

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

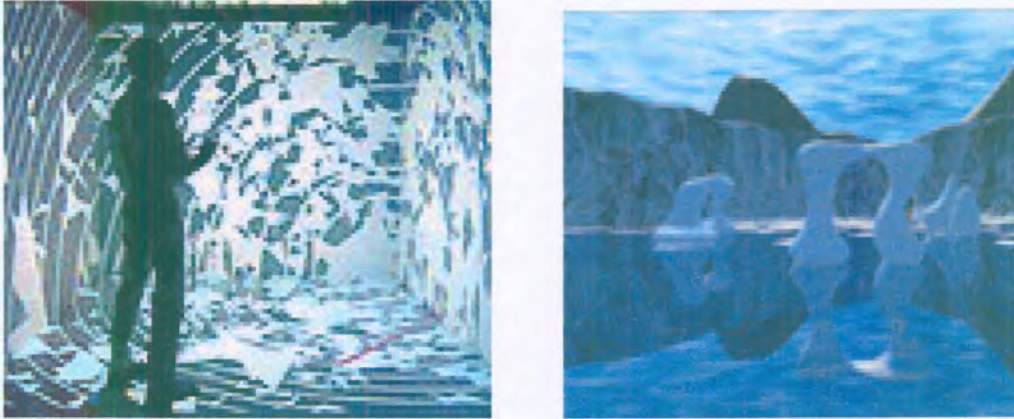


Figure 2.1 Pictures of Virtual Worlds
(Courtesy of GMD)

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.

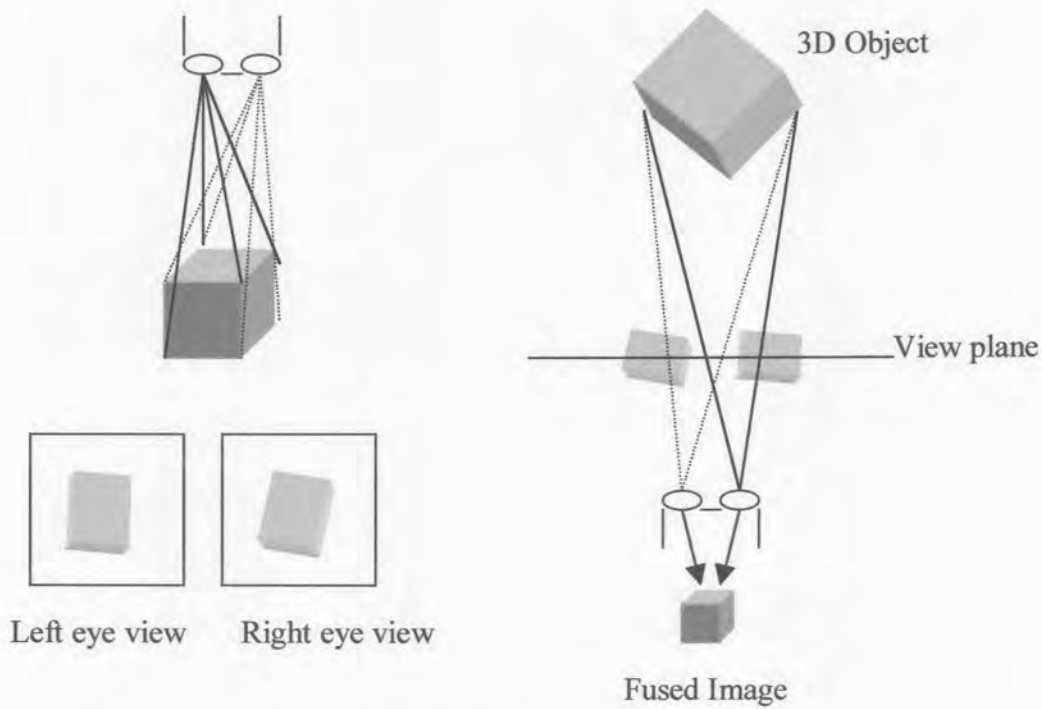


Figure 2.2 Binocular Disparity.

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 [Laloti 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.

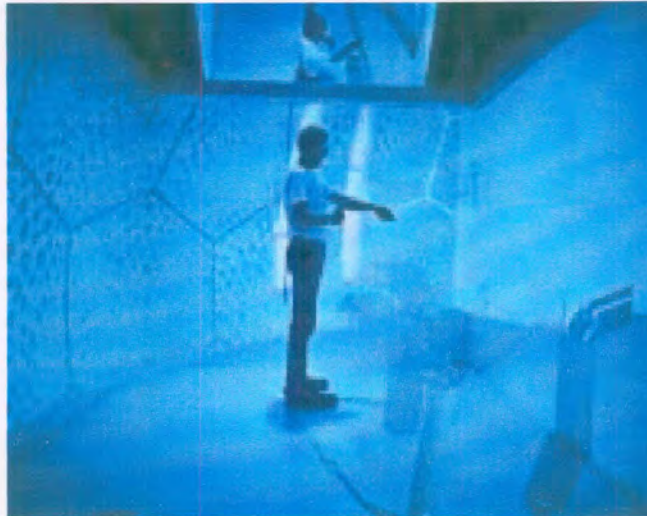


Figure 2.3 A Picture of CyberStage (Courtesy of GMD)

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 ConstellationTM [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.

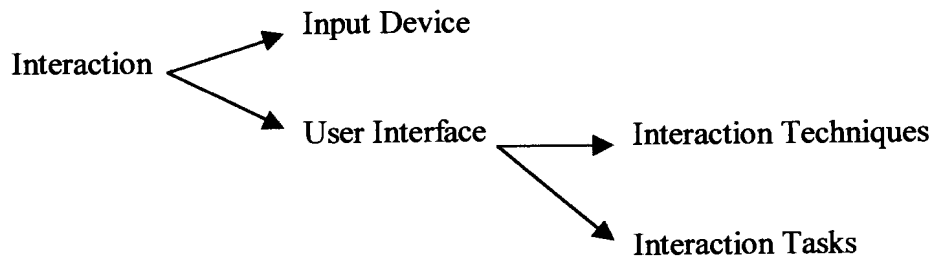


Figure 2.4 Elements of Interaction

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].

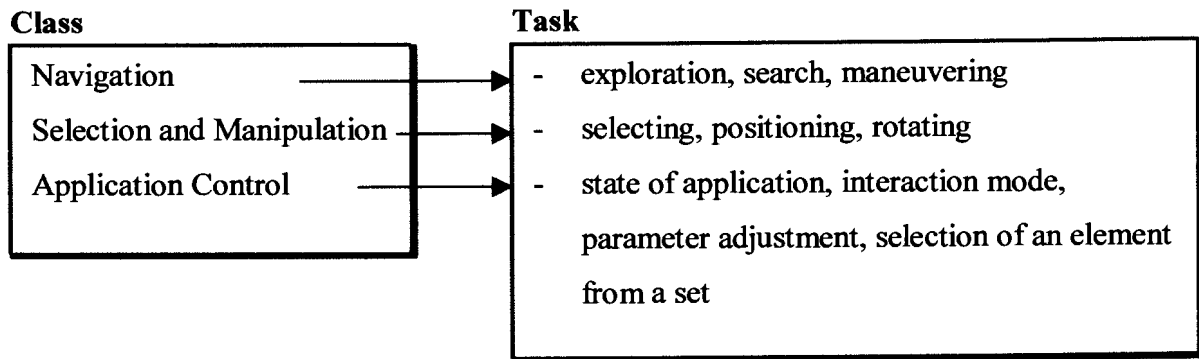


Figure 2.5 Classes of Interaction for 3D User Interfaces

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.

