# Chapter 4

# Reflections in Interactive Applications

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

## 4.1 DESIGN AND IMPLEMENTATION

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

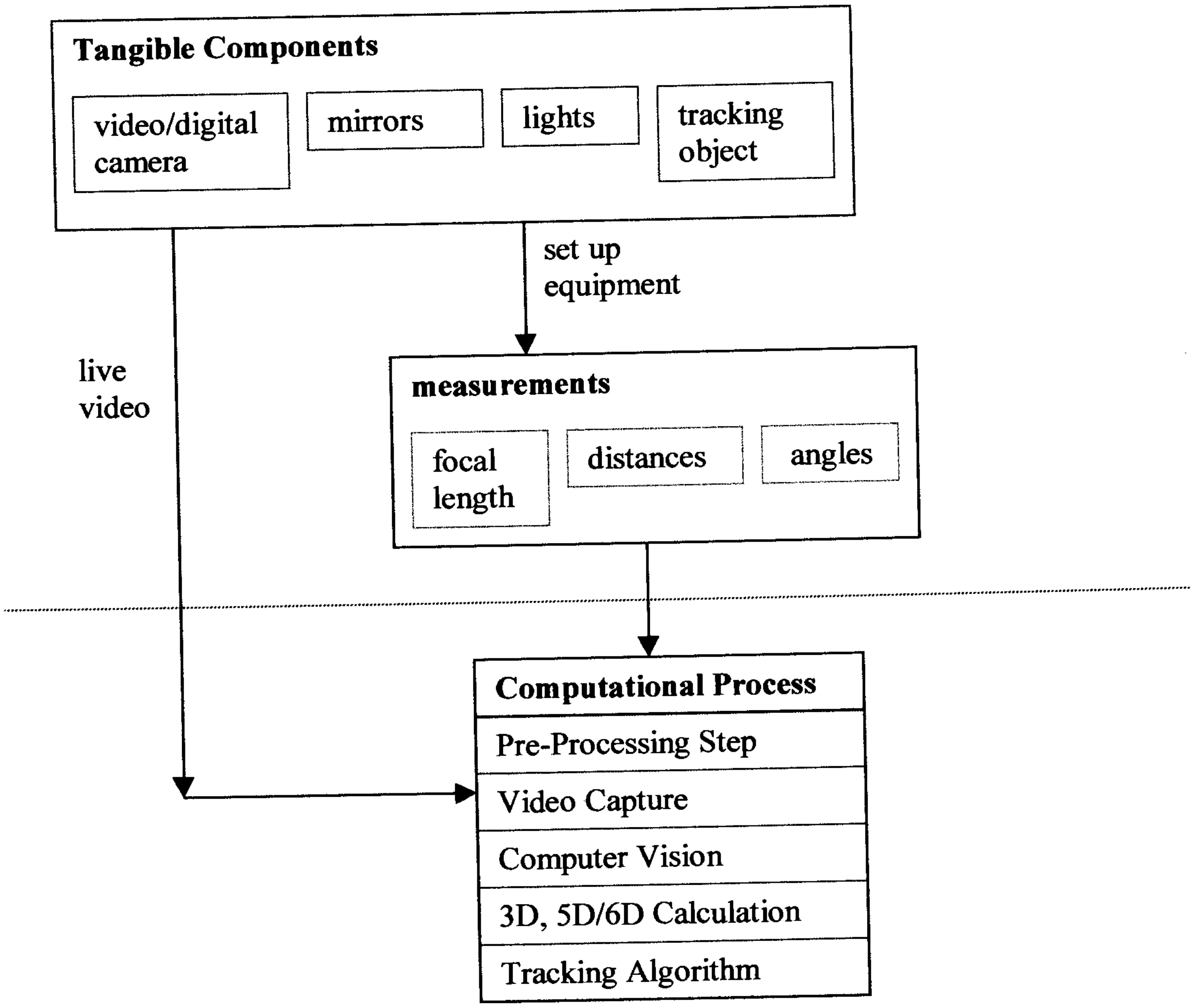


**Figure 4.1 Flowchart of a General Reflections System**

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

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

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

### 4.1.1 The Physical Installation Process

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

(1) position the image capture device so that it views the required mirrors and region of interaction,
(2) place the reflective surfaces in suitable positions with suitable orientations so that the region of interaction is viewable by the camera,
(3) install the lighting and necessary filters, and
(4) measure the position and rotations of the mirrors relative to the camera.

When orienting the mirrors and installing the equipment it is necessary to see the view of the camera. This is important for correctly and optimally viewing the region of interaction. For this reason the installation process contains a video capture software component for simply displaying live video. In the systems implemented in this thesis the computational process has the option of capturing and displaying frames without performing any calculation.

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

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

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

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

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

Finding the optimal position for the equipment requires careful planning and thought. This is because there is a tradeoff between accuracy and the size of the interaction volume. In some environments it is difficult to find a suitable position to place the camera and mirrors while maintaining the required accuracy and still covering the entire region of interaction.

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

### 4.1.2 The Computational Process

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

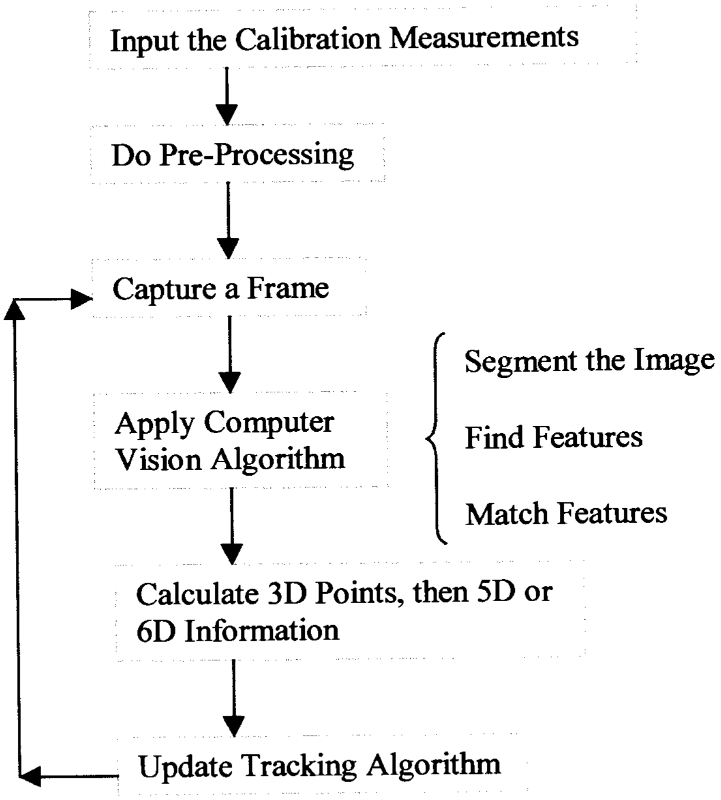


**Figure 4.2 Flow Chart of the Computational Process**

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

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

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

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

## 4.2 VIRTUAL DRUMS

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

### 4.2.1 The Application

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

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

#### (a) Visualization

The first aspect of the virtual drum kit is the visual component. This includes using 3D computer graphics to create a realistic model of a drum kit. A picture of a real drum kit is shown in figure 4.4 This real drum kit was used as the model for the creation of the AVANGO version of the virtual drum kit.

