must be minimal lighting in the physical environment. Since the approach tracks a single point on the object, it only calculates the 3D position of the light object. Although the prototype does not track the orientation of the drumstick, it does prove the point that Reflections is a suitable approach for implementing 3D interaction.

The camera and mirror are installed in the desktop environment as follows:
Two mirrors are used. These are placed about two meters to the left of the user, who sits in front of the desktop monitor. The size of the first mirror is 40x20 cm and the size of the second mirror is approximately 2x3 cm. The digital camera looks down onto these mirrors. It sees two reflections (from different angles and positions) of the interaction region. The lights in the room are turned off and the small luminous object is attached to the end of a real drumstick.

The window based tracker described in section 3.3-(a) is implemented in this system. The tracker is implemented for both of the stereo views and tracks the position of the object in the two different views.

The position of the luminous object is tracked using the trigonometric method of calculation. Although this method is not designed to work for two mirrors, it is possible to use it in this setup because the small mirror is placed so close to the camera. This allows the assumption to be made that light seen in this mirror passes directly through the pinhole of the camera. Although this assumption is a cause for error it once again proves that the system is capable of tracking 3D.

To calculate whether or not the drumstick is moving up or down, the current height of the object is compared to its height in its previous position. If the drumstick is found to be moving down, then its position is used to determine if it is moving through a cymbal or drum cover. If it is moving down and through a piece in the drum kit then the appropriate sound and visualization are triggered.

Although this first prototype does not have all the many features that are required to create a fully immersive virtual drum kit it does illustrate that the Reflections method is suitable for natural non-intrusive 3D interaction.

### 4.2.3 AVANGO Implementation

A second version of the original Virtual Drums project is created. This second version extends and improves the first prototype. The following list summarizes the features of this second implementation:

| | | |
|---|---|---|
| Graphics | - | AVANGO |
| Sound | - | AVANGO Sound Server - 3D Spatial Sound |
| Environment | - | Cave & Desktop |
| Parallel Extension | - | CORBA - Reflections run parallel with Rendering System |
| Calculation | - | Algebraic 3D calculation |
| DOF | - | 5DOF ( 3D position & 2 DOF for rotation ) |
| Tracked Object | - | Light Object or Coloured Object |
| CV Complexity | - | More Elaborative & Extensive CV (SUSAN, Chroma keying, Flood Fill and Moments) |

The visualization and graphics are implemented in AVANGO using C++ nodes for the animation of the drum kit. AVANGO allows an almost seamless transition between different VR environments [Tramberend]. Therefore it is possible to run this version in CAVEs, workbenches and desktop environments. This version of the Virtual Drums is implemented both in CyberStage and in a desktop setting. Showcase was used to create the geometry for each of the different drum pieces which are saved in VRML format as Inventor files. SceneViewer is used to edit and touch up these pieces. The graphic model of the AVANGO drum kit is illustrated in Figure 4.6.


**Figure 4.6 Visualization of the Drum Kit in the CyberStage Implementation**

The audio component is implemented using the AVANGO sound server and therefore spatial sound is implemented in this version of the virtual drum kit. However, monitoring the velocity of the drumstick is not implemented and so varying the volume proportionally to the force with which a drum is played is not implemented.

In the CAVE installation, the CORBA extension of AVANGO is used to send the tracking data from an $O_2$ Silicon Graphics machine over a network to the fast Onyx machines. This extension makes it possible to run the interaction aspect as a standalone application on one computer while the graphics and sound are run in parallel on a separate computer. This is very advantageous because the processor running the interaction code does not spend valuable time on rendering operations and the computer handling the graphics and sound is not burdened with the image processing tasks. The position and orientation matrix of an object is sent from the computer processing the video to the machine processing the graphics. The graphics machine then uses this information to trigger the appropriate visual effects. It also uses this matrix to render the drumstick in its position as reported by the Reflections code.

Another advantage of using this AVANGO/CORBA extension is that it is not necessary to integrate the interaction code into the graphics application's code. This is highly desirable because the interaction aspect can be run as an interaction device for any application without having to integrate the code into that application. This is illustrated in the implementation of the Virtual Drums Project in that the Reflections code was created and developed apart from the AVANGO script which handles the visualization and audio aspect of the virtual drum kit. Integrating the Reflections code into the AVANGO scheme file is almost transparent to the programmer.

In the desktop environment a TCP client/server is used to send the tracking data from the Reflections application to the Virtual Drums program. This is not an entirely parallel approach because the Virtual Drums code waits for the tracking data.

**Figure 4.7 Desktop Installation of the Camera and Mirrors for the Virtual Drums Application**

In the desktop implementation two mirrors are used. These are placed against the ceiling with the camera as illustrated in figure 4.7. This is possibly the best position for the camera and mirrors as it is the position in which occlusions are minimized and from

which almost the whole length of the drumsticks are seen. It is important to see the length of the drumsticks in order to calculate the 5D information.

In the CyberStage implementation of the Virtual Drums project the large floor projection mirror is used with two smaller mirrors. The camera and other mirrors are placed at the rear of the CAVE. The camera looks onto the smaller mirrors in such a way that it views the large floor projection mirror located above the CAVE. The camera sees the interaction region inside the CAVE in the reflection of the large floor projection mirror. This setup allows the camera to view a large region within the CAVE.

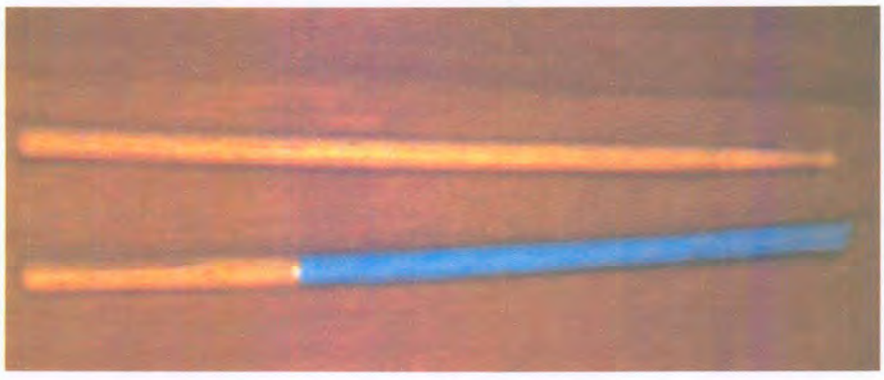


**Figure 4.8 Blue Drumsticks Tracked in the desktop Virtual Drums Installation**

The physical drumsticks are covered with blue paper and tracked by the inverse chroma-keying algorithm. The blue drumsticks used in the well lighted desktop environment are shown in figure 4.8. The environment is constrained so that there are no other blue objects in it other than the drumsticks. If there are any other blue objects their size must be significantly smaller than that of the drumsticks. Although the drumsticks are coloured blue, (covered with blue paper) this does not intrude upon the user or restrict the drummer's movement. In CyberStage the light drumstick in figure 4.9 is used in stead of a coloured drumstick.

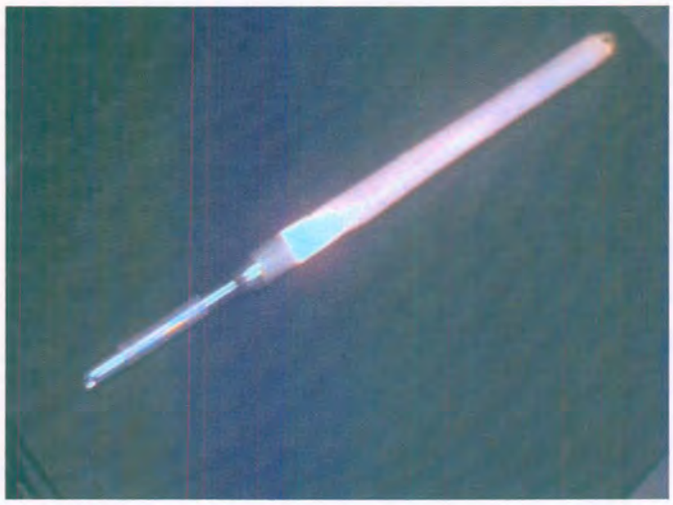


**Figure 4.9 Light Stick Used In CyberStage Virtual Drum Kit**

An intensity threshold is used in stead of the chroma-keying approach to track the light stick. The tracking algorithm works fine using this method when there is limited light in the CAVE. However, when a large bright object is projected onto the floor, such as a cymbal or snare cover, the tracker encounters problems and tracks these bright projections. This problem may be overcome by attaching a coloured filter to the torch and using the original chroma-keying algorithm to segment the coloured luminous torch. Another way of preventing the algorithm from tracking the light projections on the floor is to use a better computer vision algorithm which identifies a drumstick by its shape.

Tracking and interaction are achieved by using the different computer vision algorithms presented in section 3.2 These work well for the desktop environment and are stable in this environment. In the next section the computer vision algorithms are discussed.

### a) The Computer Vision and Tracking Algorithms

In the AVANGO version of the Virtual Drums a more elaborate computer vision algorithm is used. Figure 4.10 shows a flow chart of this algorithm.

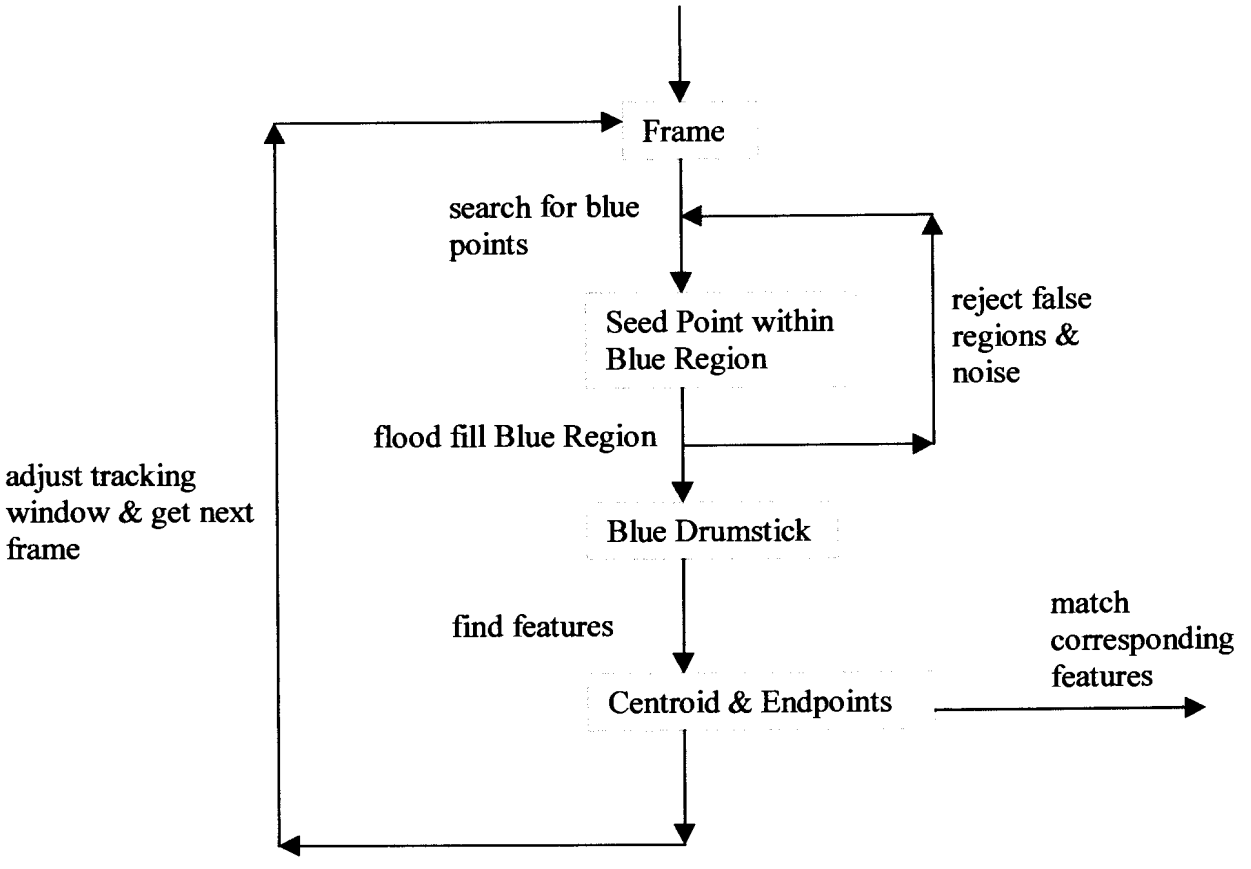



**Figure 4.10 The Computer Vision Algorithm Used in the Practical Implementation of the Virtual Drums Project**

Once a frame is captured, the algorithm searches through the region of the tracking window to find a drumstick. If it does not find a drumstick within the current tracking window, then the window is enlarged to the maximum size of its stereo view.