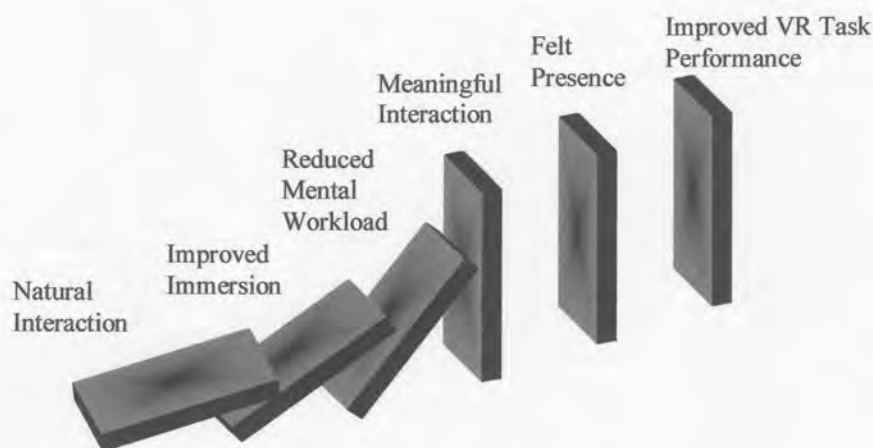


Figure 2.6 Effects of Natural Interaction

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 tetherless 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.

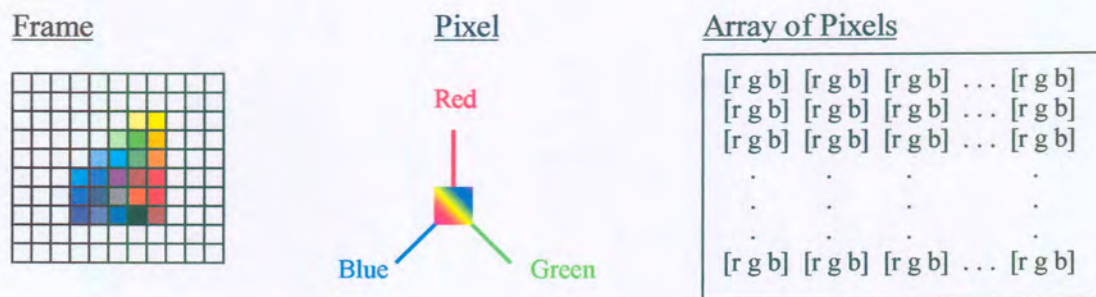


Figure 2.7 Illustration of a Pixel and a Frame and their Representation in Memory

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.

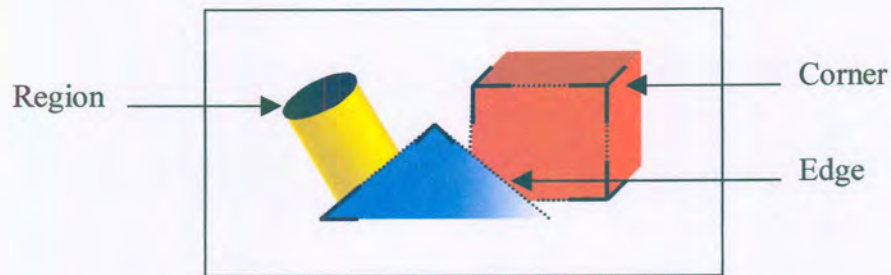


Figure 2.8 Features Obtained During Image Analysis

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 Univalued 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.

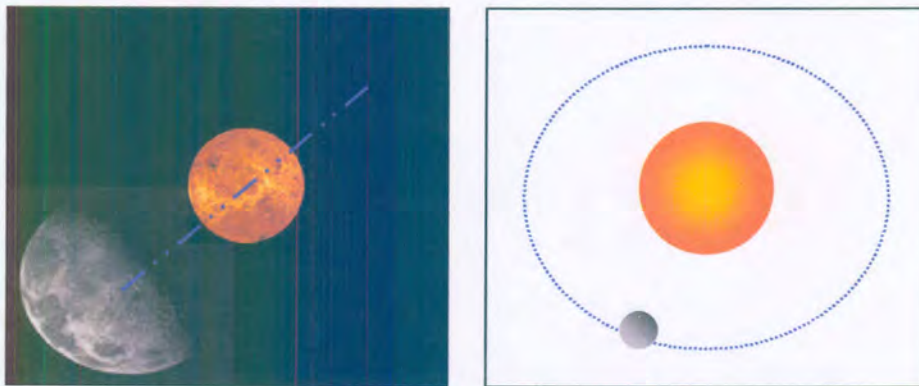


Figure 2.9 Illustration of the 2D to 3D Problem

(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 catoptrics 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

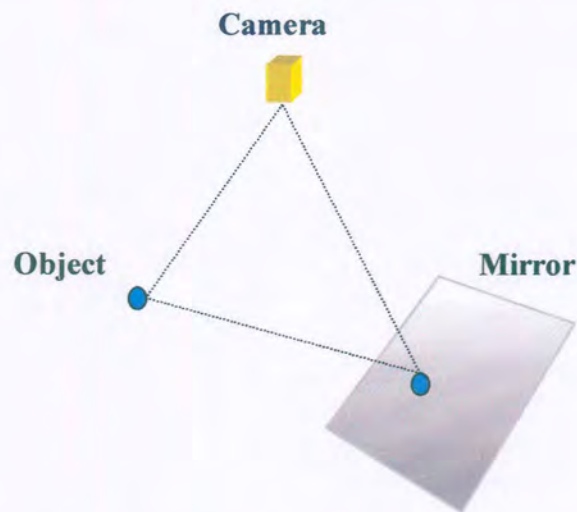


Figure 3.1 Illustration of a Camera and Mirror Setup

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.

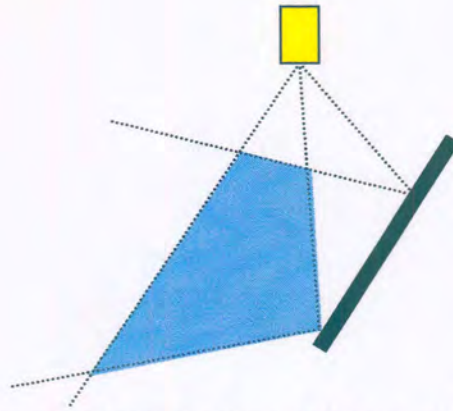


Figure 3.2 Illustration of Region of Interaction

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.

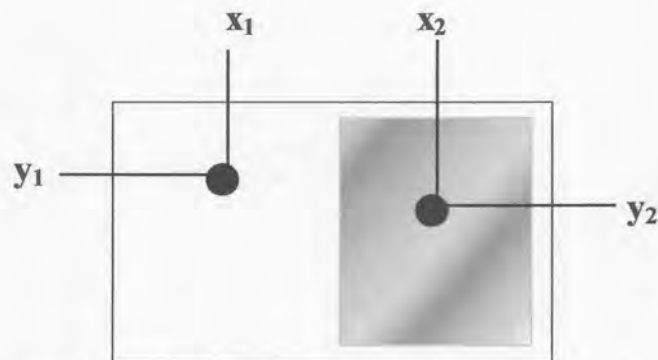


Figure 3.3 Illustration of Where the Image Analysis Phase Locates the Object and its Reflection in an Image

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.