**Figure 4.4 A Yamaha Drum Kit**

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

### (b) Sound

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

### (c) Interaction

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

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

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

### 4.2.2 The Prototype

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

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

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

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

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

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

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

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

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

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

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

### 4.2.3 AVANGO Implementation

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

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

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

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

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

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

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

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

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

In the desktop implementation two mirrors are used. These are placed against the ceiling with the camera as illustrated in figure 4.7. This is possibly the best position for the camera and mirrors as it is the position in which occlusions are minimized and from which almost the whole length of the drumsticks are seen. It is important to see the length of the drumsticks in order to calculate the 5D information.

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

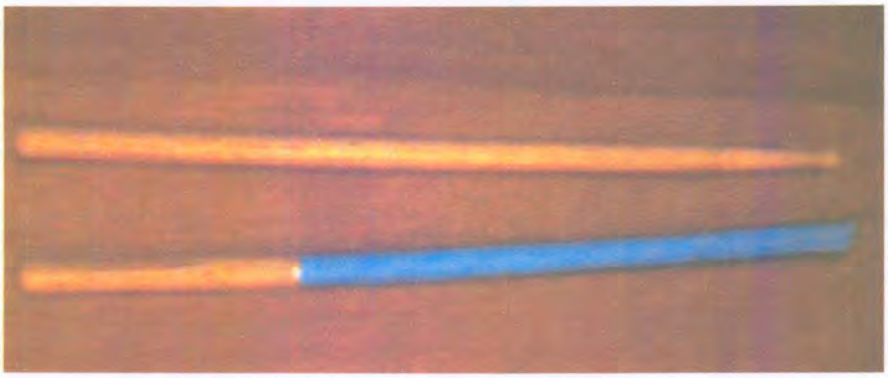


**Figure 4.8 Blue Drumsticks Tracked in the desktop Virtual Drums Installation**

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

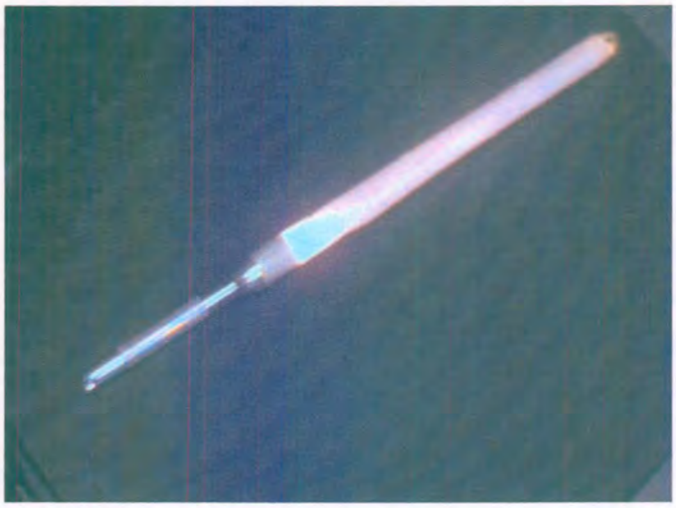


**Figure 4.9 Light Stick Used In CyberStage Virtual Drum Kit**

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

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

### a) The Computer Vision and Tracking Algorithms

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

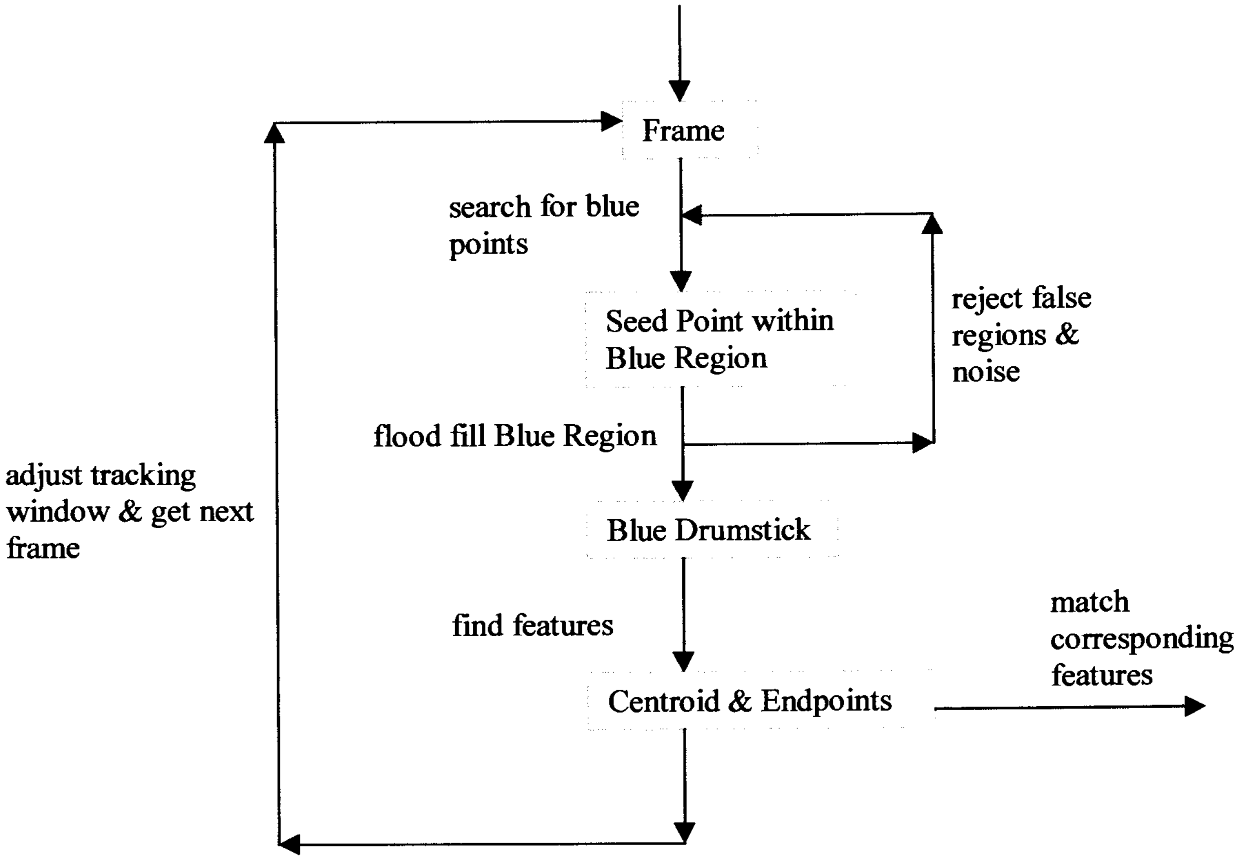


**Figure 4.10 The Computer Vision Algorithm Used in the Practical Implementation of the Virtual Drums Project**

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

Figure 4.11 illustrates a picture of a catadioptric stereo image of a drummer with a blue drumstick. To find a blue drumstick inverse chroma-keying is used with a linear 2D grid search, which, much like a dragnet, does not test every pixel in the image, but samples the pixels in a 2D grid from left to right and top to bottom until it finds a blue pixel. A blue pixel found by the search may belong to a drumstick or it may simply be noise. For this reason further testing is performed for each blue point found by the grid search. The search ends when the drumstick is found and segmented.