Figure 4.11 illustrates a picture of a catadioptric stereo image of a drummer with a blue drumstick. To find a blue drumstick inverse chroma-keying is used with a linear 2D grid search, which, much like a dragnet, does not test every pixel in the image, but samples the pixels in a 2D grid from left to right and top to bottom until it finds a blue pixel. A

blue pixel found by the search may belong to a drumstick or it may simply be noise. For this reason further testing is performed for each blue point found by the grid search. The search ends when the drumstick is found and segmented.

**Figure 4.11 Illustration of a Catadioptric Stereo Image of a Blue Drumstick**

When a blue pixel is found the SUSAN algorithm is applied. A circular mask with a radius of 3 pixels is centered on the blue pixel. If the blue pixel is random noise, then the USAN will be very small. This is because noise will only contribute to a small area of the USAN. In this way blue specks that crop up in the image are rejected and the search continues without doing extraneous processing on a noisy region. The objective of the search is to find an interior blue point within the drumstick. This is done with no extra processing cost since SUSAN has already determined whether or not the pixel lies within the interior of a region. It does this by testing the size of the USAN. An USAN belonging to an interior point of an object will have a very large value while USANs for edges, corners and noisy points will have lower values. This is because all the pixels around the nucleus of an interior point will be very similar in colour to the nucleus.

Figure 4.12 illustrates SUSAN applied to the right view of a catadioptric stereo image. In this figure the corner points are marked blue, while the interior points of a blue object are highlighted in green by the algorithm.

**Figure 4.12 SUSAN Applied to a Drumstick and a Piece of Paper**

The search along the grid continues until a blue interior pixel is found, i.e. until a blue pixel with a large USAN is found. This interior point of a blue object is then used as a seed for a fast flood fill algorithm [Fast Flood Fill]. The algorithm segments the blue region around the seed from the picture. This is very useful because if further searching must be done to find a second drumstick, the algorithm won't process the same drumstick twice. In the flood-fill algorithm, when a blue pixel is found it contributes to the sums of the image moments. The value of this pixel in memory is then changed to some other predefined colour (not blue) so that this point won't be re-filled. The flood-fill algorithm fills all the blue pixels around the seed. This flood-fill algorithm is very useful because:

- it segments a blue object from the frame (as it goes along),
- it calculates the area of the object and the sums of the image moments (sums of the blue pixels within the object), and
- it is useful for noise removal.

The reason the flood-fill algorithm is useful for noise removal is because it calculates the area of a blue region by finding the sums used for image moments. This area is a useful measure for identifying a drumstick and rejecting blue regions that are not drumsticks. If a blue region has too large or too small an area it is rejected. In the practical implementation, because the drumsticks are the largest blue objects that are seen in an image, any blue areas in the frame that are below a certain area-threshold, are counted as not being drumsticks while those regions with large enough areas are considered to be drumsticks. In this way a drumstick is identified and located while noise is segmented and rejected. This makes the computer vision algorithm more stable.

Once a drumstick is found, the sums taken by the flood-fill algorithm are used to calculate image moments. From the image moments meaningful features on the drumstick are found. Image moments could have been used as another measure for positively identifying a drumstick or rejecting other blue regions mistaken to be drumsticks. This is, however, unnecessary because of the constraints placed on the environment. The image moments determine the centroid of the drumstick, which is used for positioning the tracking window and for the calculation of the 3D position of the object. The image moments are also used to find the endpoints of the drumstick. Figure 4.13 illustrates the image moments of the drumstick.

**Figure 4.13 Image Moments of a Drumstick**

The computer vision algorithm is applied to both of the views to find the centroid and endpoints of the drumstick. The corresponding pair of centroids and endpoints are matched in the two stereo views. Matching is achieved by sorting these points according to their positions on the X-axis of the image.

The matched pairs of corresponding points are used to calculate the 3D position of these points on the object. The pair of centroids is used to calculate the 3D position of the object. The centroids found in the two views are also used to reposition the tracking windows within the two views. Figure 4.14 shows a picture of the visualization of the mathematical model in the implementation of the AVANGO Virtual Drums project. The entire process is repeated for each new frame captured by the camera.

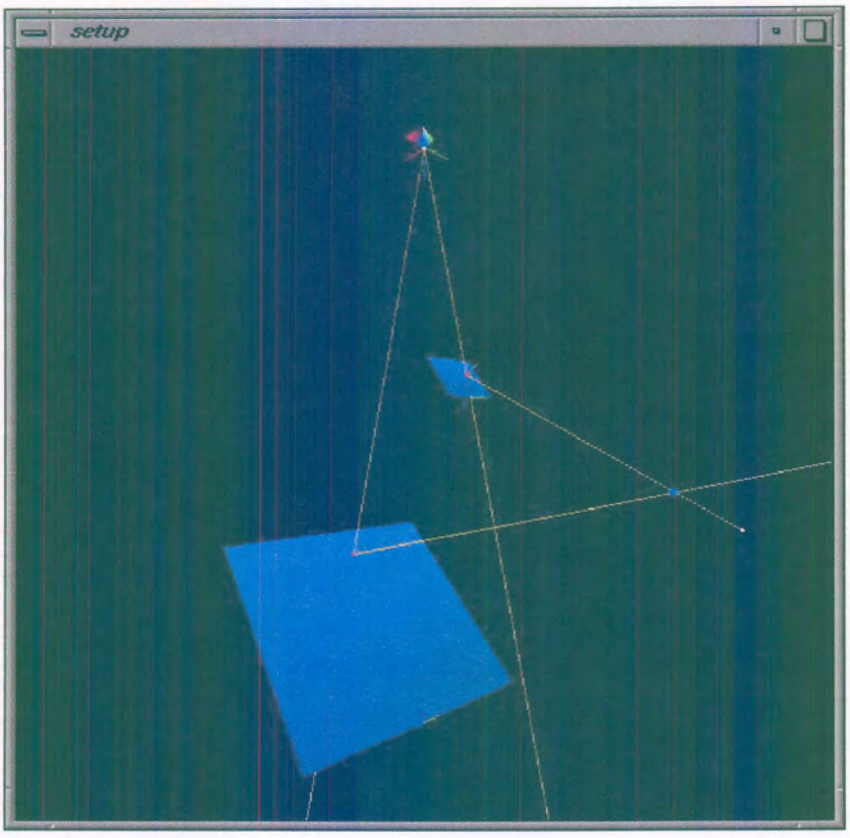



**Figure 4.14 Visualization of the Mathematical Model**

The algorithm is suitable for well-lit environments in which colour is clearly distinguishable. An adaptation of the algorithm is used in low light environments. In the adapted version the chroma-keying algorithm is replaced with an intensity thresholding algorithm and in stead of tracking blue drumsticks, a light stick is tracked. This algorithm is used in the CyberStage implementation of the Virtual Drums project.

The window tracker described in section 3.3.1 is implemented to track a drumstick from frame to frame. By using a window the region of an image that needs to be searched for the drumstick, is greatly reduced. Figure 4.15 is a picture of a tracked drumstick. The size of the window used is two-hundred by 150 pixels.

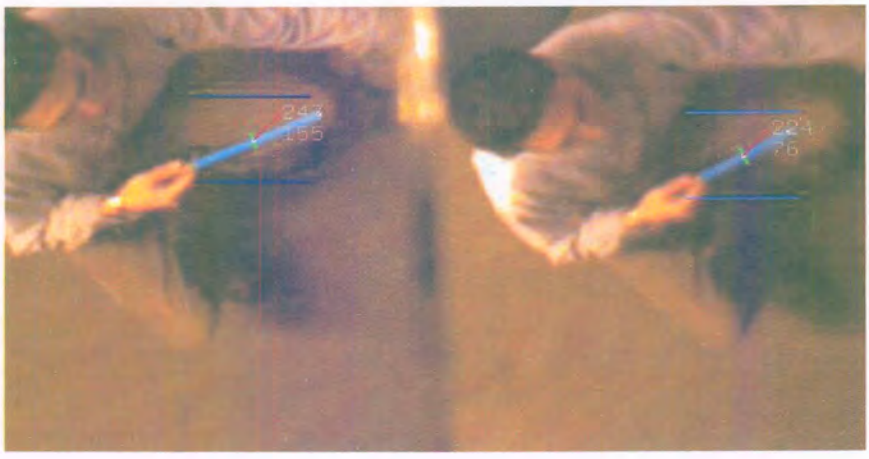


**Figure 4.15 Tracking a Blue Drumstick**

**(b) The 5DOF Extension**

While the centroids found in the two different views are matched and used to calculate the 3D position of the drumstick the endpoints are used for 5D determination.

The computer vision algorithm determines the image moments for a drumstick. These moments reflect the features of the rectangle having the same moments as the region of the drumstick. The endpoints of the drumstick are calculated by using the longest length, the centroid and the orientation of this rectangle as illustrated in figure 4.16.

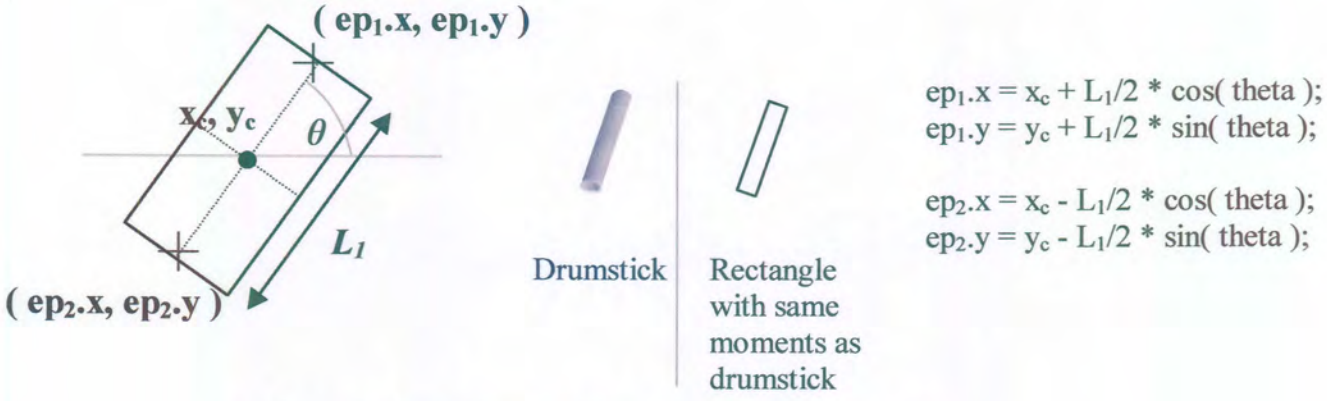



**Figure 4.16 Finding the Endpoints of a Drumstick**

The endpoints of the drumstick are found and matched in the two different stereo views. From the matched point pairs the 3D positions of the endpoints are calculated. These 3D positions are in turn used to calculate the orientation of the drumstick according to the method in section 3.1.8. The orientation (two rotation angles) and the 3D position of the drumstick together constitute the 5D information of the drumstick.

If epipolar geometry is used, then the matching algorithm used need only search the epipolar lines for matching points. If this is done then the CV algorithm is not repeated on the other stereo view which results in a significant speed up of the algorithm.

An important issue is to match the correct endpoints of the drumstick. In the Virtual Drums scenario the drumsticks never turn more than 90 degrees away from facing straight ahead and are not angled steeper than the angle from the drumstick to the lowest mirror. This ensures that the video camera always obtains images in both of its stereo views in which the tops of the drumsticks are seen. This results in an order on the endpoints in the stereo image which simplifies the computer vision algorithm. The order that exists is that, when looking at the stereo image from left to right, the front point of the drumstick will always be seen first, then the centroid followed by the rear endpoint. The endpoints are matched simply by sorting the points in the two different views and associating them according to their order. As long as the drumsticks are not angled too steeply the order on the endpoints will hold.

When more than one drumstick is used the matching process becomes more difficult. It is then necessary to distinguishing between the different drumsticks first and then match corresponding points. The procedure is as follows: Corresponding pairs of drumsticks are first found and matched. Then the corresponding points belonging to an associated pair of drumsticks are matched as before. Corresponding drumsticks are matched in the two different views by sorting the position of the centroids of the drumsticks. If the centroids are not differentiated by their order on the X-axis, then their position on the Y-axis is used. Having rectified catadioptric images and being able to match horizontal scan lines can be very useful for matching points on the drumstick.

### (c) Tracking Multiple Drumsticks

Extending the above computer vision algorithm to track multiple drumsticks is relatively simple. The first necessary change is an adjustment to the search component. The search algorithm continues searching for objects until either the maximum number of objects to be tracked are found or the entire view has been covered by the search. In the case of the Virtual Drums project once two drumsticks are located, the search stops.