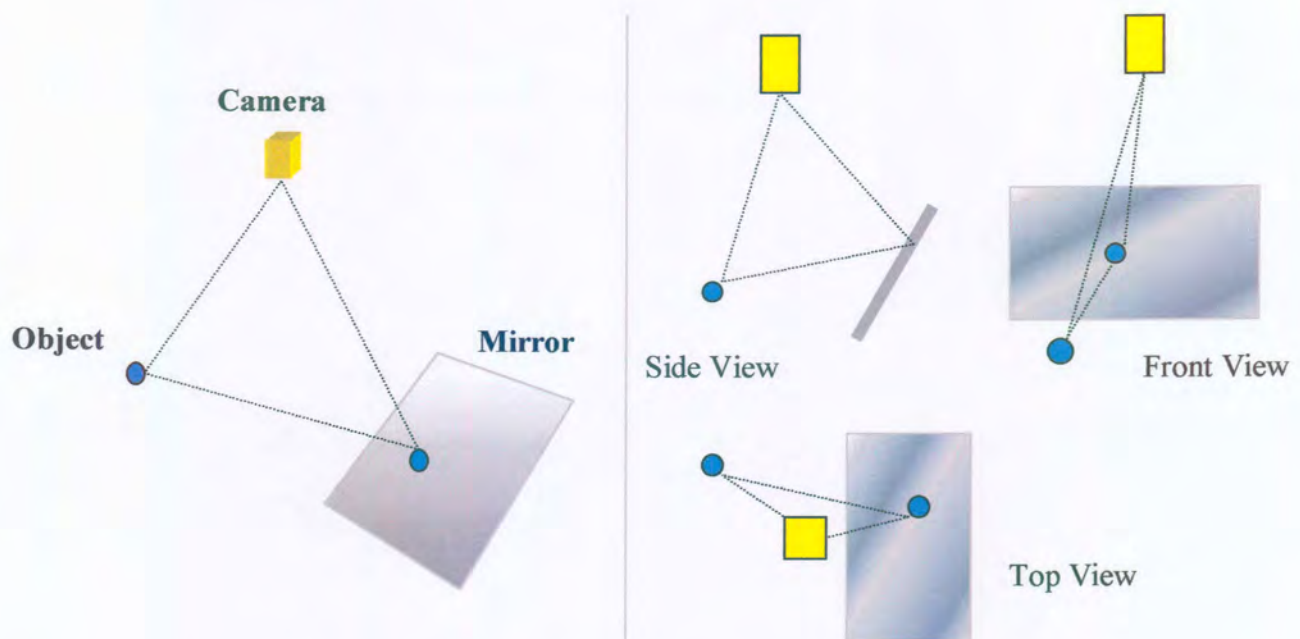


Figure 3.4 Views of the Camera and Mirror Setup

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.

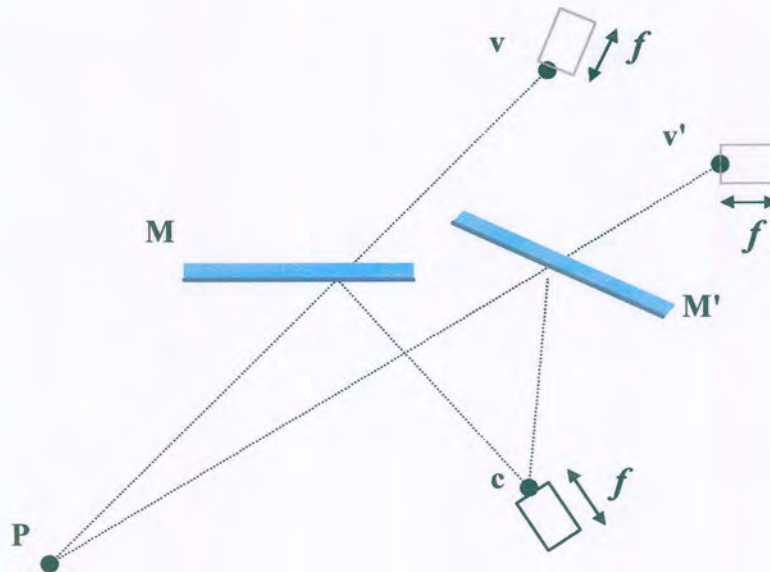


Figure 3.5 Geometry of Catadioptric Stereo

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.

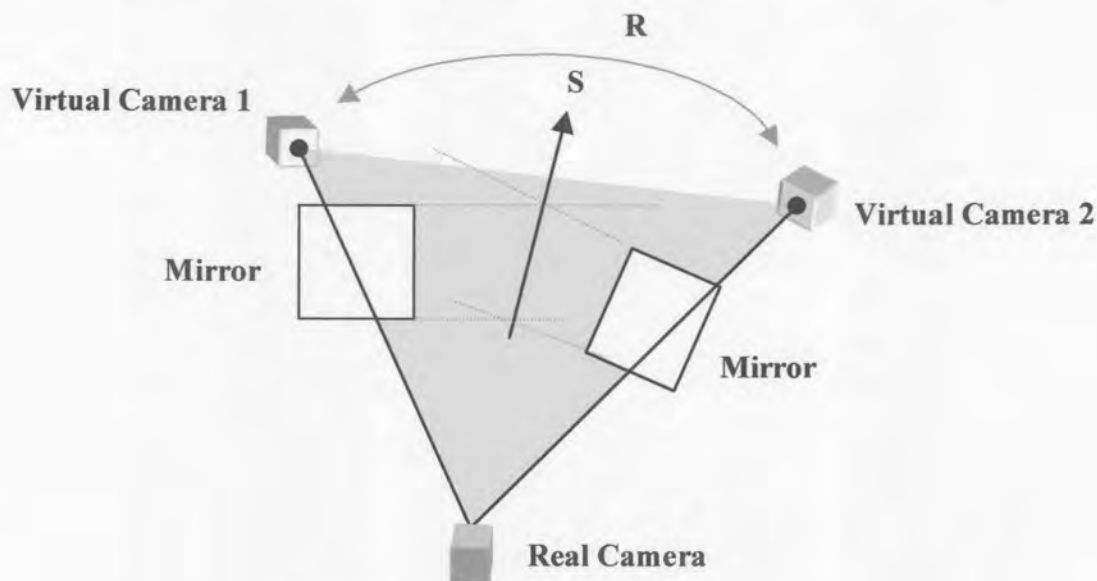


Figure 3.6 Relative Orientation

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.

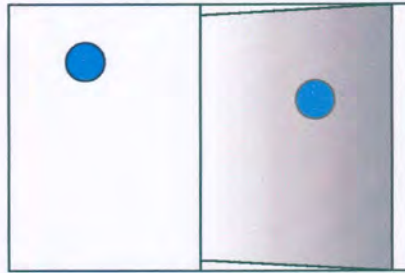


Figure 3.7 Picture Taken by Camera in a Simple Reflective Setup

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 performed 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, \mathbf{m} ,
- the epipolar geometry is simply determined from the two epipoles and the line \mathbf{m} , and
- the horizon line of the plane of motion contains the two epipoles [Nayar & Gluckman].

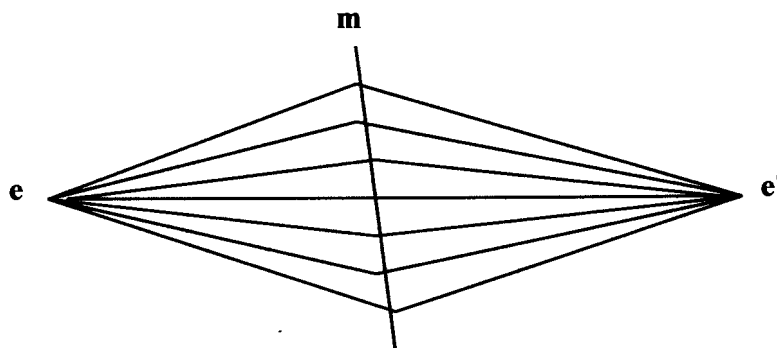


Figure 3.8 Epipolar Lines Must Meet at the Image Projection of the Screw Axis, \mathbf{m}

Figure 3.8 illustrates how corresponding epipolar lines intersect at \mathbf{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.

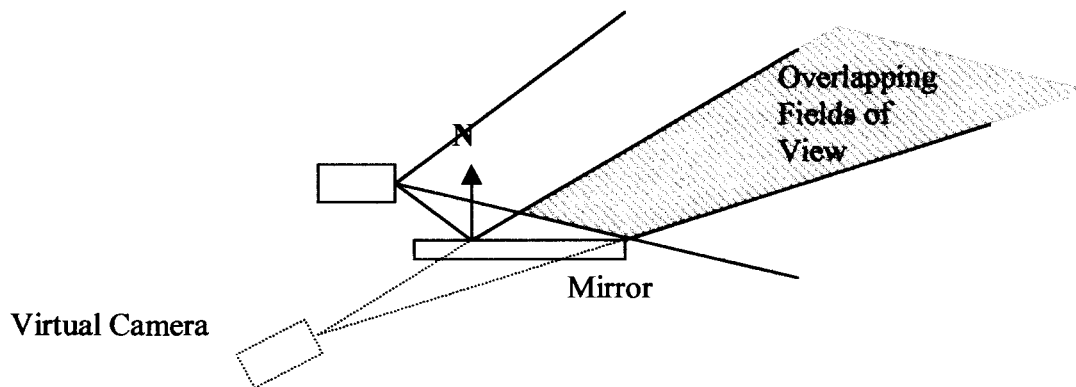


Figure 3.9 Single Mirror Rectified Catadioptric Stereo Sensor

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.