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

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

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

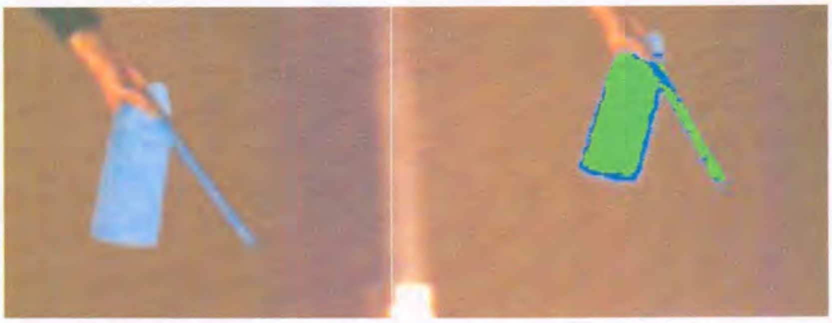


**Figure 4.12 SUSAN Applied to a Drumstick and a Piece of Paper**

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

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

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

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

**Figure 4.13 Image Moments of a Drumstick**

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

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

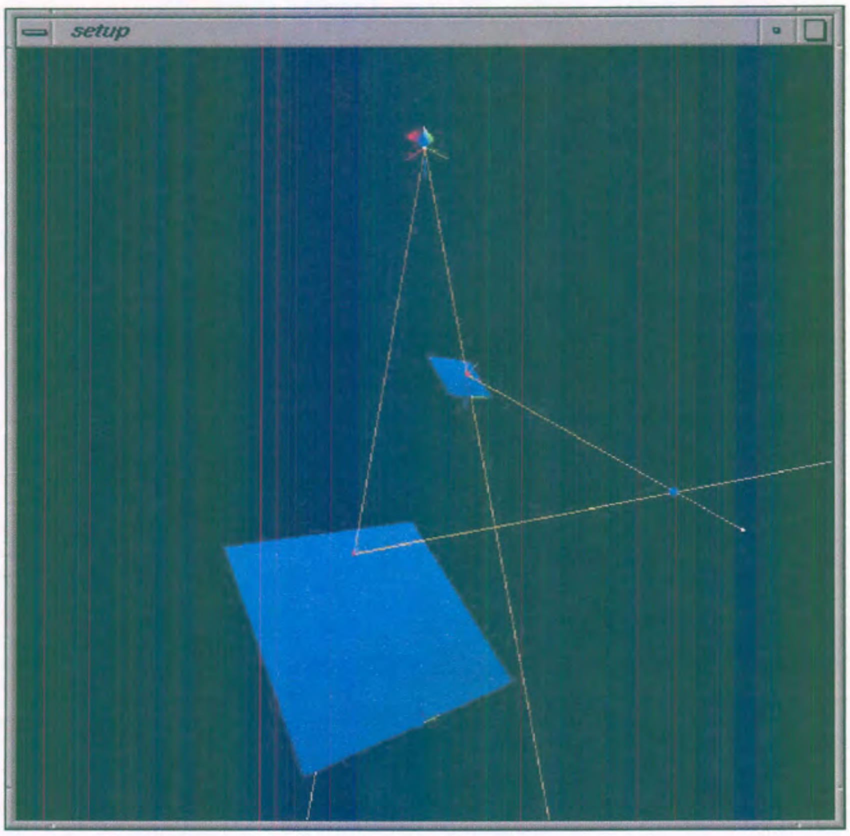


**Figure 4.14 Visualization of the Mathematical Model**

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

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

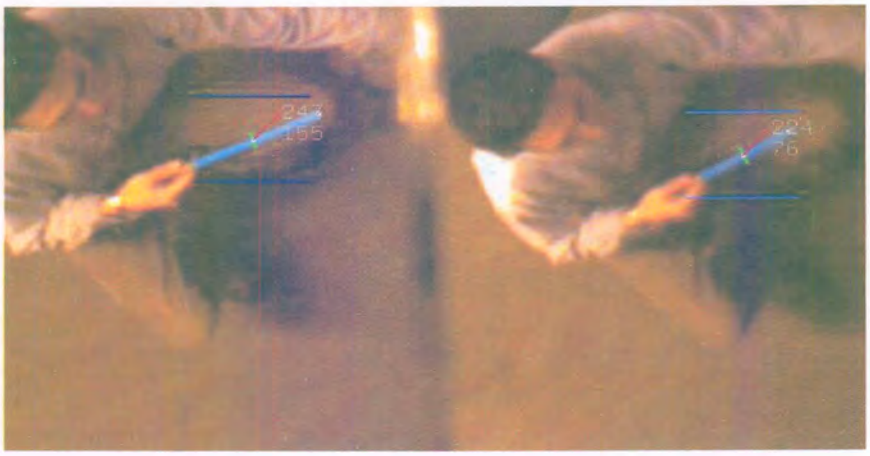
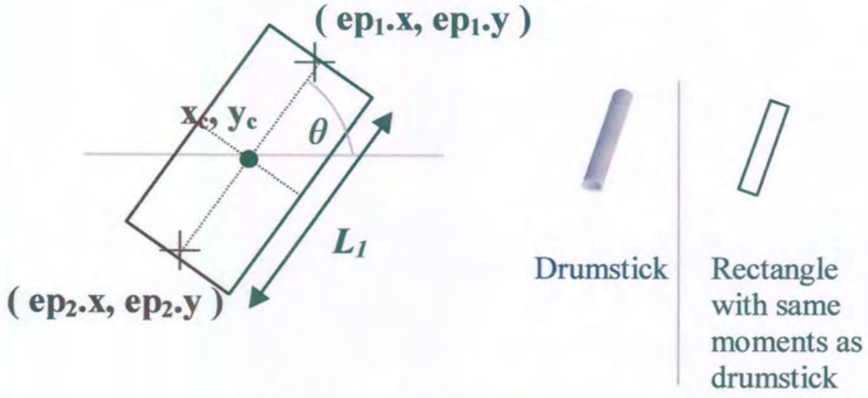


**Figure 4.15 Tracking a Blue Drumstick**

### (b) The 5DOF Extension

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

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



$ep_1.x = x_c + L_1/2 * \cos(\text{ theta });$
$ep_1.y = y_c + L_1/2 * \sin(\text{ theta });$

$ep_2.x = x_c - L_1/2 * \cos(\text{ theta });$
$ep_2.y = y_c - L_1/2 * \sin(\text{ theta });$

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

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

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

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

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

### (c) Tracking Multiple Drumsticks

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

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

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

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

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

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

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

## 4.3 NDEBELE PAINTING

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

**Figure 4.18 Ndebele Paintings**

### 4.3.1 The Existing Application

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

**Figure 4.19 Ndebele Painting in VR**

Ndebele Painting in VR is a virtual environment in which one can create Ndebele paintings. The project focus is on the process of wall painting and includes interaction metaphors that are natural to real wall painting. In the creation of this virtual painting world certain challenges which face design tools for painting in virtual environments are encountered. The project incorporates algorithms which assist the artist's creative process. Paintings are easily and effectively positioned so that they share with a snapping algorithm. Another useful algorithm for the artist is an adaptation of an algorithm implemented by GMD, which simulates painting with a real paintbrush. With an extension of this algorithm the artist paints within different boundaries and lines without the paint spilling over into other regions.