The flood fill algorithm is an essential component of the CV algorithm for tracking multiple drumsticks. It is an "in-place" algorithm (it makes changes directly to the image and not to a copy of the image) which segments a blue region completely so that it is not examined by the search algorithm again. When the drumsticks cross over each other the flood-fill algorithm finds and segments both of the overlapping drumsticks as a single region. A test is then performed to distinguish between a region formed by a single drumstick and a region formed by two drumsticks which cross over one another to form a single region. The drumsticks must be of the same colour. This is so that when the blue drumsticks cross over, they appear to be a single blue region.

A simple test using image moments is used to classify a blue region as belonging to a single or two drumsticks. Consider that image moments determine the dimensions of a rectangle that approximate the moments of an image. Using these dimensions (length and breadth) the area of the approximating rectangle is found. The area of the blue region is also known from the sums calculated by the flood fill algorithm. The area of the rectangle with the same image moments as the blue region is compared to the area of the blue region. In the case where two drumsticks overlap and form a single blue region, the area of the rectangle will be far greater than the area of the overlapping drumsticks. If the blue region is formed by a single drumstick then the area of the rectangle will be almost equal to the area of the blue region. The only problem encountered with this approach is if the drumsticks almost or completely overlap. Fortunately, when playing a drum kit the drumsticks rarely move into such positions.

Figure 4.17 shows three different blue regions, one formed by two overlapping drumsticks and two regions formed by a single drumstick. Figure 4.17 illustrates that the area of the bounding rectangle for two drumsticks which cross over one another is far greater than the area of the blue region formed by a single drumstick. It also illustrates that the area of the bounding rectangle for a single drumstick is approximately equal to the area of the blue region formed by a single drumstick.

**Figure 4.17 Illustration of the Test for One or Two Drumsticks**

If the test indicates that a region belongs to a single drumstick then the endpoints of the drumstick are found using the approach described earlier and the algorithm continues searching for drumsticks. If, however, the test indicates that a region contains two drumsticks, the search stops and the endpoints of the two drumsticks are found. Finding the endpoints of overlapping drumsticks is not a solved problem and therefore tracking multiple drumsticks is not implemented in this version of the Virtual Drums project.

The Virtual Drums project illustrates the use of the Reflections method to implement multidimensional interaction that is both natural and non-intrusive and which works in both desktop and projection based systems.

## 4.3 NDEBELE PAINTING

In figure 4.18 Ndebele paintings are seen. The image on the left is a picture of a Ndebele lady painting on a wall. The picture on the right is that of a Ndebele house with colourful Ndebele paintings on the walls. Ndebele paintings are decorative paintings created by the Ndebele tribe in Southern Africa. These cultural artworks are distinguished by their striking, geometric and colourful designs. It is these paintings which were the inspiration for a virtual environment for facilitating the creation of Ndebele paintings. The project

was developed in a collaborative effort between GMD and the University of Pretoria [Lalioti et al].

**Figure 4.18 Ndebele Paintings**

### 4.3.1 The Existing Application

Illustrations of the Ndebele Painting application in VR are given in figure 4.19.

**Figure 4.19 Ndebele Painting in VR**

Ndebele Painting in VR is a virtual environment in which one can create Ndebele paintings. The project focus is on the process of wall painting and includes interaction metaphors that are natural to real wall painting. In the creation of this virtual painting world certain challenges which face design tools for painting in virtual environments are encountered. The project incorporates algorithms which assist the artist's creative process. Paintings are easily and effectively positioned so that they share with a snapping

algorithm. Another useful algorithm for the artist is an adaptation of an algorithm implemented by GMD, which simulates painting with a real paintbrush. With an extension of this algorithm the artist paints within different boundaries and lines without the paint spilling over into other regions.

The system has two modes, a selection and manipulation mode and contains a set of pre-scanned patterns. These modes and the different virtual tools allow the user to effortlessly browse through the different Ndebele patterns (tiles) and effectively control, resize and place them. A painting mode allows the user to choose different colours and assists in painting the tiles. Visual cues are used to indicate which mode the user is in. A very realistic interaction metaphor is used in this application, namely that of dipping a virtual paintbrush into a colour pot and then painting.

A choice of a variety of different interaction devices is available including the mouse, joystick and stylus. The stylus is used in the CAVE environment and proves to be the most natural and most appropriate of these devices. The system was developed using AVANGO in a desktop environment on an SGI Octane with a R12000 processor.

### 4.3.2 Ndebele Painting Meets Reflections

Reflections is used to implement non-intrusive interaction in the Ndebele Paintings application. The wires on many devices such as the stylus restrict the user's freedom of movement.

The Reflections approach is integrated into the Ndebele Painting application, in which in stead of using a wired stylus a small torch is tracked and used as the main input device. The torch has no button and so the button on the joystick is used. The large mirror in the CAVE and a small mirror placed on the camera are used in this implementation. The camera is placed outside the CAVE. The resolution of the camera used in this implementation is 640x480 pixels. A picture of the camera and torch used in this implementation are illustrated in figure 4.20.

**Figure 4.20 The Light Candle and Camera Used in Ndebele Painting**

The picture on the left in figure 4.21 illustrates Reflections in the CyberStage, while the image on the right is a picture of the floor projection mirror used in the implementation. The method of calculation used is very similar to the approach used in the first Virtual Drums application. Although a simplified calculation algorithm is used and is not truly accurate, it does give an indication of 3D position and illustrates the use of the Reflections method in a CAVE. The integration of the Reflections method into Ndebele Painting in VR is a simple and introductory integration of the method into VR applications.

**Figure 4.21 Reflections in Ndebele Paintings**

The computer vision algorithm searches the images from left to right and top to bottom for points of light. A point with an above threshold value for one of its component colours is found. This corresponds to a point on the torch. Because this torch is uncovered the brightness of the torch is significantly brighter than anything projected

onto the walls of the CyberStage. In this application the torch is successfully tracked in both views. The window tracker used in the other applications is also used in this implementation.

## 4.4 DEVELOPMENT PLATFORM

The core of the system was developed on an SGI Octane with the following specifications:

- A single 300 MHz MIPS R12000 processor (64-bit architecture)
- 2MB L2 cache
- 256MB main memory
- SGI EMXI graphics card
- Octane XIO Personal Video option board
- SGI personal video DigCam v1.2 desktop camera
- Irix 6.5 operating system

The GCC version 2.72 C++ compiler was used to code the various aspects of the system. The video capture component is implemented using the dmedia libraries for Iris 6.5.

## 4.5 IMPLEMENTING REFLECTIONS IN DIFFERENT VIRTUAL ENVIRONMENTS

This section contains a discussion on how to implement 3D interaction using Reflections in different environments. The three environments discussed are the desktop, the workbench and the CAVE. It includes a discussion on how to install the system in these different environments and the type of objects to use as the object to be tracked.

The physical installation of the system is very much dependant on the application. The mirrors and camera should be installed in such a way that occlusions are minimized and the object is clearly visible. The camera must also be placed close enough to the interaction region to obtain the required accuracy. The resolution of the camera must also

be suitable for the application as the position of the camera and mirrors and the resolution of the camera affect the accuracy and size of the interaction volume.

### 4.5.1 Desktop

In a desktop environment possibly the most optimal position for the camera and mirrors is against the ceiling above the user (for large regions of interaction). It is also possible to place the equipment to the side of the user. The distance the equipment is placed from the region of interaction depends upon the size of the desired region of interaction and the angle of view of the camera. The region of interaction must fit within the view frustum of the camera and mirrors. In a desktop environment lighting is in many cases sufficient and controllable. This allows either a luminous or coloured object to be used as the object to be tracked. Both the prototype of the Virtual Drums project and the AVANGO implementation illustrate Reflections in a desktop environment.

### 4.5.2 Responsive Workbench

Although the Reflections method is not implemented in a workbench environment, it is theoretically possible to integrate it in this environment in the following manner:
In the environment of the responsive workbench the camera is positioned above the workbench as in the desktop environment. It is also possible to install the equipment to the left or right of the workbench so that the interaction region is viewed from the side.

If infrared is used the projection mirrors which are already part of the environment may be used. This is very desirable because it allows the camera and mirrors to view the environment from beneath or in front of the user from positions which minimize occlusions. However, when using infrared a better computer vision algorithm will be necessary because colour is no longer a suitable means of identifying an object.

### 4.5.3 CyberStage

In a CAVE environment such as the CyberStage, if infrared is used, it should be possible to use any of the projection mirrors with the Reflections method. This makes the integration robust and supports an optimal positioning of the camera because the camera and mirrors will be able to view the environment from the front or from one of the sides or even from above. However, if infrared is not used, then the large floor projection mirror against the ceiling is used to optimally view the interaction region within the CyberStage. This is done by placing the camera against the ceiling so that it views both the large floor projection mirror and a smaller mirror which is placed closer to the camera. The camera views reflections of the interaction region within the CAVE in the two mirrors.

It is also possible to allow the camera to view two smaller mirrors which in turn view the large floor projection mirror. This allows the camera and mirrors to be placed in a variety of positions from which the camera views a large region within the CAVE. The Virtual Drums implementation in CyberStage illustrates this.

## 4.6 SUMMARY

In this chapter the design and implementation of a Reflections system are discussed in detail. The blueprint of the system is given in section 4.1. A Reflections system consists of both a physical installation process and a computational process. Practical aspects and steps are given in section 4.1.1 for installing the tangible components of a Reflections system. The requirements of the computational process are set out in section 4.1.2. These design issues are implemented in two different applications. The Virtual Drums project and Ndebele Painting. The Virtual Drums project makes use of Reflections to facilitate natural, non-intrusive and multidimensional interaction in a virtual drum kit. This project illustrates the use of Reflections in both a desktop and CAVE environment. There are two versions of the Virtual Drums project. The AVANGO implementation demonstrates the many features that are implemented. The desktop version of the AVANGO Virtual Drum

kit is capable of tracking coloured drumsticks, while the CyberStage implementation tracks a light stick. An extensive computer vision algorithm is used in both of the AVANGO Virtual Drums implementations. The CV makes practical use of several algorithms including a fast flood fill algorithm, chroma-keying (for well lighted environments), image moments and SUSAN to track the drumstick and to determine its endpoints. This implementation also tracks the 5D position of the drumstick. A discussion of tracking multiple objects is also included in this chapter.

The Ndebele Painting is a virtual painting environment in which Reflections is integrated for non-intrusive interaction. It illustrates the use of Reflections in CyberStage for three-dimensional interaction. The chapter draws to an end with a discussion on implementing Reflections in different virtual environments.

# Chapter 5

# Results

In this chapter the results of tests performed on the system are presented. Tests were performed on the hardware platform described in chapter 4 except where specified. The results reflect the accuracy of the 3D and 6D calculation algorithms. Results are also given for the stability of the system. An interesting finding about the relationship that occurs between the interaction volume, the accuracy of the system and the resolution of the camera is presented. Results on the speed of the system and the different computer vision algorithms are presented. The results of tests performed on the tracker are then discussed.

## 5.1 ACCURACY

The 3D calculation algorithms are the main focus of this thesis. Furthermore the precision of these calculations affect the accuracy of the 5D and 6D calculations. Therefore it is necessary to test and obtain results for these algorithms. While testing the accuracy of the 3D and 6D calculations it is also important to test the stability of these algorithms. All tests are performed at a frame rate of 30 frames per second.

### 5.1.1 Position

Preliminary tests were performed on the accuracy of the trigonometric approach for 3D calculation. The test environment includes a camera with a resolution of 720x486 pixels and a large mirror, size 40x20 cm. This mirror is placed 1 cm to the left of the interaction volume. The camera placed above the interaction region views simultaneously the mirror and the region in which measurements are taken. The size of the interaction volume used is approximately 30x30x25 $cm^3$. The front tip of a blue triangular object is tracked and matched in the two views of the camera.

| | distance | x | Y | z |
|---|---|---|---|---|
| Max | 6.02 | 4.00 | 4.50 | 4.50 |
| Min | 1.73 | 0.00 | 0.00 | 0.00 |
| Ave | 4.16 | 2.02 | 2.41 | 1.87 |
| Std | 1.10 | 1.26 | 1.28 | 1.35 |

**Table 5.1 Trigonometric 3D Calculation Accuracy Results**

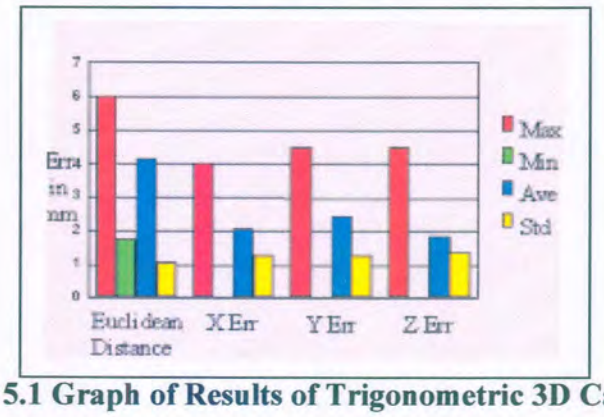



**Figure 5.1 Graph of Results of Trigonometric 3D Calculation**

Table 5.1 and figure 5.1 illustrate the results of this test. Thirty trials were performed where a trial included positioning the blue object and measuring its position in a Cartesian co-ordinate system. The position of the object calculated by the 3D algorithm is noted in these trials. These values are then used to determine the error in each of the component directions. The total error in position is also determined by using the Euclidean distance between the measured point and the point calculated by the 3D algorithm.