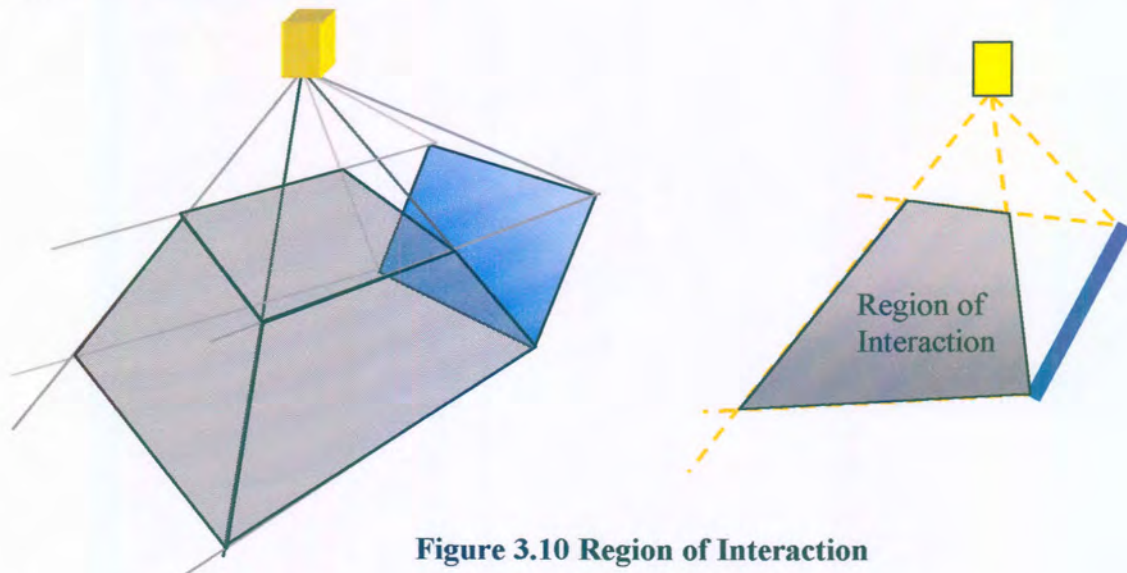


Figure 3.10 Region of Interaction

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.

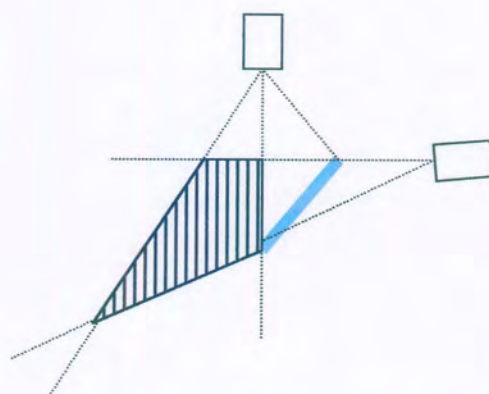


Figure 3.11 Illustration of Overlapping Fields of View

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.

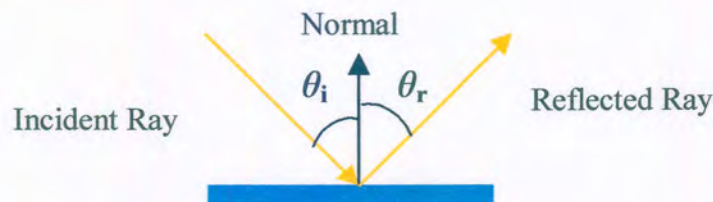


Figure 3.12 Behavior of Light at a Flat Reflective Surface

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 θ_r is equal to the angle of incidence θ_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.

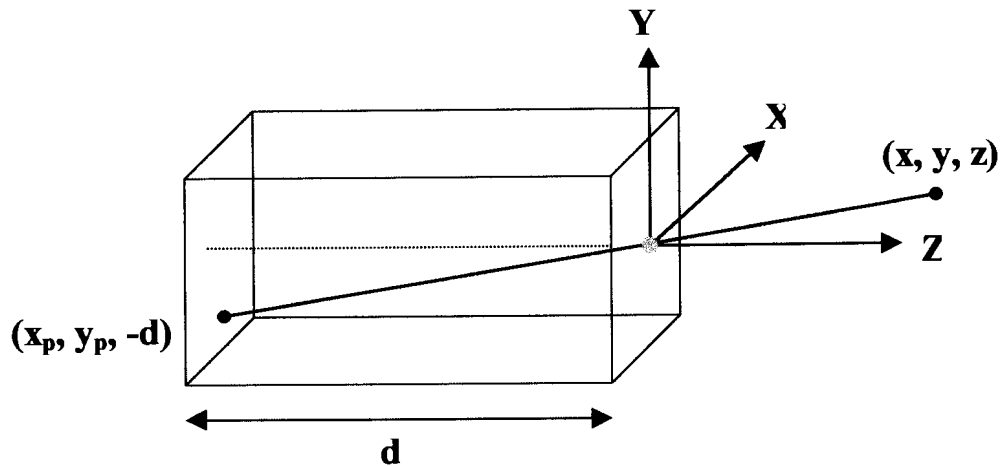


Figure 3.13 The Pinhole Camera

It is important to calculate the projection of the point (x,y,z) onto the film plane. The projection point at $(x_p, y_p, -d)$ 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.

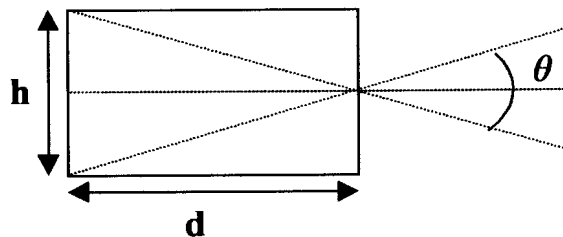


Figure 3.14 The View Angle of a Pinhole Camera

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.

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

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 known, is found in the procedure set out below:

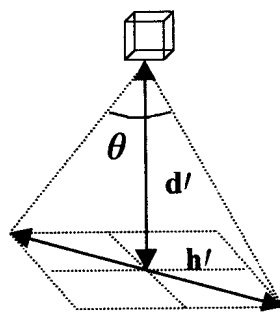


Figure 3.15 Measurements to be Taken to Find the Angle of View

- (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 h' is the largest diagonal length that can be imaged by the camera at a distance d' from the camera. h' and d' are measured in millimeters.

- (2) Next the field of view, θ for this camera is calculated by using the following equation:

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

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

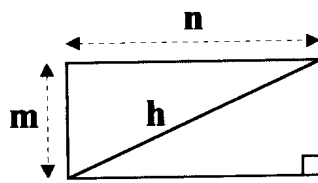


Figure 3.16 Dimensions of Image

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

If the diagonal length of the film is 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 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 d is calculated in millimeters as follows:

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

where θ is the field of view and 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 D_1 units in the direction of the positive X-axis, i.e. it has a point that lies at $(D_1, -H, 0)$.
- (v) The mirror is oriented so that its edges run parallel to the Z and Y-axes.
- (vi) The mirror forms an angle θ 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.