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

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

### 4.3.2 Ndebele Painting Meets Reflections

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

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

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

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

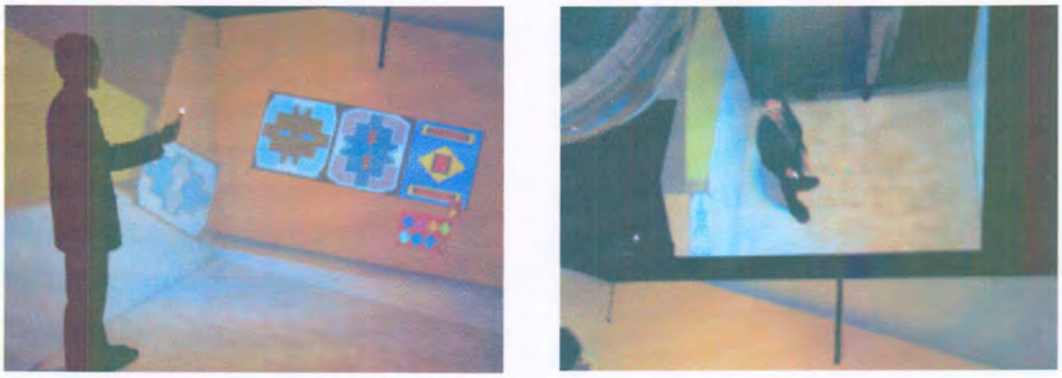


**Figure 4.21 Reflections in Ndebele Paintings**

The computer vision algorithm searches the images from left to right and top to bottom for points of light. A point with an above threshold value for one of its component colours is found. This corresponds to a point on the torch. Because this torch is uncovered the brightness of the torch is significantly brighter than anything projected onto the walls of the CyberStage. In this application the torch is successfully tracked in both views. The window tracker used in the other applications is also used in this implementation.

## 4.4 DEVELOPMENT PLATFORM

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

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

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

## 4.5 IMPLEMENTING REFLECTIONS IN DIFFERENT VIRTUAL ENVIRONMENTS

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

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

### 4.5.1 Desktop

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

### 4.5.2 Responsive Workbench

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

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

### 4.5.3 CyberStage

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

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

## 4.6 SUMMARY

In this chapter the design and implementation of a Reflections system are discussed in detail. The blueprint of the system is given in section 4.1. A Reflections system consists of both a physical installation process and a computational process. Practical aspects and steps are given in section 4.1.1 for installing the tangible components of a Reflections system. The requirements of the computational process are set out in section 4.1.2. These design issues are implemented in two different applications. The Virtual Drums project and Ndebele Painting. The Virtual Drums project makes use of Reflections to facilitate natural, non-intrusive and multidimensional interaction in a virtual drum kit. This project illustrates the use of Reflections in both a desktop and CAVE environment. There are two versions of the Virtual Drums project. The AVANGO implementation demonstrates the many features that are implemented. The desktop version of the AVANGO Virtual Drum kit is capable of tracking coloured drumsticks, while the CyberStage implementation tracks a light stick. An extensive computer vision algorithm is used in both of the AVANGO Virtual Drums implementations. The CV makes practical use of several algorithms including a fast flood fill algorithm, chroma-keying (for well lighted environments), image moments and SUSAN to track the drumstick and to determine its endpoints. This implementation also tracks the 5D position of the drumstick. A discussion of tracking multiple objects is also included in this chapter.

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

# Chapter 5

# Results

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

## 5.1 ACCURACY

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

### 5.1.1 Position

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

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

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

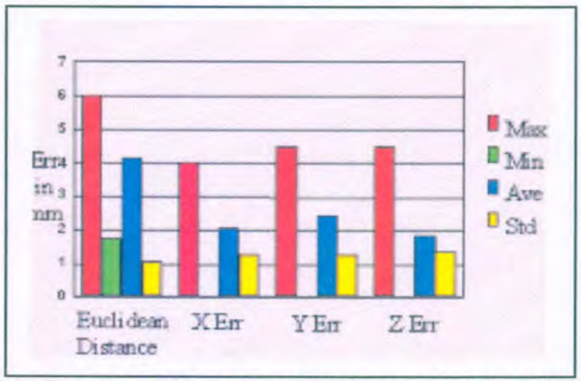


**Figure 5.1 Graph of Results of Trigonometric 3D Calculation**

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

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

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

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

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

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

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

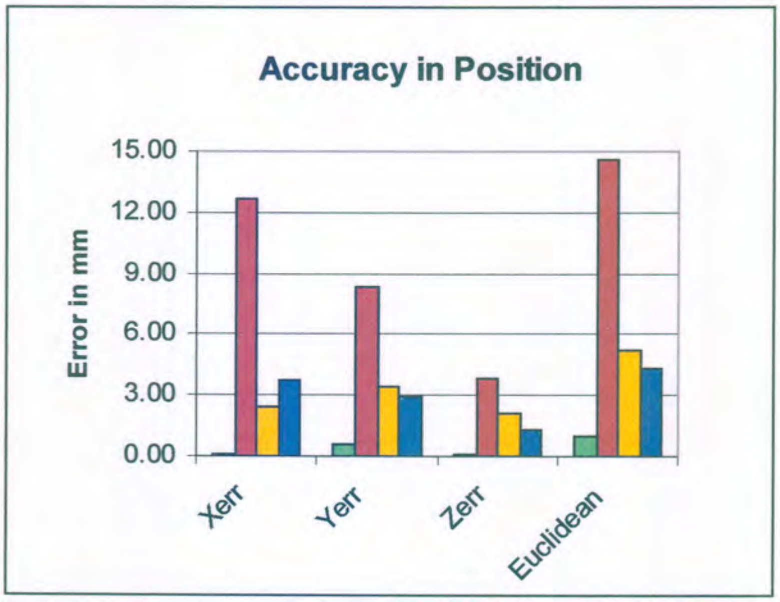
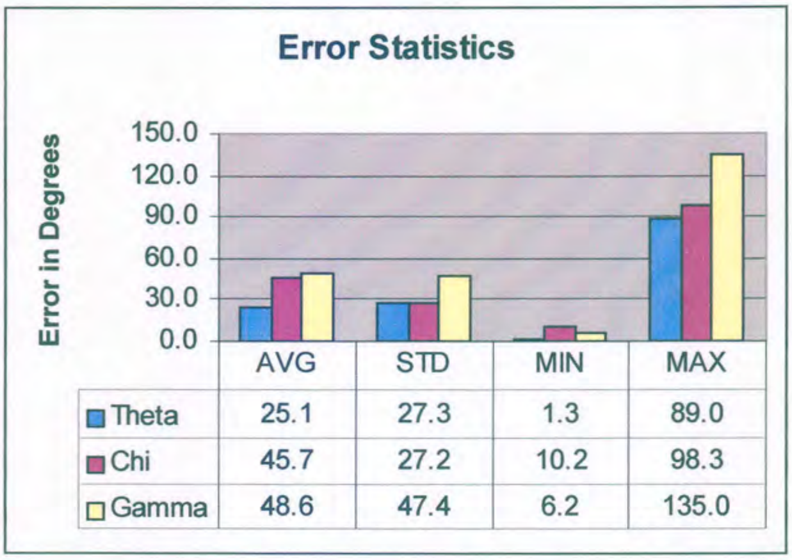


**Figure 5.2 Accuracy of 3D Calculation Using the Algebraic Approach**

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

### 5.1.2 Angular Accuracy

The accuracy of the 6D calculation is tested using the algebraic approach for 3D calculation and for the same environment. In this environment the four midpoints on the edges of an A4 sized blue page are found using image moments and matched with their corresponding points in the stereo image. The matched four endpoints are used to determine the angular axes of the page. From these axes the orientation of the page is calculated. The results are listed below in table 5.3 with error statistics given in degrees.