Both the maximum and minimum error are determined and the average error and standard deviation are calculated for the error in position. The largest overall error (Euclidean distance) is 6mm and the smallest error is 2mm. The average error is 4mm. This error may have been caused by:

- the matching of incorrect points in the stereo image,
- error of parallax while reading the values from the rulers,
- errors in the initial measurements, and
- distortions which occur near the edges of the images due to the curvature of the lens.

These results reveal some interesting and significant relationships between the accuracy, the resolution of the camera and the size of the interaction volume. The size of the interaction volume, the resolution of the camera and the overall accuracy in position are dependent upon each other. For example if a camera with the same resolution is used in a CAVE with an interaction volume of size 3x3x2.5 meters, then the accuracy will degrade from millimeter error to centimeter error. To improve or maintain the same accuracy it is necessary to use a camera with a greater resolution. As the interaction volume increases, with the camera resolution remaining constant, the accuracy degrades. Alternatively if the camera resolution increases for a fixed interaction volume then the accuracy will improve. Accuracy depends upon the distance of the camera from the region of interaction.

Tests on the algebraic approach for 3D calculation were also performed. The size of the interaction volume used is 30x20x20 $cm^3$. The same mirror used in the tests performed on the trigonometric approach is used. The results are given below:

| | X error | Y error | Z error | Euclidean |
|---|---|---|---|---|
| Max | 0.08 | 0.63 | 0.13 | 1.03 |
| Min | 12.71 | 8.35 | 3.87 | 14.61 |
| Ave | 2.41 | 3.47 | 2.16 | 5.23 |
| Std | 3.76 | 2.93 | 1.34 | 4.37 |

**Table 5.2 Algebraic 3D Calculation Accuracy Results**

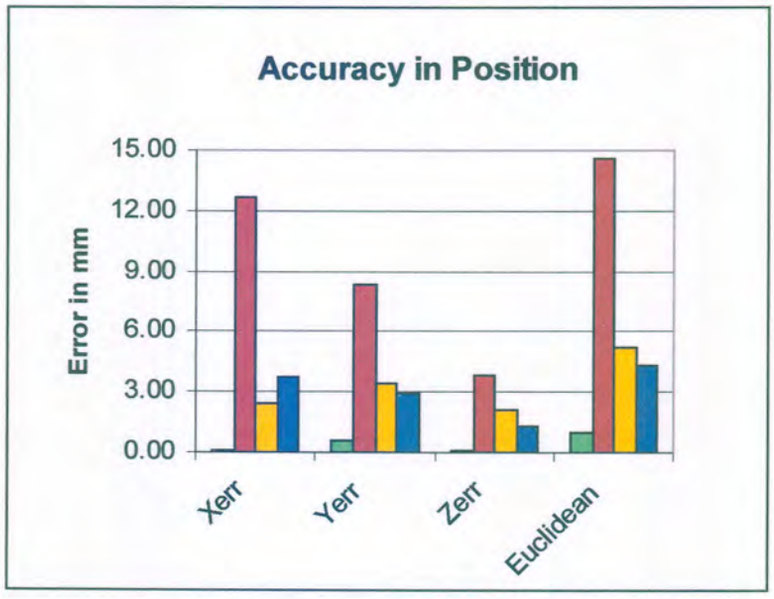


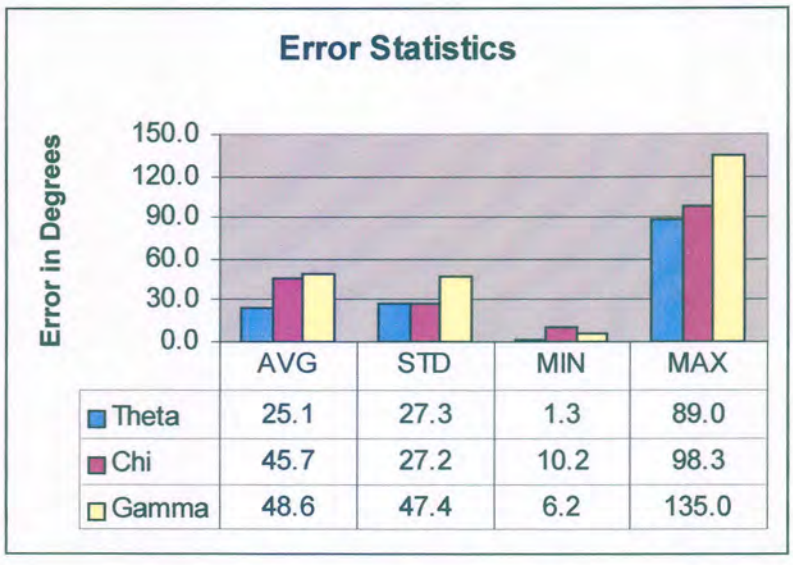

**Figure 5.2 Accuracy of 3D Calculation Using the Algebraic Approach**

By comparing the results in table 5.1 with table 5.2 it is clear that the trigonometric approach performed only slightly better than the algebraic approach. The average error in distance from the measured points for the algebraic approach is 5.23 mm. Figure 5.2 graphically represents the results in table 5.2.

### 5.1.2 Angular Accuracy

The accuracy of the 6D calculation is tested using the algebraic approach for 3D calculation and for the same environment. In this environment the four midpoints on the

edges of an A4 sized blue page are found using image moments and matched with their corresponding points in the stereo image. The matched four endpoints are used to determine the angular axes of the page. From these axes the orientation of the page is calculated. The results are listed below in table 5.3 with error statistics given in degrees.




| | AVG | STD | MIN | MAX |
|---|---|---|---|---|
| Theta | 25.1 | 27.3 | 1.3 | 89.0 |
| Chi | 45.7 | 27.2 | 10.2 | 98.3 |
| Gamma | 48.6 | 47.4 | 6.2 | 135.0 |

**Table 5.3 Angular Accuracy Results**

The error in these tests is significant. However, the error is a result of an irregularity which occurred in the image moments. When the blue paper is tilted, the moments become distorted because of the change in lighting and shadows that form on the page. This affects the accuracy and correctness of the other computer vision algorithms. This distortion in the image moments results in the incorrect calculation of the endpoints of the page. These displaced endpoints are matched. This leads to a matching of points that are not associated with each other; e.g. a true endpoint may be matched with a corner in the stereo pair. Accurate 3D calculation is invalid because of the mismatched endpoints. The 3D points calculated from the matched points are used to determine the axes of orientation of the page which are necessary for the calculation of the rotations of the page. The inaccuracies in the calculation of the 3D co-ordinates of these points affect the accuracy of the 6D calculations.

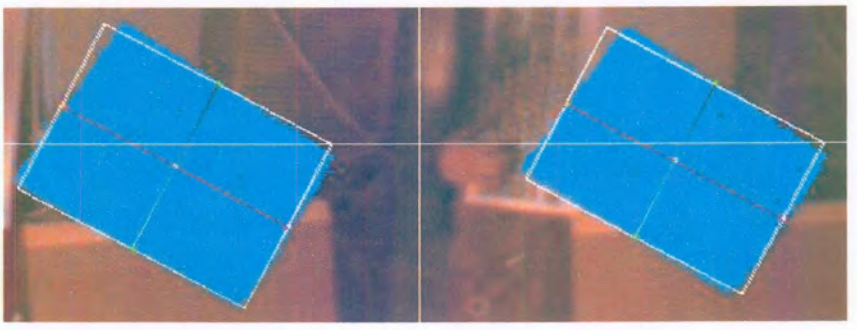


**Figure 5.3 Image Moments of A4 Page Used in Angular Tests**

Figure 5.3 shows the image moments for a page that is almost horizontal. Note that the image moments and endpoints are almost correct, although there is some slight distortion in these moments.


**Figure 5.4 Endpoints Found by Image Moments**

Figure 5.4 illustrates the endpoints found by the image moments for the blue page rotated about the different axes. Observe that the calculated endpoints do not correspond with the true endpoints of the blue page in this figure.

Another issue pertaining to angular measurements is the range of the angles of rotation. Angular rotations are limited by the midline between the camera and mirror, or between the two virtual cameras. In the case of using the A4 page the range of rotation around the Z-axis is 180 degrees since an 180-degree rotation of the page is identical to a zero

degree rotation of the page. This is because, to the computer vision algorithm a page rotated $\alpha$ degrees appears identical to a page rotated $\alpha + 180$ degrees and the angle reported is therefore the same. The ranges on the other angles depend on the shape and position of the object relative to the camera. The rotation of an object can be determined as long as the object is not tilted in such a way that it obstructs the camera's view of it's important features, i.e. the object must not become self occluding.

### 5.1.3 Stability

It is necessary to determine the stability of the algorithms. For this reason the jitter is measured using the algebraic approach for 3D calculation. The blue object is positioned and attached to a piece of wood with no movement for three seconds or 90 frames and readings are taken of the deviations in position. In each trial, if extreme results are obtained, these are extracted from the data set and noted. Figure 5.5 illustrates how many extreme erroneous results occurred in each trial. Those results remaining in the data set are used to determine the results listed in table 5.4.

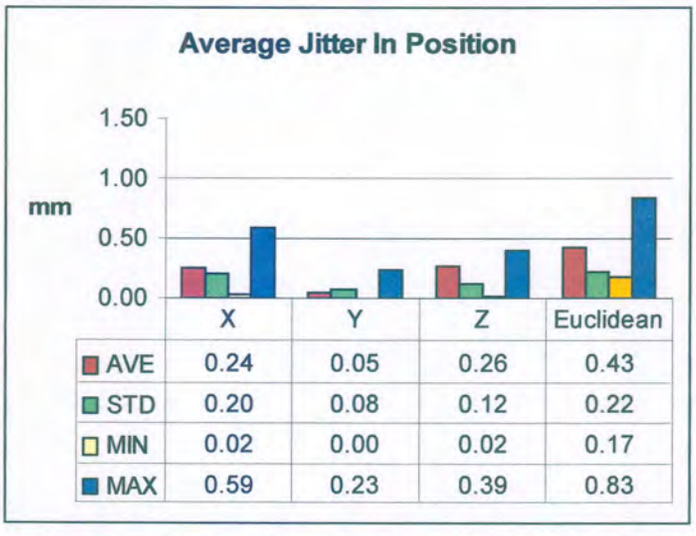



**Table 5.4 Stability Results for 3D Calculation Using the Algebraic Approach**

In each of the stability tests the camera captured images at 30 frames per second. The above stability results for position are very encouraging. The average fluctuation in

position is only 0.43 mm with a standard deviation of 0.22 mm. This indicates that the object's calculated position is very stable and does not on average vary more than 0.65 mm, although the occasional outlier does occur. These readings exclude extreme errors. The graph in figure 5.5 illustrates where extreme readings occured in the 10 different trials, in which for each trial, the object is held in place for 90 frames. Most of the extreme errors only occur during the 8th reading. This extreme jump in stability occurs due to the tracker loosing the object for a split second. The more minor errors possibly occurred due to mismatched points.