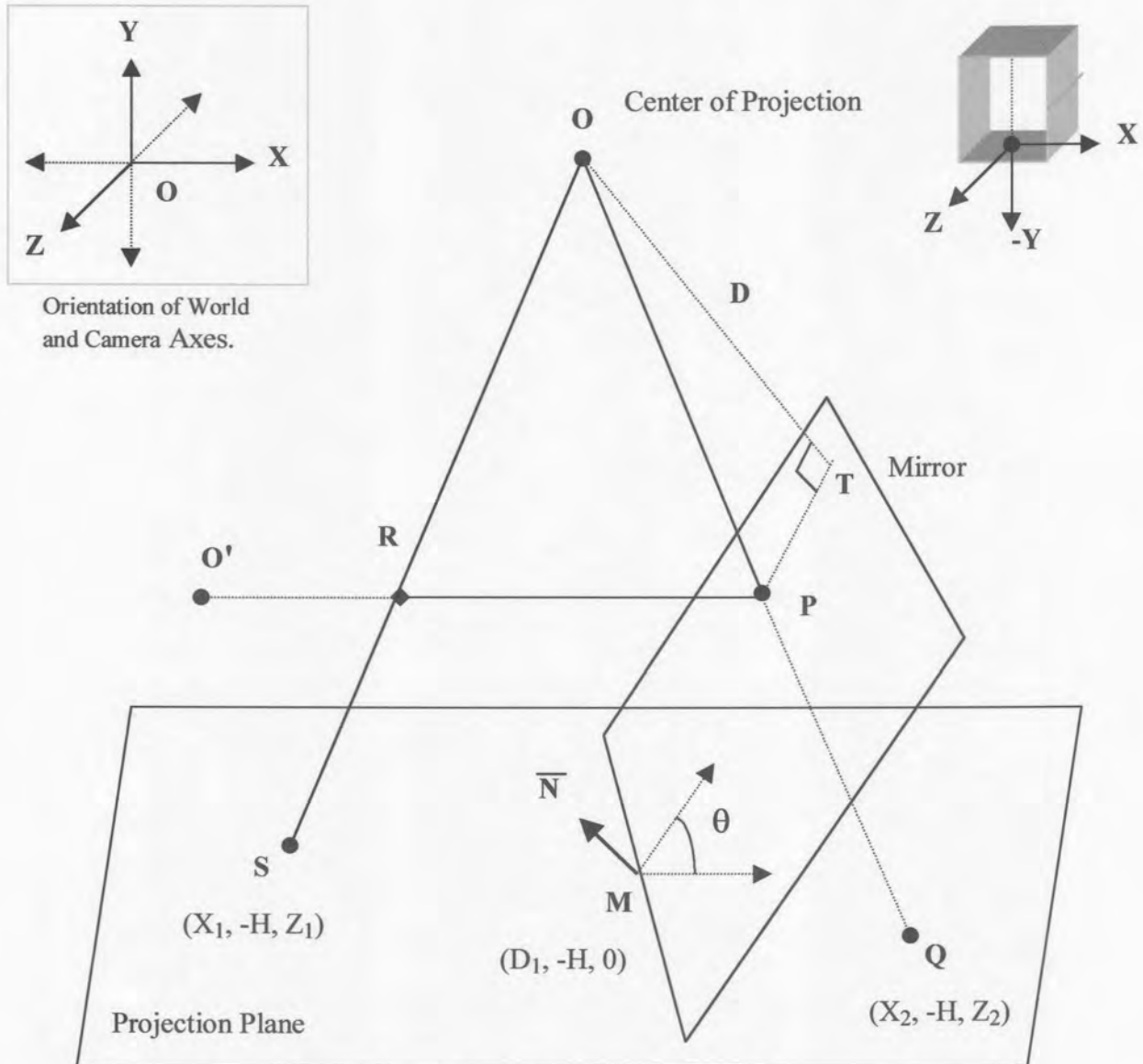


Figure 3.17 Mathematical Model

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 θ 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.

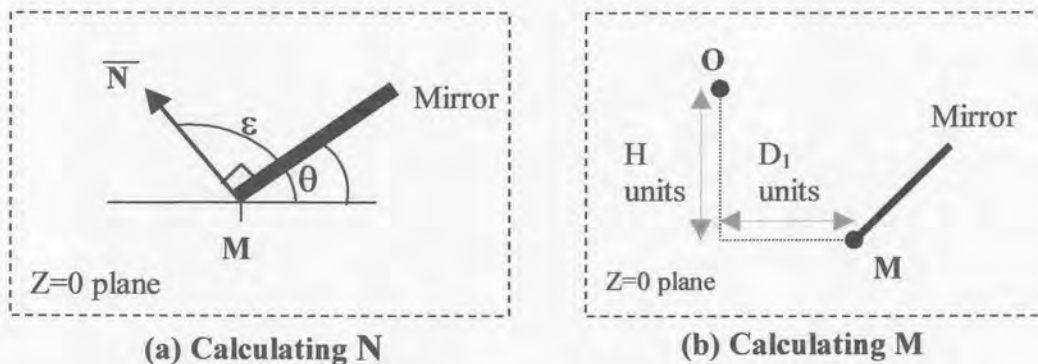


Figure 3.18 Illustration of Measurements

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:

$$\cos \varepsilon (x - D_1) + \sin \varepsilon (y + H) = 0 \quad (3.1.5)$$

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

$$D = \frac{|\mathbf{N} \cdot \overrightarrow{P_0P_1}|}{\|\mathbf{N}\|} \quad (3.1.6)$$

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

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

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

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

$\overrightarrow{P_0P_1}$ 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

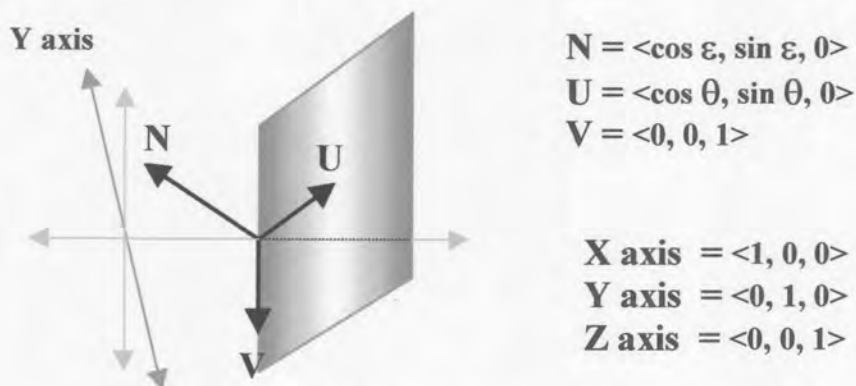


Figure 3.19 The Different Co-ordinate Bases

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:

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

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

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:

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

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

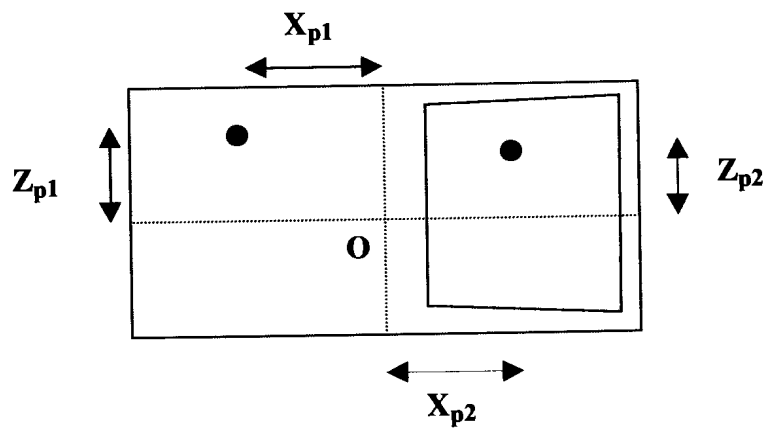


Figure 3.20 Location of Points in an Image.