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

**Table 5.3 Angular Accuracy Results**

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

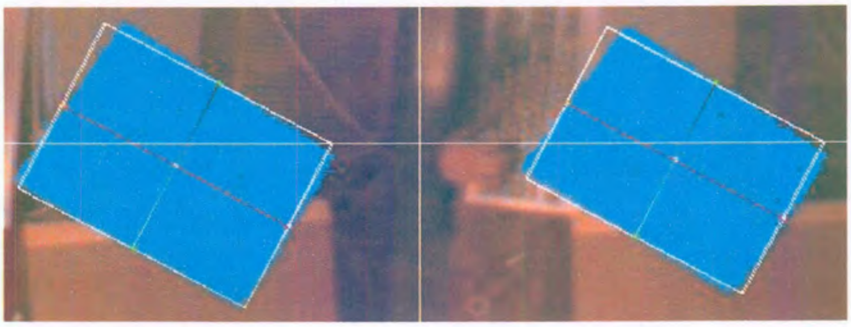


**Figure 5.3 Image Moments of A4 Page Used in Angular Tests**

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

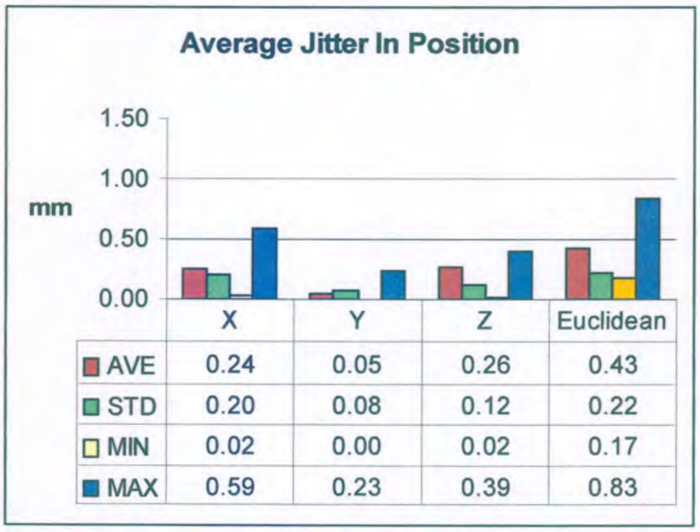
**Figure 5.4 Endpoints Found by Image Moments**

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

Another issue pertaining to angular measurements is the range of the angles of rotation. Angular rotations are limited by the midline between the camera and mirror, or between the two virtual cameras. In the case of using the A4 page the range of rotation around the Z-axis is 180 degrees since an 180-degree rotation of the page is identical to a zero degree rotation of the page. This is because, to the computer vision algorithm a page rotated $a$ degrees appears identical to a page rotated $a$ + 180 degrees and the angle reported is therefore the same. The ranges on the other angles depend on the shape and position of the object relative to the camera. The rotation of an object can be determined as long as the object is not tilted in such a way that it obstructs the camera's view of it's important features, i.e. the object must not become self occluding.

### 5.1.3 Stability

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



| | X | Y | Z | Euclidean |
|---|---|---|---|---|
| 🟥 AVE | 0.24 | 0.05 | 0.26 | 0.43 |
| 🟩 STD | 0.20 | 0.08 | 0.12 | 0.22 |
| 🟨 MIN | 0.02 | 0.00 | 0.02 | 0.17 |
| 🟦 MAX | 0.59 | 0.23 | 0.39 | 0.83 |

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

In each of the stability tests the camera captured images at 30 frames per second. The above stability results for position are very encouraging. The average fluctuation in position is only 0.43 mm with a standard deviation of 0.22 mm. This indicates that the object's calculated position is very stable and does not on average vary more than 0.65 mm, although the occasional outlier does occur. These readings exclude extreme errors. The graph in figure 5.5 illustrates where extreme readings occured in the 10 different trials, in which for each trial, the object is held in place for 90 frames. Most of the extreme errors only occur during the 8th reading. This extreme jump in stability occurs due to the tracker loosing the object for a split second. The more minor errors possibly occurred due to mismatched points.

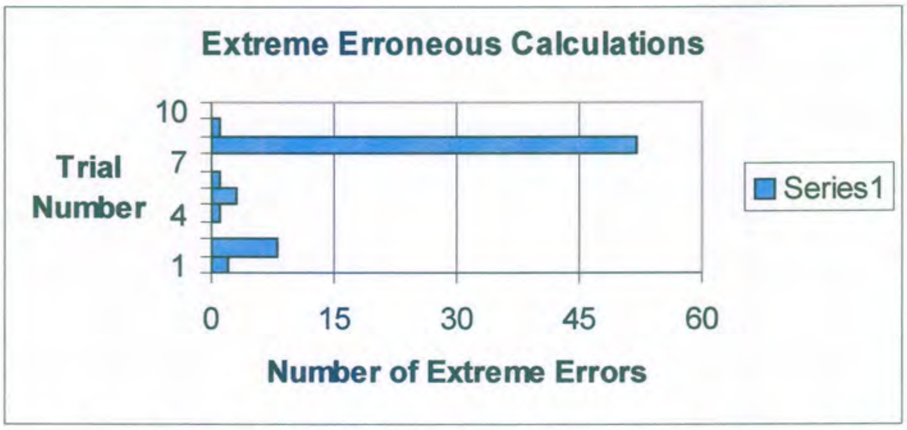
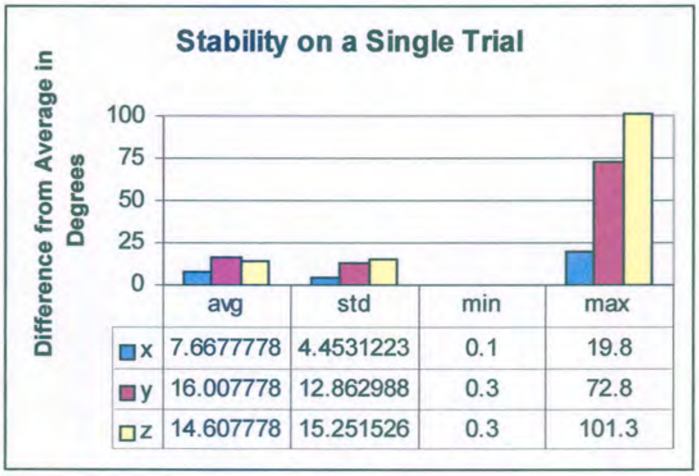