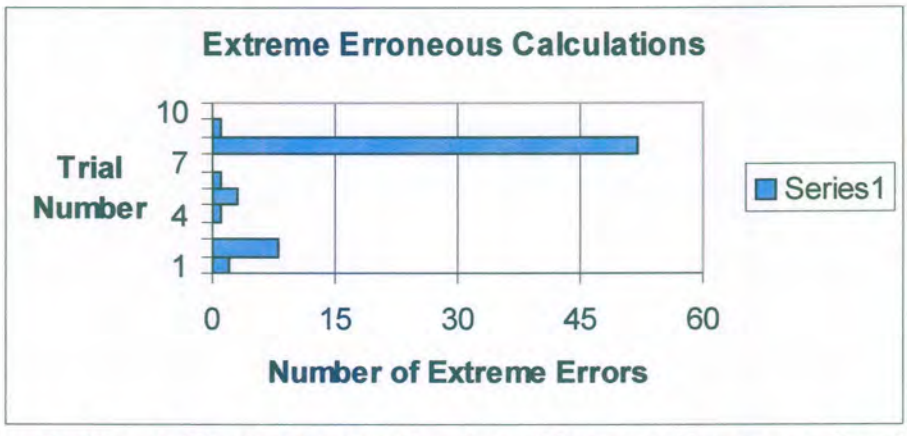
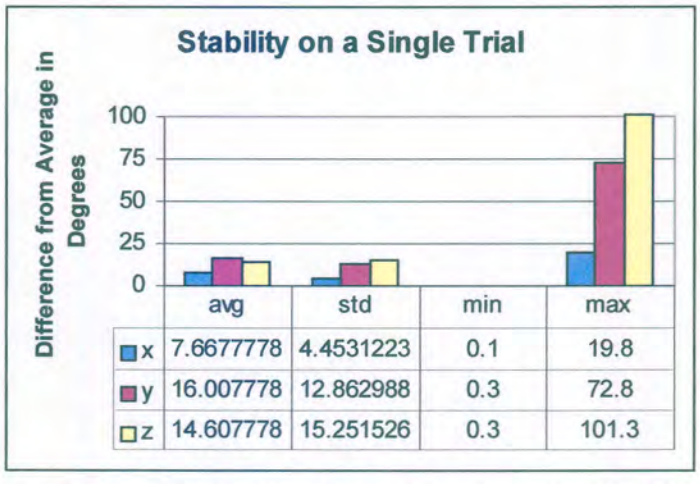



**Figure 5.5 Graph Illustrating the Number of Extreme Erroneous Results**

The stability of the angular measurements were also measured. The results for a single trial performed on the stability of the 6D calculations are given in table 5.5 below:




| | avg | std | min | max |
|---|---|---|---|---|
| x | 7.6677778 | 4.4531223 | 0.1 | 19.8 |
| y | 16.007778 | 12.862988 | 0.3 | 72.8 |
| z | 14.607778 | 15.251526 | 0.3 | 101.3 |

**Table 5.5 Results of a Single Trial for Angular Stability**

From this table it is evident that the angular stability is poor. This is once again a result of the anomalous error in the image moments. The moments for these readings fluctuate widely and affect the stability. A cause for this fluctuation in the moments is partly due to the lighting used in the environment. The test environment only contained fluorescent light sources. Fluorescent light flickers, causing the lighting on an object to change. This in turn causes variations in the output of the image moments, flood fill, chroma keying and SUSAN algorithms.

### 5.1.4 Interaction Volume

The interaction volume is dependent upon the distance the camera is from the region of interaction and the camera's angle of view. The interaction volume is also dependent upon the size of the resolution of the camera. There is no clear simple rule for how large or small a region of interaction can be covered. However, the resolution of the camera and desired level of accuracy play a role in determining the size of the interaction volume that is monitored. In the tests a camera with resolution of 720x486 pixels is used to obtain millimeter accuracy for an interaction volume with dimensions 30x30x25cm$^3$. Centimeter accuracy is anticipated when using a camera with the same resolution in an environment 3x3x2.5 meters$^3$. This means that the approach is suitable for small and medium sized environments.

## 5.2 SPEED

The image capture component of the system captures 30 frames per second (fps). Therefore the maximum speed of the tracker is 30 Hz. The image capture component is stable at this rate. All of the tests were performed at this speed. It is important for the algorithms to finish their processing within the period of time between the capture of one frame and the next. If this is not done then the stability and overall speed are compromised. Results below will show that the algorithms meet this necessary requirement and therefore the system is stable at 30 frames per second.

## 5.3 PERFORMANCE OF THE COMPUTER VISION ALGORITHMS

### 5.3.1 Efficiency of the Algorithms

The algorithms need to be fast enough to keep up with the frame rate. Since the camera captures 30 frames per second the entire process has only 33.3 ms to perform all the necessary calculations for a single frame.

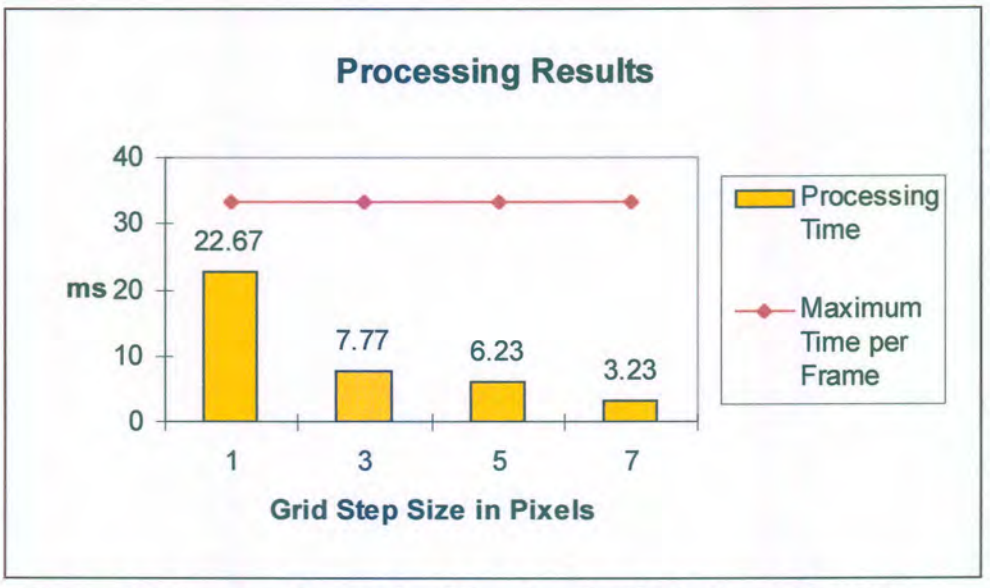



**Figure 5.6 Average Processing Time for the Reflections Algorithm**

| Step Size | 1 | 3 | 5 | 7 |
|---|---|---|---|---|
| Reflection Algorithm | 22.67 | 7.77 | 6.23 | 3.23 |

**Table 5.6 Average Processing Time for the Reflections Algorithm**

Figure 5.6 Illustrates the time it takes for the combined algorithms to process a single frame verse the maximum processing time per frame. In table 5.6 the time (in milliseconds) it takes for all the combined algorithms to process and search through the tracking windows, calculate the 3D and 6D information and track a single drumstick is given. In these tests the dimensions of the tracking windows are 200x150 pixels.

Execution times are given for performing the algorithms with different grid sizes (in pixels) for the dragnet search. Note that the algorithm still completes within the allotted time even for the smallest grid size. Therefore the algorithm is fast enough to process each frame and perform all the necessary calculations before the next frame arrives. This allows the entire process to function at the required frame rate and keeps the image capture component stable. Furthermore the algorithms are not optimized and still perform within the required time. Optimizations will make for even better results. The results for the different components of the computational process are given below:

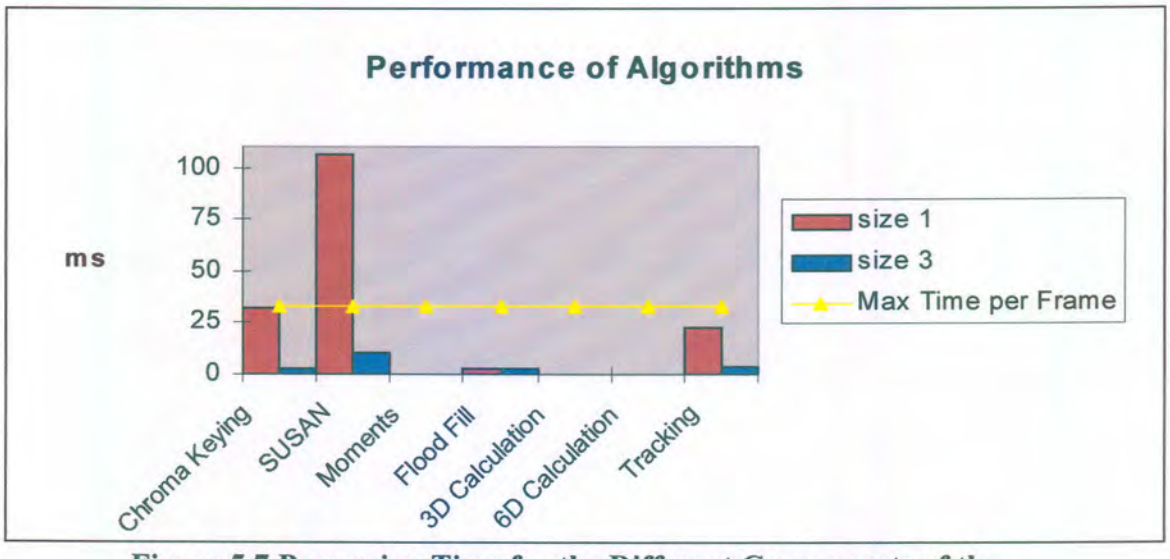



**Figure 5.7 Processing Time for the Different Components of the System**

In the graph in figure 5.7 the time the different components take to perform their processing, are illustrated. The results were obtained by tracking a single drum stick in two different tracking windows each of size 200x150 pixels in the stereo image. The results are given for grid step sizes of one and three. The chroma keying and SUSAN algorithms process this entire region and therefore use more time. The SUSAN algorithm also includes a chroma keying component, which allows it to only process blue pixels. This is why its performance is slower than that of the chroma keying algorithm.

Prior results performed on the chroma keying algorithm on an Intel Celeron 450MHz processor indicate that it processes 43.16 frames per second with a resolution of 720x486 pixels or 17.90 megapixels per second. It is therefore very suitable for real-time use [Van den Bergh].

SUSAN performs above the 33.33 ms available when a grid size of 1 is used. This is because it processes the entire region of both windows. In the combined algorithm SUSAN is only used by the search until a point within the interior of an object is found. Once an interior point is found the flood fill algorithm is then used. This is why, when a drumstick is found in a window, the performance improves. The time SUSAN and the chroma keying algorithms require to process a window to a large extent depends upon the size of the window and on the size of the blue region within the window. In table 5.7 below results for the chroma keying algorithm and the combined SUSAN and chroma keying algorithms are given for different window dimensions and grid sizes.

| ***Step Size*** | ***1*** | ***3*** | ***5*** | ***7*** |
|---|---|---|---|---|
| **Chroma Keying** | | | | |
| **view 1 298x449** | 69.67 | 5.33 | 2.63 | 1.9 |
| **view 2 298x428** | 47.87 | 7.17 | 2.2 | 1.1 |
| **Sum** | 47.87 | 12.5 | 4.83 | 3 |
| **view 1 200x150** | 14.1 | 1.5 | 0.43 | 0.33 |
| **view 2 200x150** | 17.87 | 1.63 | 0.5 | 0.23 |
| **Sum** | 31.97 | 3.13 | 0.93 | 0.57 |
| **SUSAN & Chroma Keying** | | | | |
| **view 1 298x449** | 100.07 | 12.33 | 3.3 | 1.63 |
| **view 2 298x428** | 57.23 | 13.3 | 3.7 | 1.73 |
| **Sum** | 157.3 | 25.63 | 7 | 3.37 |
| **view 1 200x150** | 67.1 | 5.57 | 2.5 | 1.47 |
| **view 2 200x150** | 39.5 | 4.53 | 2.23 | 0.97 |
| **Sum** | 106.6 | 10.1 | 4.73 | 2.43 |

**Table 5.7 Results of the Chroma Keying and SUSAN Algorithms**

Results illustrating the time it takes to process the left stereo window are given in the rows labeled view 1, while the processing time for the right stereo window are listed in rows labeled view 2. The rows named sum, give the combined time for both views.

The results of the image moments and flood fill algorithms go hand in hand. This is because the sums for the image moments are calculated within the flood fill algorithm. The results given in table 5.8 are for the performance of the fast flood fill algorithm on both a normal stereo view of the drumstick and for a view of the drumstick held up close to the camera. The flood fill algorithm takes longer to process the larger of the two blue regions (the drumstick held closer to the mirrors).

| | *view 1* | *view 2* | *sum* |
|---|---|---|---|
| **Image Moments** | 0.03 | 0.03 | 0.07 |
| **Flood Fill** | | | |
| **Normal drumstick** | 1.8 | 1.1 | 2.9 |
| **Large view of drumstick** | 4.1 | 5.1 | 9.2 |

**Table 5.8 Results of Image Moments and the Fast Flood Fill Algorithm**

Figure 5.8 illustrates the two different views of the drumstick used for the flood fill algorithm's results. The time required to calculate the image moments is almost negligible being only 0.07ms.