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:

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

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

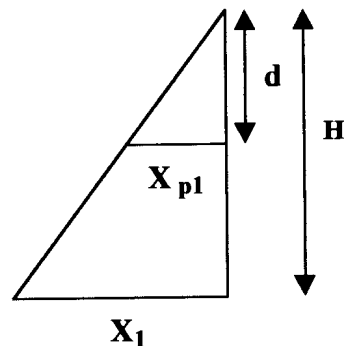


Figure 3.21 Similar Triangles

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:

$$\mathbf{OQ} = \mathbf{0} + t_1 \mathbf{V}_1$$

$$\mathbf{OS} = \mathbf{0} + t_2 \mathbf{V}_2$$

where $\mathbf{V}_1 = \mathbf{Q} - \mathbf{O} = \langle \mathbf{X}_2, -\mathbf{H}, \mathbf{Z}_2 \rangle$

and $\mathbf{V}_2 = \mathbf{S} - \mathbf{O} = \langle \mathbf{X}_1, -\mathbf{H}, \mathbf{Z}_1 \rangle$

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:

$$\cos \varepsilon (r\mathbf{X} * t_1 - \mathbf{D}_1) + \sin \varepsilon (-\mathbf{H} * t_1 + \mathbf{H}) = 0 \quad (3.1.13)$$

Solving for t_1 gives:

$$t_1 = \frac{\cos \varepsilon * \mathbf{D}_1 - \sin \varepsilon * \mathbf{H}}{\cos \varepsilon * (r\mathbf{X}) + \sin \varepsilon * (-\mathbf{H})} \quad (3.1.14)$$

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:

$$\begin{aligned} \mathbf{P} &= \mathbf{0} + t_1 * \langle \mathbf{X}_2, -\mathbf{H}, \mathbf{Z}_2 \rangle \\ &= (\mathbf{X}_2 * t_1, -\mathbf{H} * t_1, \mathbf{Z}_2 * t_1) \end{aligned}$$

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

$$|\mathbf{OP}| = \sqrt{(\mathbf{X}_2 * t_1)^2 + (-\mathbf{H} * t_1)^2 + (\mathbf{Z}_2 * t_1)^2} \quad (3.1.15)$$

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

$$S = (X_1, -H, Z_1), P = (X_2 + t_1, -H + t_1, Z_2 + t_1), O = (0, 0, 0)$$

Calculate length: t_2

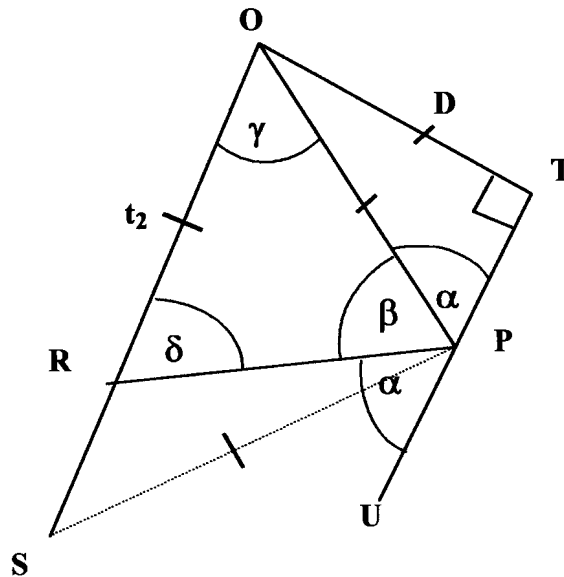


Figure 3.22 Trig/Geometric View of the Mathematical Model

$$\alpha = \sin^{-1} (D / |OP|)$$

$$\sin \alpha = D / |OP|$$

$$\beta = 180 - 2 * \alpha$$

L's on Straight Line = 180

$$|PS|^2 = (X_1 - (X_2 + t_1))^2 + (-H - (-H * t_1))^2 + (Z_1 - (Z_2 + t_1))^2$$

distance of PS

$$|OS|^2 = (X_1)^2 + (-H)^2 + (Z_1)^2$$

distance of PS

$$\gamma = \frac{\cos^{-1} ((PS^2 - OS^2 - OP^2))}{(-2 * OS * OP)}$$

Triangle Rule

$$\delta = 180 - \gamma - \beta$$

L's of $\Delta = 180$

$$OR = \frac{OP \sin \beta}{\sin \delta}$$

$$\frac{\sin \delta}{OP} = \frac{\sin \beta}{OR}$$

$$t_2 = \frac{OR}{OS}$$

Ratio of Line

Since $OS = \mathbf{0} + t_2 \langle X_1, -H, Z_1 \rangle$

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

(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):

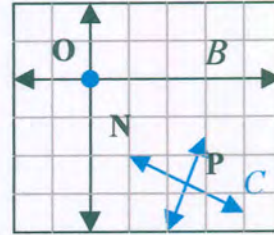
- (i) Determine the equations of the lines \mathbf{OQ} , \mathbf{OS} and the equation of the plane.
- (ii) Determine the point \mathbf{P} where the line \mathbf{OQ} intersects the plane.
- (iii) Find \mathbf{O}' , the reflection of point \mathbf{O} about the normal of the plane at point \mathbf{P} .
This is done using the change of co-ordinate matrices and the reflection matrix.
- (iv) Determine the equation of the line $\mathbf{O}'\mathbf{P}$.
- (v) Find the intersection of line $\mathbf{O}'\mathbf{P}$ and line \mathbf{OS} . This point of intersection gives the 3D co-ordinates of the desired point \mathbf{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 \mathbf{O} about the normal \mathbf{N} located at point \mathbf{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 \mathbf{N} at a point \mathbf{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:

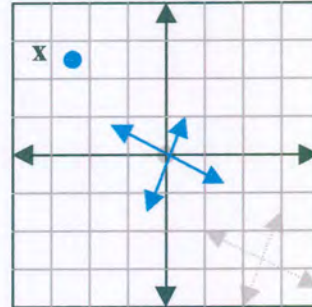


First translate the system of axes at the point **P** and the point **O** so that the system at point **P** lies at the origin (it is necessary to translate the point **O**, so that it lies in its relative position to the system of axes at point **P**).



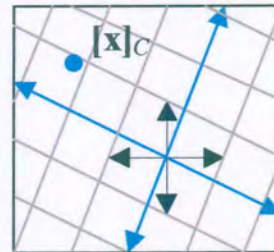
$$\mathbf{x} = \mathbf{O} - \mathbf{P}$$

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



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

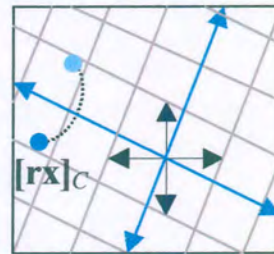
This is done by multiplying **x** by the appropriate change of basis matrix.



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

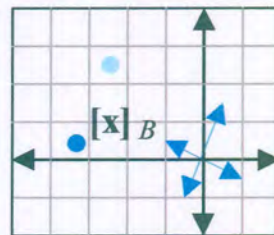
$$[\mathbf{rx}]_C = \mathbf{R} [\mathbf{x}]_C$$

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**.



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

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



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

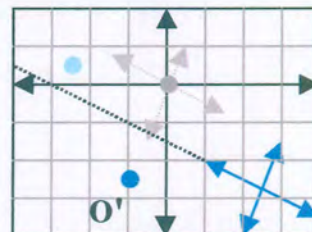


Figure 3.23 Illustration of Reflecting the Point about the Mirror

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

$$O'P = P + t_3 * V_3$$

Where $V_3 = \langle P - O' \rangle$.

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.