**Figure 5.5 Graph Illustrating the Number of Extreme Erroneous Results**

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



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

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

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

### 5.1.4 Interaction Volume

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

## 5.2 SPEED

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

## 5.3 PERFORMANCE OF THE COMPUTER VISION ALGORITHMS

### 5.3.1 Efficiency of the Algorithms

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

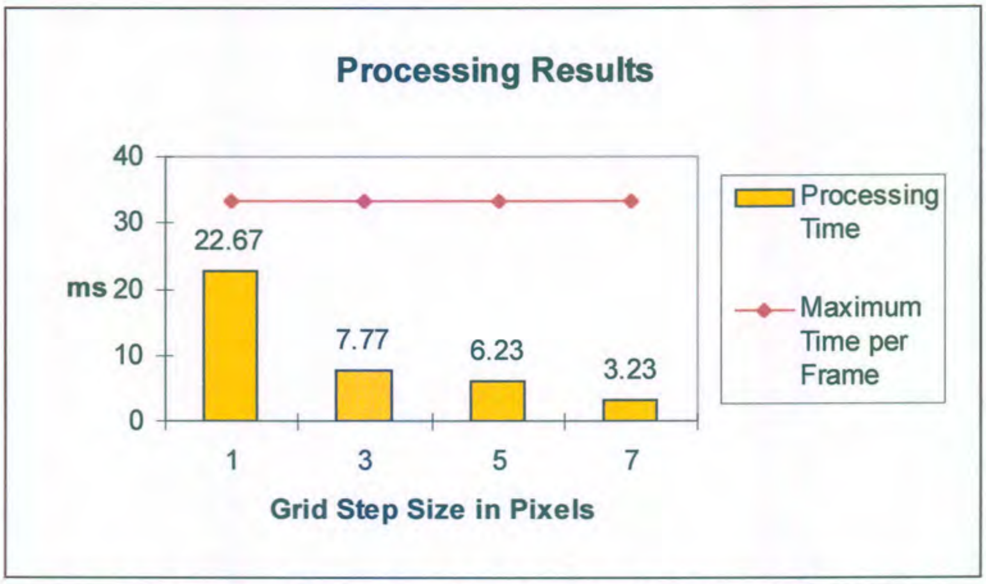


Figure 5.6 Average Processing Time for the Reflections Algorithm

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

Table 5.6 Average Processing Time for the Reflections Algorithm

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

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

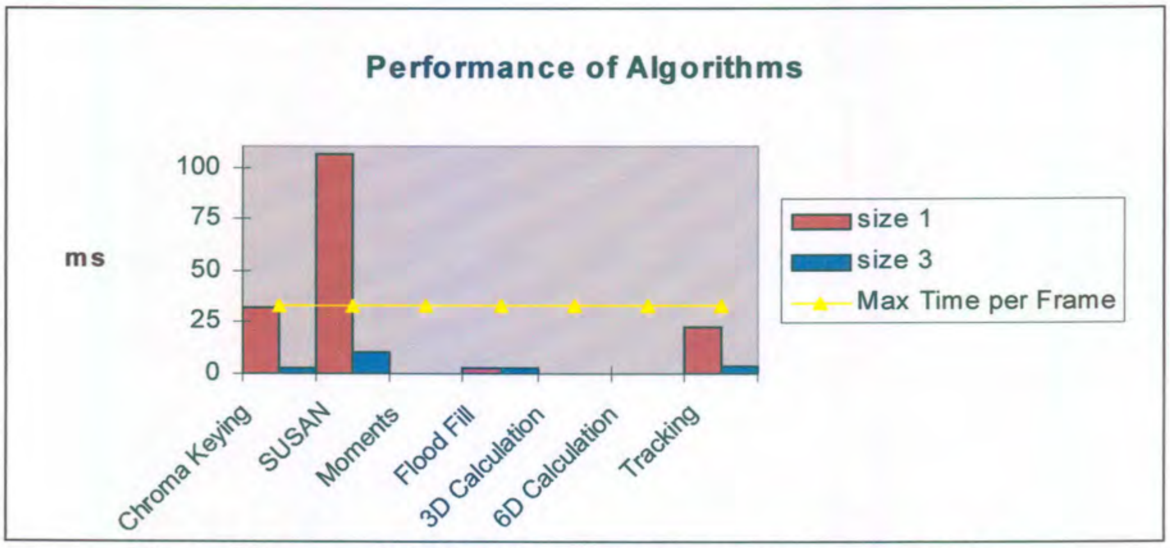


**Figure 5.7 Processing Time for the Different Components of the System**

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

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

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

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

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

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

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

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

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

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

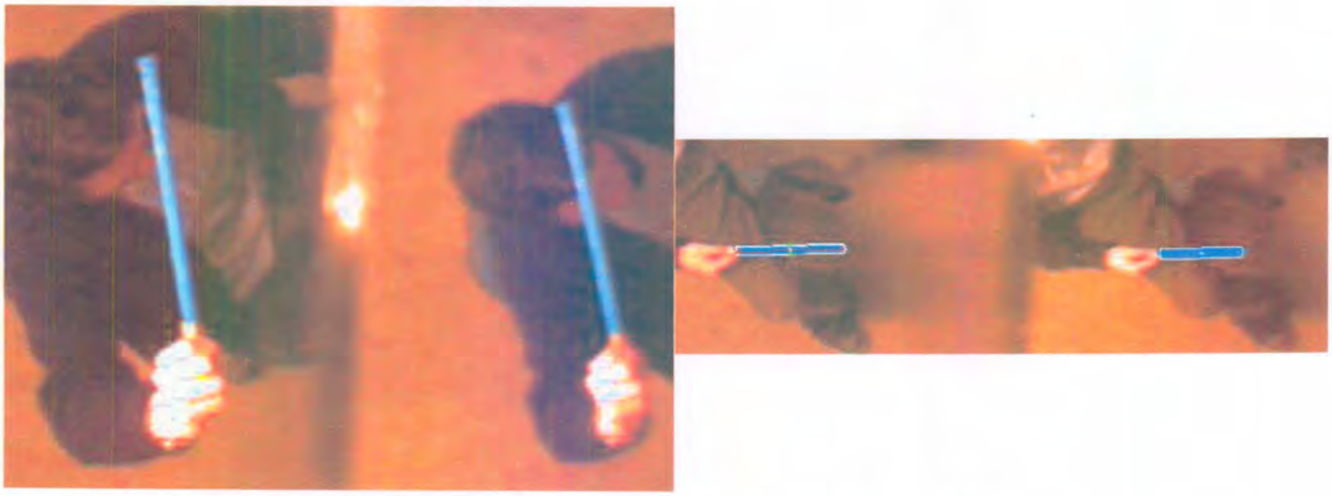


**Figure 5.8 Two Different Views of a Drumstick**

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

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

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

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

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

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