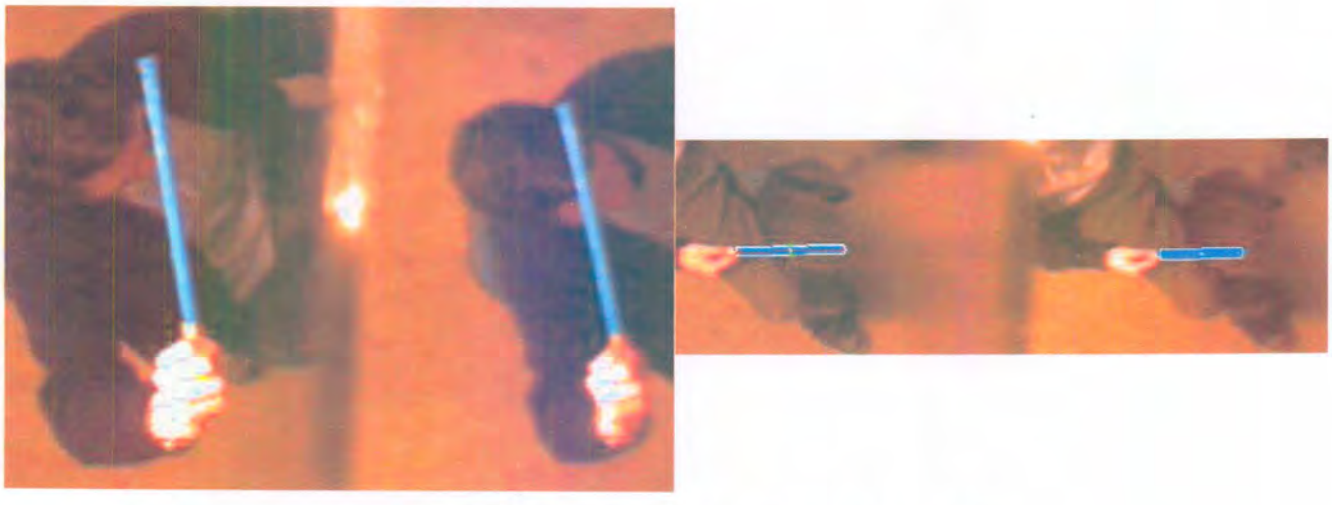


**Figure 5.8 Two Different Views of a Drumstick**

The time required to perform the 3D calculation is almost negligible. The 6D calculation requires more time because it calculates multiple 3D points and vectors. It also performs rotations on these vectors. The 6D calculation is still satisfactorily fast. These results are illustrated in table 5.9.

| | |
|---|---|
| **3D Calculation** | 0.03 |
| **6D Calculation** | 0.23 |

**Table 5.9 Results of the 3D and 6D Calculation Algorithms**

The time the algorithm requires to track a single drumstick for different grid sizes in the two stereo views using windows with dimensions 200x150 pixels are given in table 5.10. These measurements include the time it takes for the search algorithm, chroma keying and SUSAN algorithms to find the drumstick and the time it takes for the flood fill algorithm to segment the image and determine the area of the drumstick to identify it. The above times are safely within the maximum available time to process a frame.

| **Tracking (Chroma Keying, SUSAN and Flood Fill)** | | | | |
|---|---|---|---|---|
| **view 1 - window 200x150** | 11.73 | 2 | 1.77 | 1.5 |
| **view 2 - window 200x150** | 10.77 | 2.1 | 1.83 | 1.03 |
| **sum** | 22.5 | 4.1 | 3.6 | 2.53 |

**Table 5.10 Time to Track a Single Drumstick**

Table 5.11 illustrates the results of adding the different times it takes to perform the different algorithms. The tracking algorithm consists of the chroma keying, SUSAN and flood fill algorithms. The other algorithms are added to the tracking times. Once again the overall time required to perform all the calculations is well within the 33.33ms limit.

| Step Size | 1 | 3 | 5 | 7 |
|---|---|---|---|---|
| Tracking | 22.5 | 4.1 | 3.6 | 2.53 |
| Image Moments | 0.07 | 0.07 | 0.07 | 0.07 |
| 6D Calculation | 0.23 | 0.23 | 0.23 | 0.23 |
| Total | 22.8 | 4.4 | 3.9 | 2.83 |

**Table 5.11 Total of the Combined Times for the Different Algorithms**

The algorithms for tracking a single drumstick achieve what they set out to, in that they are fast enough to meet the real time requirements.

### 5.3.2 Effectiveness of the Algorithms

After having looked at the efficiency of the algorithms it is necessary to examine the effectiveness of the algorithms. The effectiveness of the algorithms is illustrated on the blue piece of paper used in the 3D calculation tests. Figure 5.9 shows the image with no computer vision techniques applied to it.

**Figure 5.9 Image Before Application of the Algorithms**

The chroma keying algorithm segments blue pixels successfully if there is appropriate lighting and if no shadows fall on the blue object. This is illustrated in figure 5.10. The algorithm highlights blue pixels in white. Prior results on the chroma keying algorithm show that it is comparable to hardware chroma keying devices when it comes to image quality [Van den Bergh]

**Figure 5.10 Image After Chroma Keying is Applied to it**

SUSAN finds the edges of an object effectively and is able to locate interior points as it is supposed to. Figure 5.11 illustrates the effects of SUSAN applied to the blue piece of paper. SUSAN marks the edges in blue and the interior points in green.

**Figure 5.11 SUSAN Applied to Image**

The fast flood fill algorithm satisfactorily fills a blue region this is illustrated in figure 5.12. The flood filled region is physically changed to yellow in this figure.

**Figure 5.12 Affects of the Flood Fill Algorithm**

Figure 5.13 illustrates the accuracy of the image moments on finding the endpoint of the piece of paper under even lighting. Image moments are effective in finding the centroid of the object.

**Figure 5.13 Calculation of Image Moments**

## 5.4 TRACKER PERFORMANCE

Results of tests performed on the window tracking algorithm that was implemented, are reported below. The algorithm tracks a single drumstick over a period of 10 seconds at 30 frames per second. The graphs and figures that are given below illustrate how many times the tracking window loses track of the drumstick. Figure 5.14 is a picture of the region of interaction in which the tests on tracking were performed.

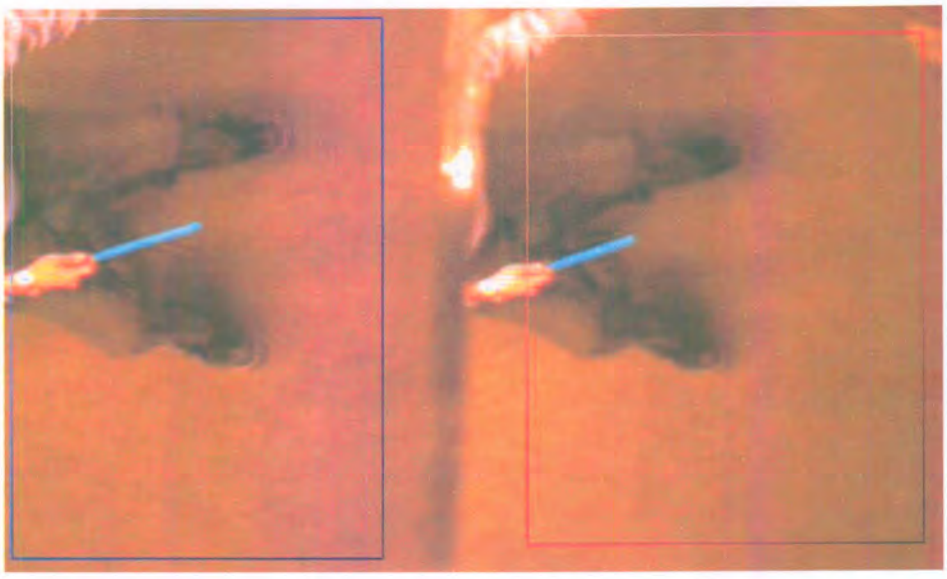


**Figure 5.14 Region of Interaction in which Tracking is Performed**

Tracking windows of size 200x150 pixels are used. These follow the centroid of the drumstick in both of the different views. When a frame is lost the window is resized to the maximum size of the view-region. The tracker is crucial because it reduces the search space to only a small portion of the entire image in which the drumsticks are to be found. This allows the computer vision components to meet the real time performance requirements.

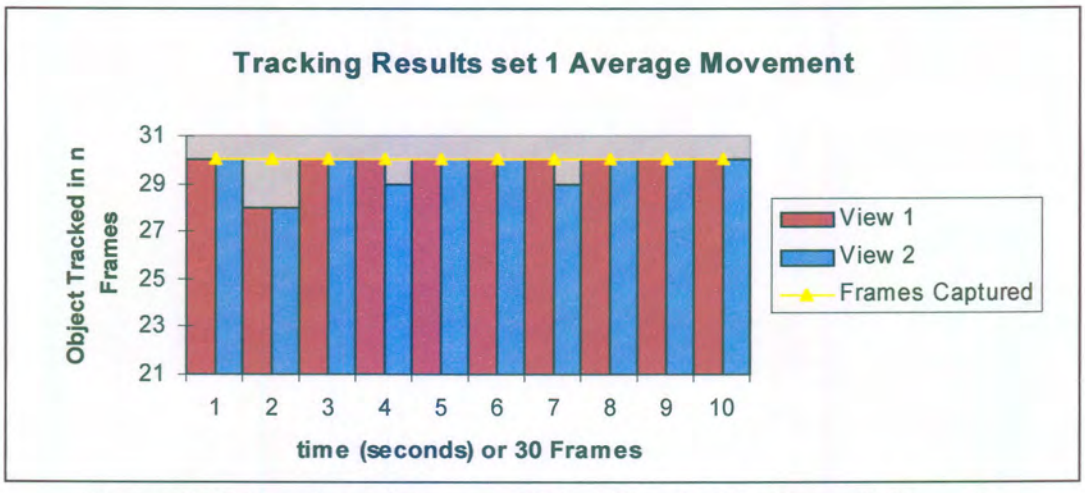



**Figure 5.15 Number of Frames Object Tracked in Over 10 Seconds**

Figure 5.15 illustrates the number of frames in which the tracking window keeps track of the drumstick in the different stereo views.

The movement of the object in the X, Y and Z directions is illustrated in figure 5.16.

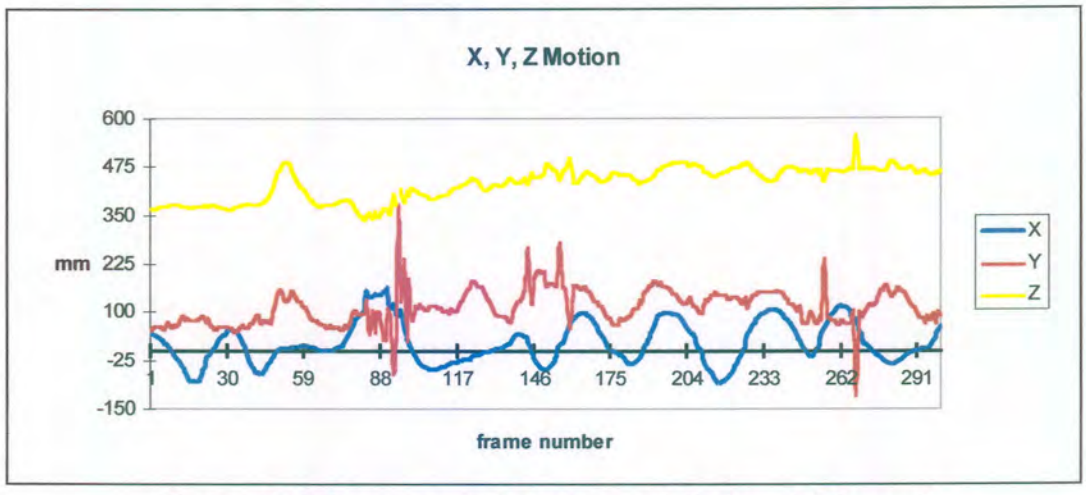
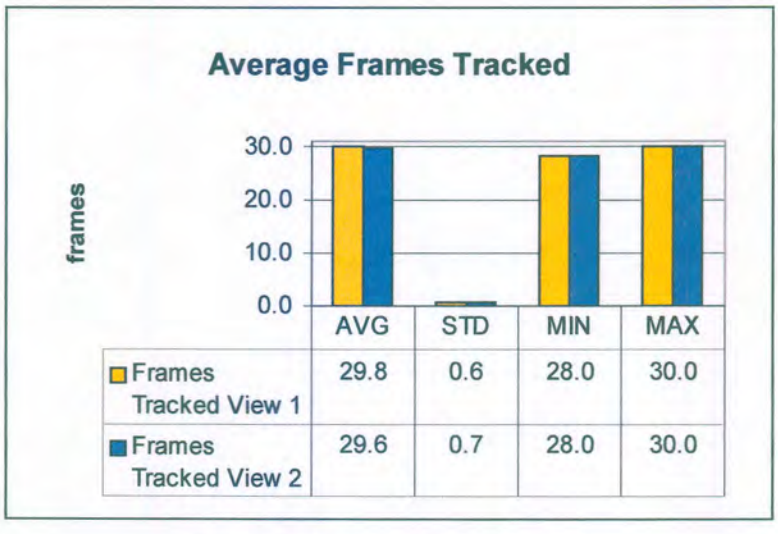



**Figure 5.16 Movement of the Drumstick Over 10 Seconds**

Table 5.12 summarizes the results of the tracker for the movements of the drumstick.




| | AVG | STD | MIN | MAX |
|---|---|---|---|---|
| Frames Tracked View 1 | 29.8 | 0.6 | 28.0 | 30.0 |
| Frames Tracked View 2 | 29.6 | 0.7 | 28.0 | 30.0 |

**Table 5.12 Tracking Results**

The drumstick is tracked in the left view (view 1) at an average of 29.8 frames per second over the 10 seconds. The tracker in the right view keeps track of the object on average 29.6 frames per second for the 10 seconds. The standard deviation is small and illustrates the effectiveness of the tracker. From this test it is clear that the tracker performs

exceptionally well. The movement in this test is moderate. The next set of results illustrate tracking of faster movement. This movement is illustrated in figure 5.17.