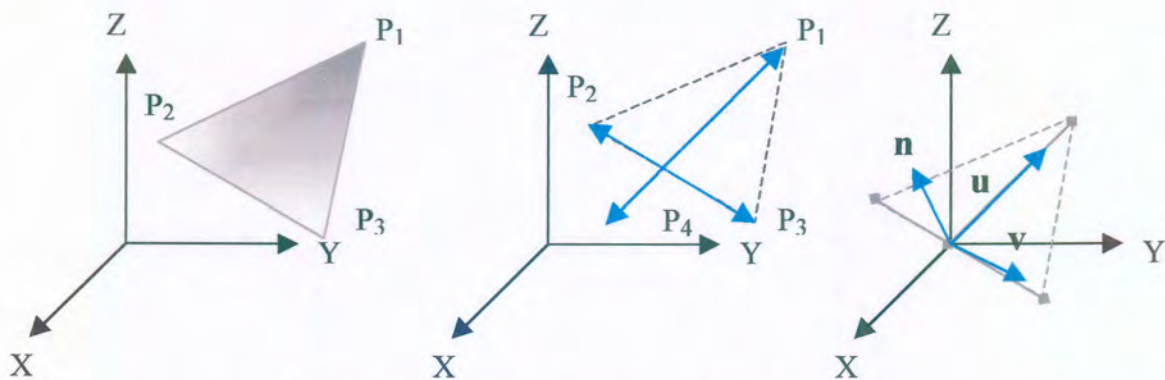


Figure 3.24 Calculation of 5D/6D Information

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:

$$u = P_4P_1 = \langle P_1 - P_4 \rangle$$

$$v = P_2P_3 = \langle P_3 - P_2 \rangle$$

$$n = u \times v$$

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

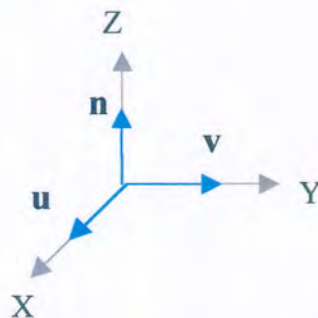


Figure 3.25 The System of Axes after Applying All the Rotations

The angles of rotation are found by the following procedure:

- (1) Calculate θ , then ϕ using the spherical polar co-ordinates of \mathbf{u} . These are simply the angles \mathbf{u} forms with the XZ plane and the X axis.
- (2) rotate the vectors \mathbf{u} , \mathbf{v} and \mathbf{n} , $-\phi$ about the Z-axis so that \mathbf{u} lies in the XZ plane.
- (3) rotate the vectors \mathbf{u} , \mathbf{v} and \mathbf{n} , $-(90-\theta)$ about the Y-axis so that \mathbf{u} lies on the X-axis.
- (4) Calculate γ , the angle that the rotated vector \mathbf{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 \mathbf{u} is shown and the angles it makes with the Y and Z-axes. Steps one and two above require \mathbf{u} to be reflected in such a way that it lies on the X-axis.

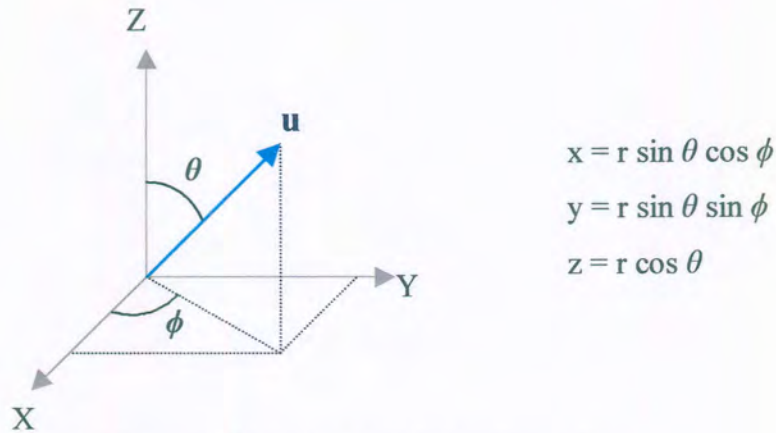


Figure 3.26 Spherical Polar Co-ordinates

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:

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

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

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

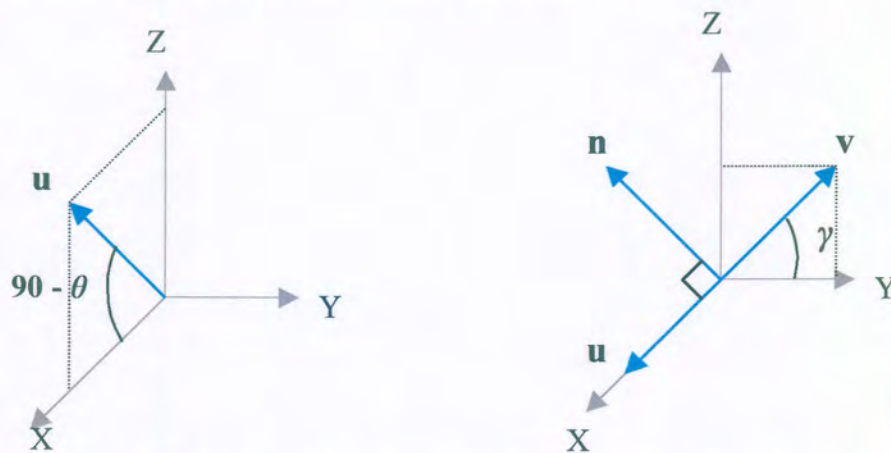


Figure 3.27 System After Rotating by $-\phi$ and by $-(90 - \theta)$

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

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

The final rotation by γ must result in \mathbf{v} lying on the Y-axis and \mathbf{n} lying on the Z-axis.

The final angles of rotation that express the orientation of the object are thus: (θ, ϕ, γ) .

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

(θ, ϕ, γ) . - 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 \mathbf{u} , \mathbf{v} and \mathbf{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 its 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,

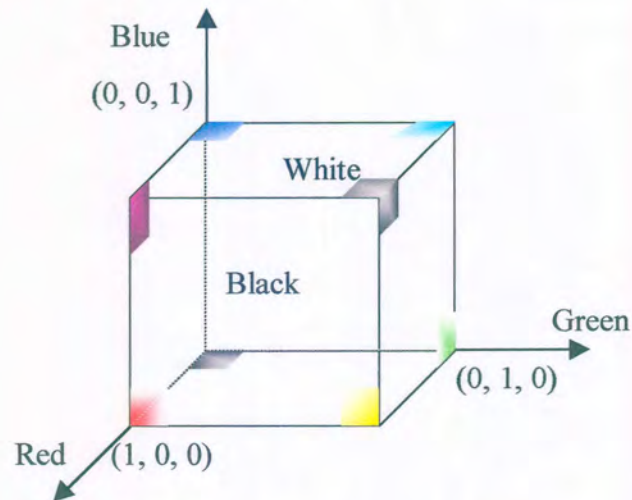


Figure 3.28 RGB Colour Space

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.

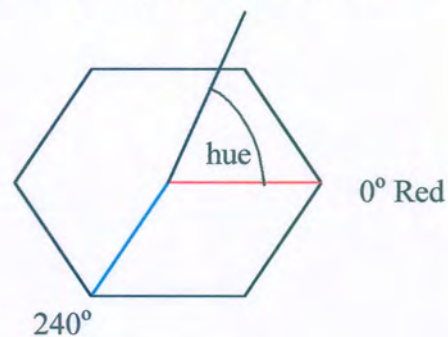


Figure 3.29 HLS Color Model Viewed From Above

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:

$$\mathbf{B} > \mathbf{R} \text{ and } \mathbf{B} > \mathbf{G} \quad (3.2.1)$$

This criteria does not prevent impure shades of gray from being considered blue. For instance a pixel with values $[\mathbf{R}, \mathbf{G}, \mathbf{B}] = [0.99, 0.99, 1]$ 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:

$$\mathbf{d} = \sqrt{(\mathbf{B} - \mathbf{R})^2 + (\mathbf{B} - \mathbf{G})^2} > \mathbf{d}_{\max} \quad (3.2.2)$$

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

$$\mathbf{d} = 2\mathbf{B} - \mathbf{R} - \mathbf{G} \quad (3.2.3)$$

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 $B > G$.

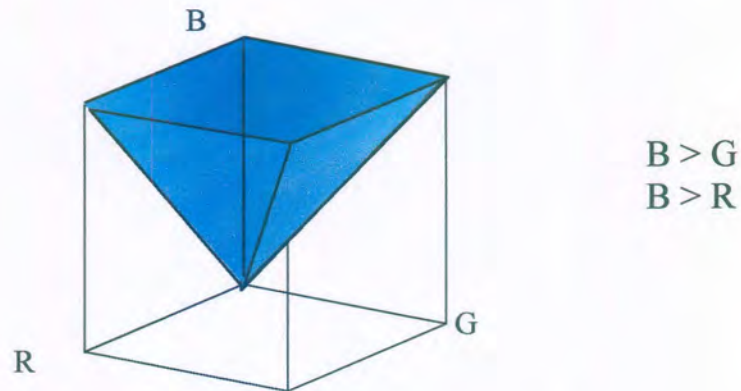


Figure 3.30 First Set of Constraints

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.