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

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

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

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

### 5.3.2 Effectiveness of the Algorithms

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

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

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

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

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

**Figure 5.11 SUSAN Applied to Image**

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

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

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

**Figure 5.13 Calculation of Image Moments**

## 5.4 TRACKER PERFORMANCE

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

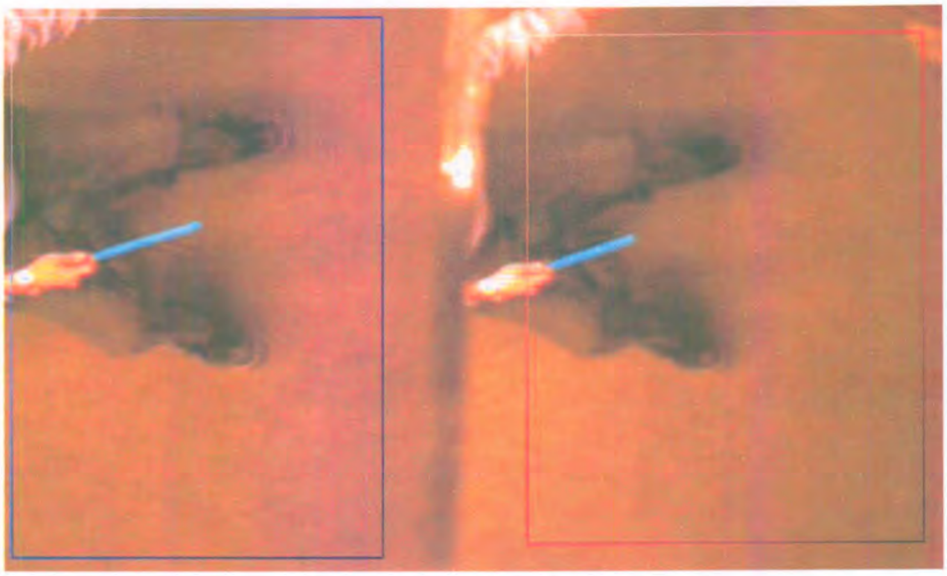


Figure 5.14 Region of Interaction in which Tracking is Performed

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

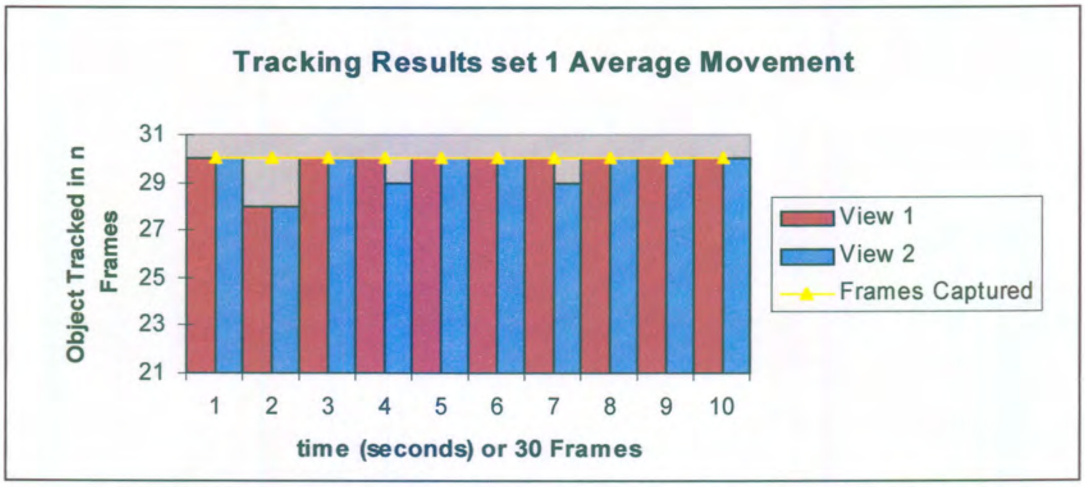


Figure 5.15 Number of Frames Object Tracked in Over 10 Seconds

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

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

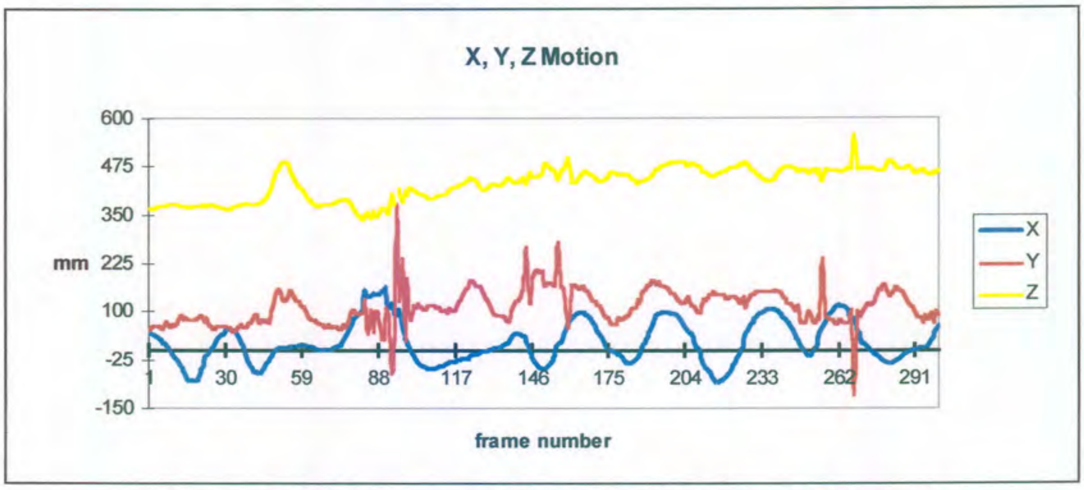
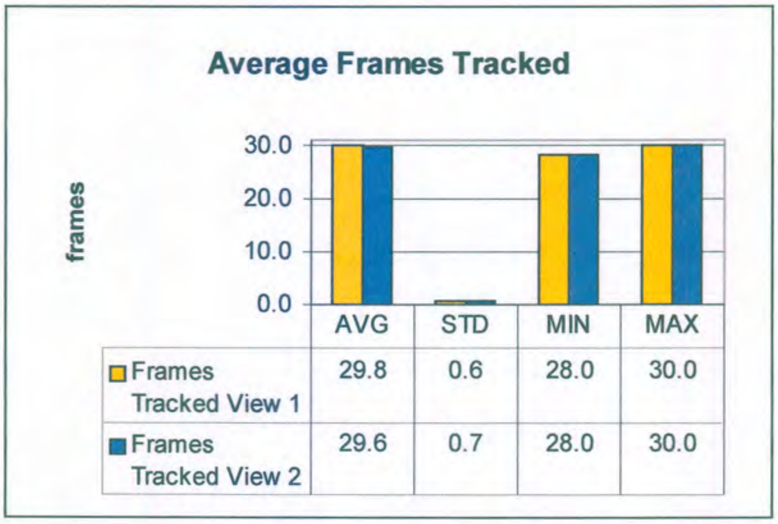


**Figure 5.16 Movement of the Drumstick Over 10 Seconds**

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



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

**Table 5.12 Tracking Results**

The drumstick is tracked in the left view (view 1) at an average of 29.8 frames per second over the 10 seconds. The tracker in the right view keeps track of the object on average 29.6 frames per second for the 10 seconds. The standard deviation is small and illustrates the effectiveness of the tracker. From this test it is clear that the tracker performs exceptionally well. The movement in this test is moderate. The next set of results illustrate tracking of faster movement. This movement is illustrated in figure 5.17.