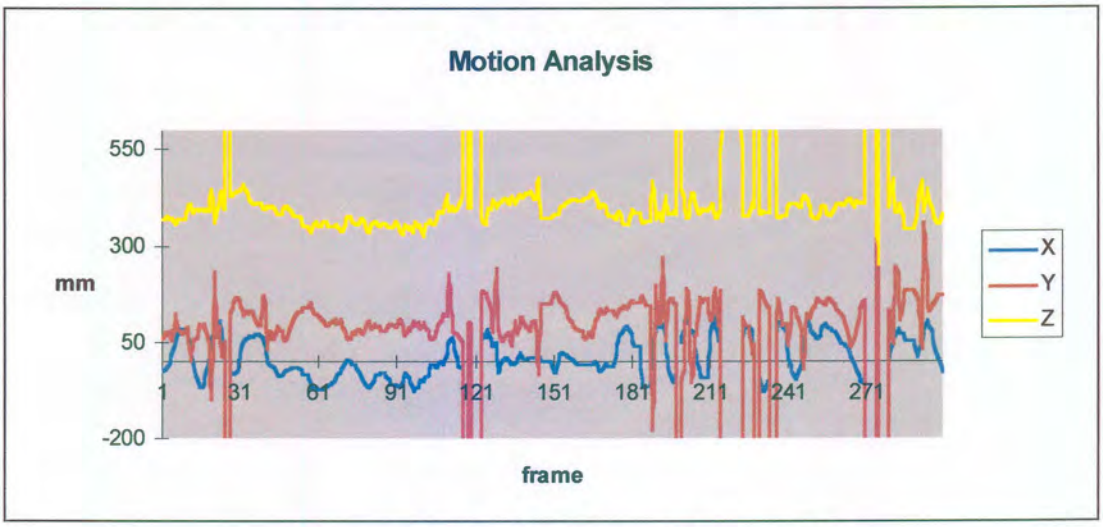



**Figure 5.17 Rapid Motion of Drumstick**

The graph in figure 5.18 shows the number of frames per second in which the window tracker followed the drumstick.

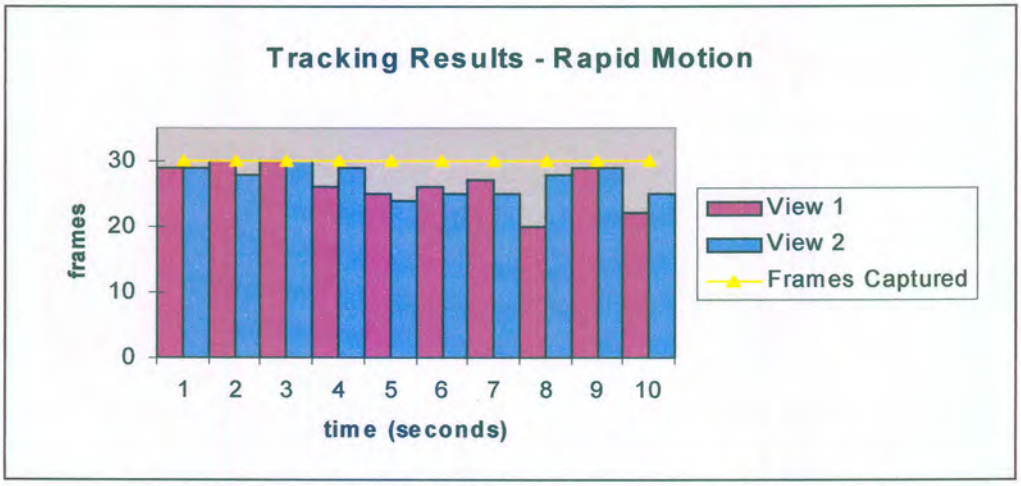



**Figure 5.18 Number of Frames for which the Drumstick is Tracked in the Different Views**

The results in table 5.13 show that the average number of frames in which the drumstick is successfully tracked is 26.4 frames per second out of a maximum of 30 frames per second in the first view and 27.2 frames per second in second view. For rapid movement this is still very good.

| | view 1 | view 2 |
|---|---|---|
| AVG | 26.4 | 27.2 |
| STD | 3.373 | 2.201 |
| MIN | 20 | 24 |
| MAX | 30 | 30 |

**Table 5.13 Tracking Results for Faster Movement**

Prior results on the window tracker indicate that at a frame rate of 25 frames per second and with a 3m horizontal field-of-view, objects are tracked at a speed of 11.7 m/s in the horizontal direction and at 8.8m/s in the vertical direction. Even at these speeds the window will not loose track of the object. Tracking however remains dependent on the resolution of the video camera and the speed of the computer [Van den Bergh].

Figure 5.19 illustrates the effects of a rapid moving object and interlaced images. The reason more frames are dropped for rapid movement is not because the tracker looses track of the drumsticks but is a result of image blur and interlaced images which affect the computer vision algorithms. The interlacing causes gaps (blank lines) to occur in the image of the drumstick. This causes the flood fill algorithm not to segment the entire drumstick and results in the tracker not finding the drumstick.

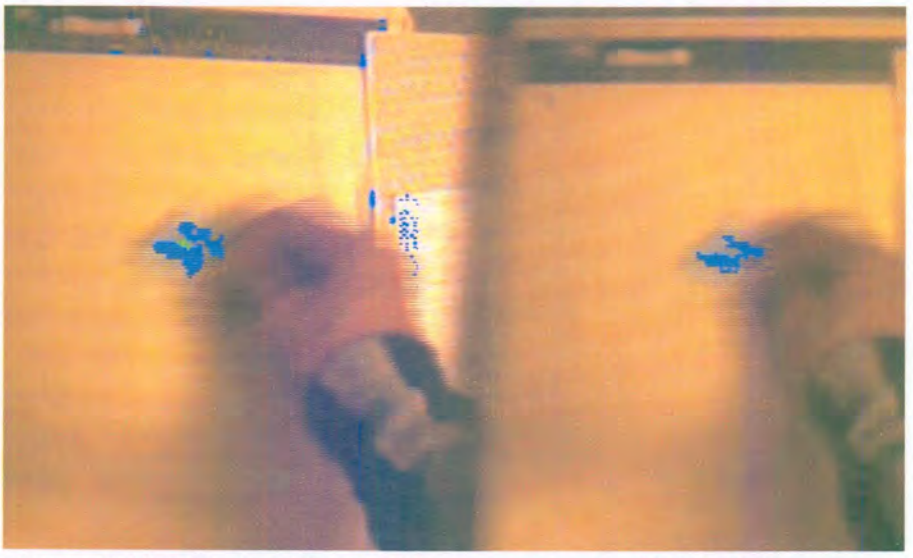


**Figure 5.19 Effects of Rapid Movement on the SUSAN Algorithm**

Another effect that has a negative influence on the tracker is when the object moves rapidly SUSAN counts the blurred interior points as edge points. This problem is overcome by reducing the edge threshold. The problem is illustrated in figure 5.19, in which SUSAN has coloured interior points as edge points.

## 5.5 SUMMARY

In this chapter the performance of the overall system is presented. Certain test results for the different aspects of the system are also covered. The accuracy of the 3D calculation algorithms are investigated and illustrate that the approach is suitable for 3D calculation. Results performed on the 6D calculation reveal that changes in light and shadows cast on an object influence the algorithms which determine the image moments for a shape. This causes inaccuracies in the calculation of the endpoints of an object and hence causes significant error in the 6D calculation. Stability tests performed indicate that the 3D calculation is very stable. Angular stability is susceptible to flickering light sources and is unstable.

The system performance is stable at 30 frames per second. The combined algorithms are able to complete the necessary processing within 33.33 ms (the time the algorithms have to process a single frame). The different algorithms are fast enough to meet the real time requirements of the system.

The algorithms are effective in that they perform the tasks they are required to. Tests performed on the window based tracker are presented in section 5.4. The tracker is able to track the movement of a drumstick from frame to frame and only occasionally looses track of the object. The tracker is stable and very effective but suffers from image blur and image interlacing (a property of the image capture device) when the drumstick moves too fast.

# Chapter 6

# Conclusions And Future Work

In this final chapter a discussion of the work which has been done is given and conclusions are drawn. Possibilities for future research and extensions are then presented. Some future extensions to the Virtual Drums project are proposed.

## 6.1 CONCLUSIONS

The central focus and aim of this work is the use of catadioptric stereo to implement multidimensional (3D, 5D or 6D) interaction for computer graphics. In the pursuit of this goal several overarching themes have been researched. These include virtual reality, interaction, computer vision and catadioptric stereo.

In the course of this work the following conclusions are realized and solutions developed:

(1) There is a need for non-intrusive interaction devices which support natural interaction for both 3D computer graphics and virtual reality. These devices need to support multidimensional interaction.

(2) The use of image capture (live video) provides a powerful means for achieving natural and non-intrusive interaction in real-time. However, a single camera can not accurately calculate depth and therefore can not on its own be used for accurate multidimensional interaction.

(3) Catadioptric stereo provides a means of accurately calculating 3D information using only a single camera and some mirrors. This approach has several advantages over conventional stereo.

(4) The orientation of an object is calculated by tracking and finding the 3D position of two or more points on the object. This means that catadioptric stereo can be used as a multidimensional input and tracking device.

(5) Computer vision is an essential component of a catadioptric stereo sensor. The computer vision algorithms must find and match associated points of an object in two stereo views. CV is crucial for implementing natural and non-intrusive interaction. The quality of the vision algorithms determines the limits of the interaction that can be implemented.

In this thesis an approach called Reflections is developed for implementing multidimensional interaction by using catadioptric stereo. Furthermore Reflections is suitable for natural and non-intrusive interaction. The merit of using this approach to implement such interaction in 3D computer graphics and virtual reality is seen in the

practical implementation of the Virtual Drums project and Ndebele Painting. The Virtual Drums project illustrates the use of the method for implementing natural and non-intrusive 5D interaction by allowing a user to play a virtual drum kit with real drum sticks. This is achieved by monitoring the position of a blue drumstick in 3D in a desktop environment. The project is also capable of tracking a light stick in low light environments. The Ndebele Painting application makes use of Reflections for natural non-intrusive 3D interaction in the CyberStage. Reflections may be implemented in a variety of different virtual environments ranging from a desktop monitor to a large projection based display like the CAVE.

## 6.2 FUTURE WORK

There are a several possibilities for future work in the topics covered in this research. Several of these are discussed below. The use of non-planar (hemi-spherical) mirrors for monitoring larger interaction volumes could be considered. Implementing 3D reconstruction using the Reflections method is also a possible area of exploration.
A further important extension to the method is to improve the computer vision algorithms to handle complete occlusions. The algorithms can also be extended for use with infrared. An extension to the computer vision algorithms that may prove beneficial to the approach, is the use of contour and silhouette tracking. If this is implemented the tracked object need not be a specific colour. This will also be advantageous for the method if infrared is implemented. Tracking the contour of objects can allow the approach to track more degrees of freedom.

The latency of the system needs to be closely investigated because of the adverse effects latency has on presence in virtual reality. Predictive tracking presents a means for overcoming latency and for this reason should be integrated into the method. Implementing predictive tracking could also be used to enhance the speed and stability of the tracker. The system can be optimized to perform optimally on the platform it is currently running on. Implementing the system on a faster processor and making use of a digital or video camera with a higher frame rate will also improve the speed of the

tracker, e.g. the speed of the tracker may be increased to 1000Hz by using the Artificial Retina Chip [Freeman et al]. The use of rectified catadioptric stereo requires further investigation as it presents yet another means of improving the speed of the tracker and achieving greater stability and accuracy, especially for point matching.