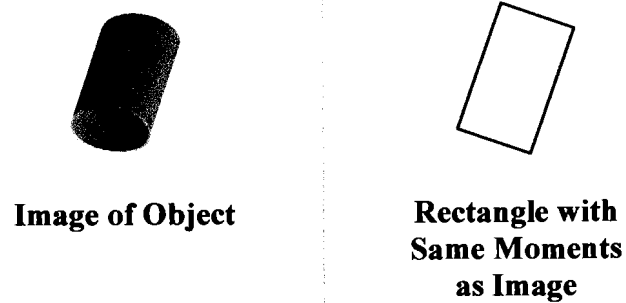


Figure 3.31 Illustration of Image Moments

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

- position, x_c, y_c ,
- orientation, θ , and
- the dimensions, L_1, L_2 .

These features are illustrated in figure 3.32.

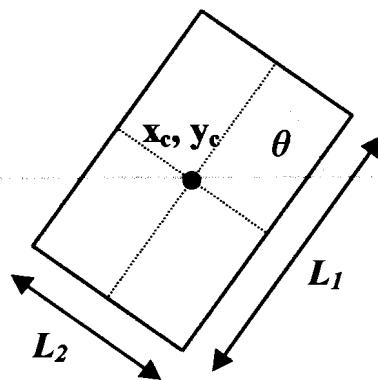


Figure 3.32 Features Calculated by Image Moments

Image moments are calculated as follows:

$$\begin{aligned}
 M_{00} &= \sum_x \sum_y I(x, y) & M_{11} &= \sum_x \sum_y xy I(x, y) \\
 M_{10} &= \sum_x \sum_y x I(x, y) & M_{01} &= \sum_x \sum_y y I(x, y) \\
 M_{20} &= \sum_x \sum_y x^2 I(x, y) & M_{02} &= \sum_x \sum_y y^2 I(x, y)
 \end{aligned}
 \tag{3.2.4}$$

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

$$x_c = \frac{M_{10}}{M_{00}} \quad y_c = \frac{M_{01}}{M_{00}} \quad (3.2.5)$$

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

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

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

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

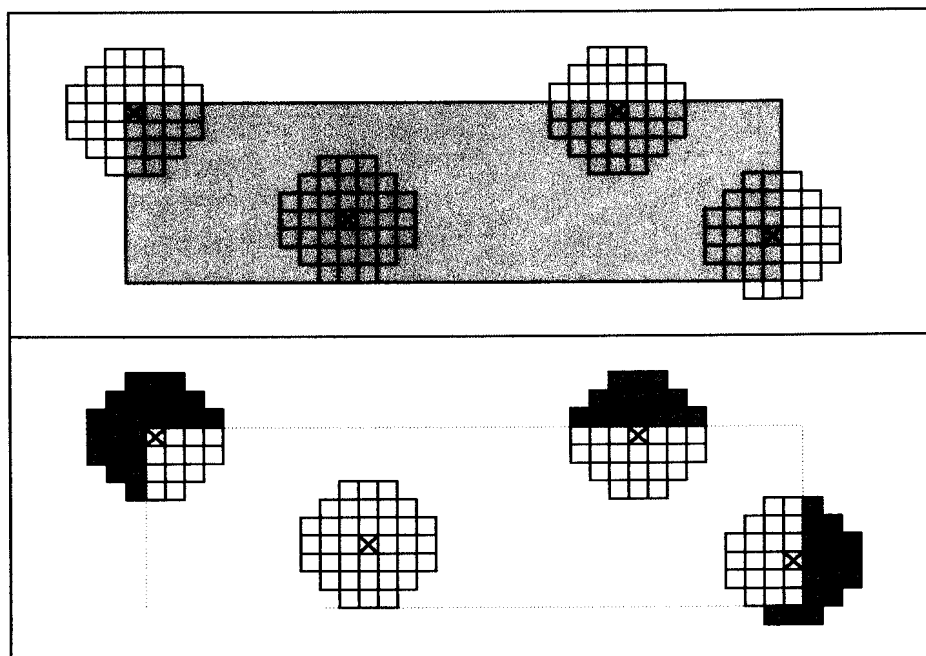


Figure 3.33 SUSAN Masks at Different Positions in an Image

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 (Univalued 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

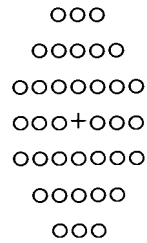


Figure 3.34 A Typical SUSAN Mask

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:

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

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).

$$n(\mathbf{r}_0) = \sum c(\mathbf{r}, \mathbf{r}_0) \quad (3.2.11)$$

(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:

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

(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.

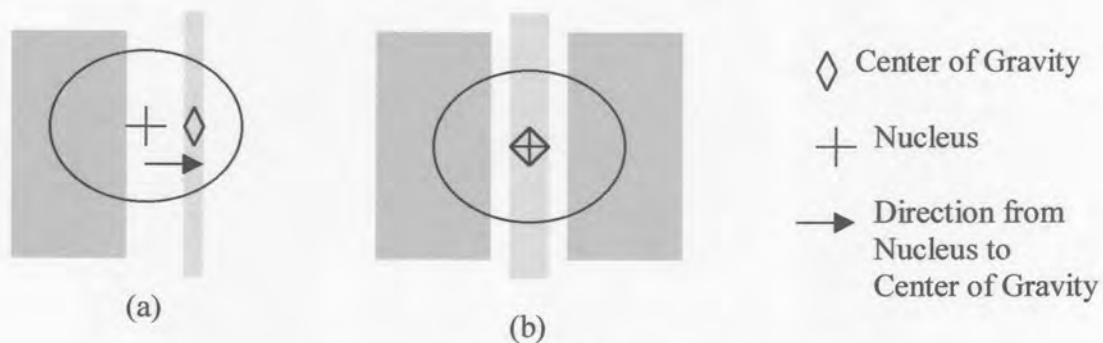


Figure 3.35 Two Different Edge Types

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 \mathbf{v} from the nucleus to the center of gravity are found. The direction perpendicular to \mathbf{v} gives the direction of the edge. The center of gravity is calculated as follows:

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

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:

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

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

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

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.

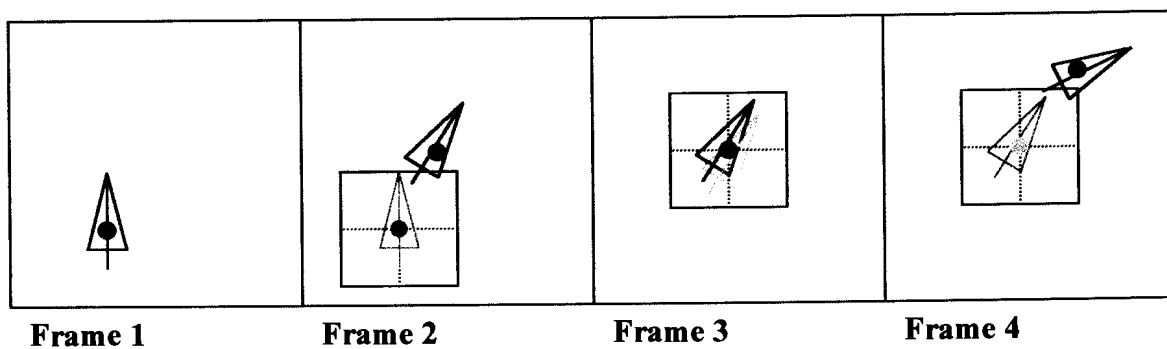


Figure 3.37 A simple Window Tracking System

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.