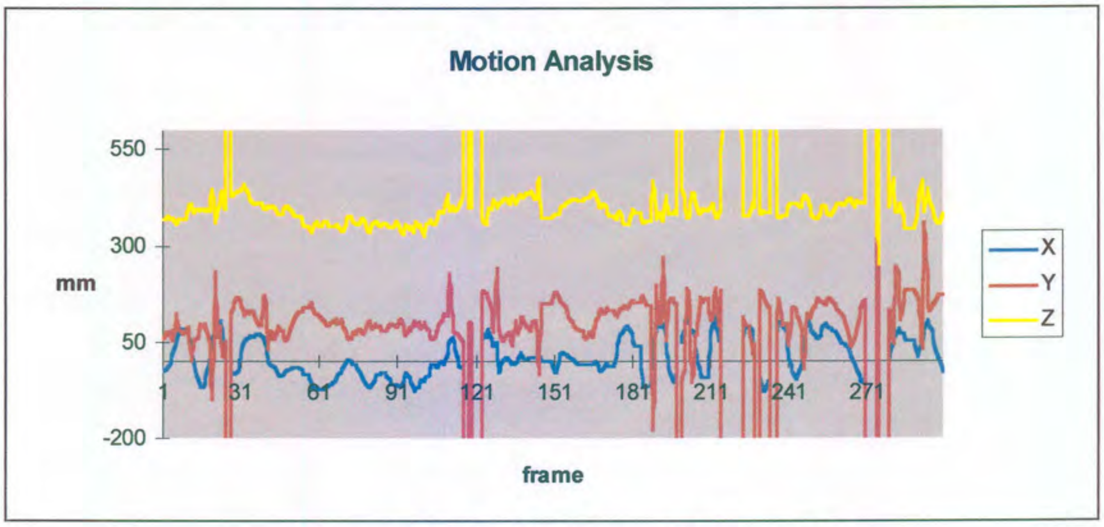


**Figure 5.17 Rapid Motion of Drumstick**

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

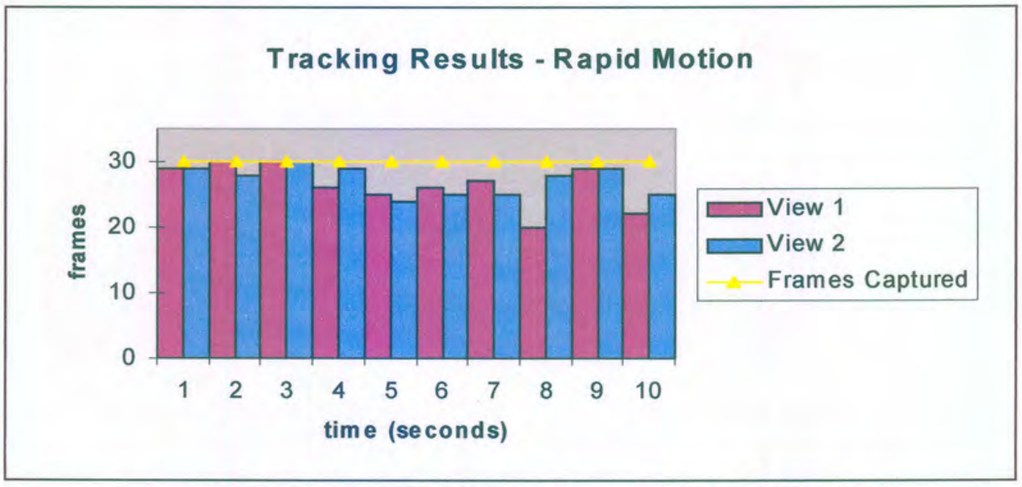


**Figure 5.18 Number of Frames for which the Drumstick is Tracked in the Different Views**

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

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

**Table 5.13 Tracking Results for Faster Movement**

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

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

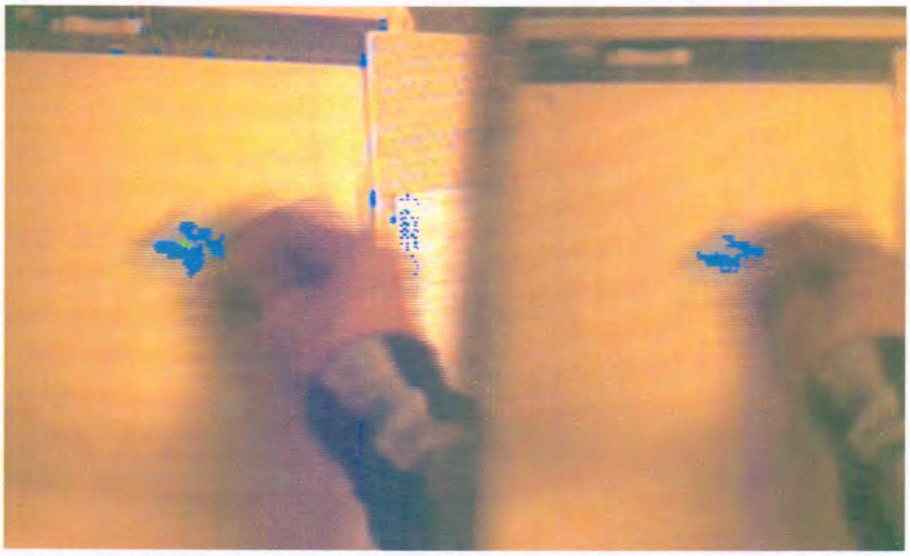


**Figure 5.19 Effects of Rapid Movement on the SUSAN Algorithm**

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

## 5.5 SUMMARY

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

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

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

# Chapter 6

# Conclusions And Future Work

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

## 6.1 CONCLUSIONS

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

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

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

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

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

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

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

In this thesis an approach called Reflections is developed for implementing multidimensional interaction by using catadioptric stereo. Furthermore Reflections is suitable for natural and non-intrusive interaction. The merit of using this approach to implement such interaction in 3D computer graphics and virtual reality is seen in the practical implementation of the Virtual Drums project and Ndebele Painting. The Virtual Drums project illustrates the use of the method for implementing natural and non-intrusive 5D interaction by allowing a user to play a virtual drum kit with real drum sticks. This is achieved by monitoring the position of a blue drumstick in 3D in a desktop environment. The project is also capable of tracking a light stick in low light environments. The Ndebele Painting application makes use of Reflections for natural non-intrusive 3D interaction in the CyberStage. Reflections may be implemented in a variety of different virtual environments ranging from a desktop monitor to a large projection based display like the CAVE.

## 6.2 FUTURE WORK

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

The latency of the system needs to be closely investigated because of the adverse effects latency has on presence in virtual reality. Predictive tracking presents a means for overcoming latency and for this reason should be integrated into the method. Implementing predictive tracking could also be used to enhance the speed and stability of the tracker. The system can be optimized to perform optimally on the platform it is currently running on. Implementing the system on a faster processor and making use of a digital or video camera with a higher frame rate will also improve the speed of the tracker, e.g. the speed of the tracker may be increased to 1000Hz by using the Artificial Retina Chip [Freeman et al]. The use of rectified catadioptric stereo requires further investigation as it presents yet another means of improving the speed of the tracker and achieving greater stability and accuracy, especially for point matching.

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

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

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

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