Motion tracking can be used to implement more natural interaction based on a user's movements. Tracking hand movements and gestures using Reflections and vector keying is an interesting possibility for future research in applications such as hand-based object manipulation.

The final desktop version of the Virtual Drums project is made to run the graphics on a computer and the Reflections application which tracks the drum sticks on a separate computer. However in the lab the Onyx machine crashed and tests across a network could not be performed. The two applications had to be run on the Octane together. The system experiences latency problems and although the tracker performs at 30Hz the graphics does not seem to reflect this. Furthermore playing the drums while looking at a 2D display is not as easy as it should be. These problems need to be addressed. Further issues that need to be looked into in the future include dealing with jitter and creating a good calibration algorithm for the approach. This will allow the calculated 3D co-ordinates to be better transformed into virtual world 3D co-ordinates.

The Virtual Drums project may be improved by implementing force feedback to simulate the reverberation experienced when a stick strikes a cover or cymbal. Another extension to the virtual drum kit is to implement an audio haptic display in which contacts with the drumsticks and drums are rendered with both haptics and sound [Pai].

Using the approach for head tracking presents a practical application of the method for projection based virtual environments. The use of the approach in new and diverse applications should be contemplated. Although there are several possibilities to extend and improve the system, this work provides a thorough coverage of relevant topics.

# Bibliography

[Abdel-Hamid & Yang] Gamal Abdel-Hamid and Yee-Hong Yang, "*Electrostatic Field-Based Multiscale Corner Detection: A Physics-Motivated Approach*", technical report, December 14, 1994.

[Angel] Edward Angel, "*Interactive Computer Graphics – A Top-Down Approach with OpenGL, second edition*", Addison-Wesley, pp. 17-18, United States of America, 2000.

[Baker & Nayar] Simon Baker and Shree K. Nayar, "*A Theory of Catadioptric Image Formation*", Proceedings of the 6th International Conference on Computer Vision, pp. 35-42, Bombay, January 1998.

[Bilinghurst] Mark Bilinghurst, "*Put That Where? Voice and Gesture at the Graphics Interface*", Computer Graphics, pp. 60-63, November 1998.

[Blake & Isard] Andrew Blake and Michael Isard, "3D position, attitude and shape input using video tracking of hands and lips", Computer Graphics (Proc. Sigraph 94), ACM Press, New York, 1994.

[Blake et al] E Blake, J Casanueva and D Nunez, "*Presence as a Means for Understanding User Behavior in Virtual Environments*", South African Computer Journal, No. 26, pp. 247, South Africa, 2000.

[Buxton & Fitzmaurice] Bill Buxton and George W. Fitzmaurice, "*HMDs Caves & Chameleon: A Human-Centric Analysis of Interaction in Virtual Space*", Computer Graphics, pp. 69-73, November 1998.

[Brooks] Fredrick P. Brooks, Jr., "*What's Real About Virtual Reality?*", IEEE Computer Graphics and Applications, pp. 16-17, November/December 1999.

[Casanueva & Blake] Bill J. Casanueva and E. Blake, "*Small Group Collaboration and Presence in a Virtual Environment*", South African Computer Journal, No. 26, pp. 163, 164, South Africa, 2000.

[Creek & Moccia] Patricia Creek and Don Moccia, "*Digital Media Programming Guide*", Silicon Graphics, 1996.

[Corbett] Daniel R. Corbett, "*Multiple Object Tracking in Real-Time*", Thesis submitted for Bachelor of Engineering, Department of Computer Science and Electrical Engineering University of Queensland, October 20, 2000.

[Davis & Bobick] James W. Davis and Aaron F. Bobick, "*A Robust Human-Silhouette Extraction*

*Technique for Interactive Virtual Environments*", CAPTECH'98, LNAI 1537, pp. 12-24, 1998.

[Delamarre & Faugeras] Quentin Delamarre and Olivier Faugeras, "*Finding pose of hand in video images: a stereo-based approach*", Proceedings of FG'98, April 14-16, 1998, Nara, Japan.

[Ellis & Gullick] Robert Ellis, Denny Gulick, "*Calculus with Analytic Geometry*", Fifth Edition, Saunders College Publishing, pp. 721-724, United States of America, 1978.

[Foley et al] Foley, Van Dam, Feiner, Hughes, Phillips, "*Introduction to Computer Graphics*", Addison-Wesley, pp. 297-319, United States of America, 1997.

[Forsberg et al] Andrew S. Forsberg, Joseph J. LaViola, Jr., Lee Markosian, and Robert C. Zeleznik, "*Seamless Interaction in Virtual Reality*", IEEE Computer Graphics and Applications, pp. 6-9, November/December 1997.

[Foxlin et al] Eric Foxlin, Michael Harrington and George Pfeifer, "*Constellation™ : A Wide-Range Wireless Motion Tracking System for Augmented Reality and Virtual Set Applications*", Proceedings of SIGGRAPH 98 (Orlando, Florida, July 19-24, 1998) Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH.

[Freeman et al] William T. Freeman, David B. Anderson, Paul A. Beardsley, Chris N. Dodge, Michal Roth, Craig D. Weissman, and William S. Yerazunis, Hiroshi Kage, Kauo Kyuma, Yasunari Miyake, Ken-ichi Tanaka, "*Computer Vision for Interactive Computer Graphics*", IEEE Computer Graphics I/O Devices, pp. 42, May/June 1998.

[Garcia & Tziritas] Christophe Garcia and Georgios Tziritas, "*Face Detection Using Quantized Skin Color Regions Mergin and Wavelet Packet Analysis*", IEEE Transactions on Multimedia, Vol. 1, No. 3, pp. 264-277, September 1999.

[Gluckman & Nayar] Joshua Gluckman and Shree K. Nayar. "*Planar Catadioptric Stereo: Geometry and Calibration*",In Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press, Fort Collins, Colorado, volume I, pp. 22-28, June 1999.

[Gluckman & Nayar, 2000] Joshua Gluckman and Shree K. Nayar, "*Rectified Catadioptric Stereo Sensors*", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, June 2000.

[Grigorishin et al] Tanya Grigorishin, Gamal Abdel-Hamid and Yee-Hong Yang, "*Skeletonization: An Electrostatic Field-Based Approach*", Pattern Analysis and Applications, Vol. 1, pp. 163-177, 1998.

[Koschan] Andreas Koschan, "*A Comparative Study On Color Edge Detection*", 2nd Asian Conference on Computer Vision ACCV'95, Vol. III, pp. 574-578, Singapore, 5-8 December 1995.

[Lalioti et al] Vali Lalioti, Andries Malan, James Pun, Juergen Wind, "*Ndebele Painting in VR*", IEEE Computer Graphics and Applications, Vol 20, No 6, pp54-65, November/December 2000.

[Lalioti et al, 1998] V. Lalioti, F. Hasnbrink, H. Tramberend, M. Goebel, "*Immersive Telepresence in Responsive Virtual Environments*", 9th NEC Research Symposium, Yokohama, Japan, 1998.

[Lane] James Lane, "*Capturing the Third Dimension*", pp. 1-11. South African Computer Journal, Electronic Papers, No. 26, pp. 1-11, South Africa, 2000.

[Lane & Lalioti] J. Lane and V. Lalioti, "*Reflective Interaction in Virtual Environments*", EUROGRAPHICS 2001, Manchester, United Kingdom, 4-7 September 2001.

[Laviola] Joseph J. Laviola, Jr., "*Interaction in Virtual Reality: Categories and Metaphors*", IEEE Computer Graphics and Applications, pp. 33-34, November/December 2000.

[Lastra] Anselmo A. Lastra, "*Technology for Virtual Reality*", Department of Computer Science, University of North Carolina Chapel Hill, NC 27599-3175, pp. 2-18 - 2-23, May 1994.

[Lay] David C. Lay, "*Linear Algebra and Its Applications*", Addison-Wesley, United States of America, 1994.

[Leibe et al] Bastian Leibe, Thad Starner, William Ribarsky, Zachary Wartell, David Krum, Justin Weeks, Bradley Singletary, and Larry Hodges, "*Toward Spontaneous Interaction with the Perceptive Workbench*", IEEE Computer Graphics and Applications, pp. 54-65, November/December 2000.

[Pai] Dinesh K. Pai, "*Interactive Simulation for Multimodal Virtual Environments*", Eurographics, Tutorial T5, Manchester, 2001.

[Paley] W.Bradford Paley, "*Designing Special-Purpose Input Devices*", Computer Graphics, pp. 55-59, November 1998.

[Parker et al] J. Parker, C. Jennings, D. Molaro, "*A Force-Based Thinning Strategy With Sub-Pixel Precision*", In Proceeding of Vision Interface 94, Banff, AB, 18-20 May 1994.

[Peterson] Barry Peterson, "*The Influence of Whole-Body Interaction on Wayfinding in Virtual Reality*", Thesis for Master of Science of Engineering, University of Washington, pp. 17-25, 1998.

[Segen & Kumar] Jakub Segen and Senthil Kumar, "*Simplifying human-computer interaction by using hand gestures. Look Ma No Mouse*", Communications of the ACM, vol. 43, No. 7, pp. 103-109, July 2000.

[Shi & Tomasi] Jianbo Shi and Carlo Tomasi, "*Good Features to Track*", IEEE Conference on Computer Vision and Pattern Recognition (CVPR94), Seattle, June 1994.

[Smith et al] Joshua Smith, Tom White, Christopher Dodge, Joseph Paradiso, Neil Gershenfeld, "*Electric Field Sensing For Graphical Interfaces*", IEEE Computer Graphics I/O Devices, pp. 54, May/June 1998.

[Smith & Brady] S.M. Smith and J.M. Brady, "*SUSAN - A New Approach to Low Level Image Processing*", Technical Report TR95SMS1c, Defence Research Agency, Franborough, Hapshire, GU14 LTD, UK, 1995.

[Strauss] Lotz Strauss, "*Introductory Physics*", Published by Lotz Strauss, pp. 2-4, University of Pretoria, 1995.

[Svoboda & Pajdla] Tomas Svoboda, Tomas Pajdla, "*Panoramic cameras for 3D Computation*", Czeck Pattern Recognition Workshop 2000, Tomas Svoboda (Ed.), Perslak, Czech Republic, pp. 1-8, February 2-4, 2000.

[Tramberend] H Tramberend, "*AVANGO: A Distributed Virtual Reality Framework*", In Proceedings of the IEEE Virtual Reality '99, 1999.

[Turban] Efraim Turban, "*Expert Systems and Applied Artificial Intelligence*", Prentice Hall, pp. 337-365, New Jersey, 1992.

[UnderKoffler et al] John UnderKoffler, Brygg Ullmer, and Hiroshi Ishii, "*Emancipated Pixels: Real-World Graphics In The Luminous Room*", MIT Media Laboratory, USA.

[van Dam et al] Andries van Dam, Andrew S. Forsberg, David H. Laidlaw, Joseph J. LaViola, Jr., Rosemary M. Simpson, "*Immersive VR for Scientific Visualization: A Progress Report*", IEEE Computer Graphics and Applications, pp. 26, November/December 2000.

[Van den Bergh] Frans Van den Bergh, "*A Device-free Locator using Computer Vision Techniques*", University of Pretoria, South Africa, 1999.

[Wartell et al] Zachary Wartell, Larry F. Hodges, William Ribarsky, "*Balancing Fusion, Image Depth and Distortion in Stereoscopic Head-Tracked Displays*", SIGGRAPH 99 Conference Proceedings, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, p351-357, August 1999.

[Yang & Gillies] G.Z. Yang and D.F. Gillies, "*Matching Relational Structures*", Computer Vision, Department of Computing, Imperial Colleage, Chapter 18, pp. 1-4, http://www.doc.ic.ac.uk/~gzy

[Young] Hugh D. Young, "*University Physics*", Eighth Edition, Addison-Wesley, pp. 945-1006, United States, 1992.

[Zhang & Gimel'farb] J. Q. Zhang and G. L. Gimel'farb, "On detecting points-of-interest for relative orientation of stereo images", Proc. of Image and Vision Computing New Zealand'99 Intern. Conf., 30-31 August 1999, pp.61-66, Christchurch, New Zealand (Eds.: D.Pairman, H.North). Landcare Research: Christchurch, 1999.

[Zhai] Shumin Zhai, "*User Performance in Relation to 3D Input Device Design*", Computer Graphics, pp. 50-54, November 1998.

[Spatial Audio] Spatial Audio Work In the Multimedia Computing Group http://www.cc.gatech.edu/gvu/multimedia/spatsound/spatsound.html/

[Fast Flood Fill] http:// graphics. lcs. mit. edu/ classes/ 6.837/ F00/ Lecture04/ Slide14.html [9/ 19/ 2000 4: 09: 48 PM]