3D Articulated Model from a Stereo Camera

November 6, 2006

Steve Sarjeant
srs67@student.canterbury.ac.nz
Department of Computer Science and Software Engineering
University of Canterbury, Christchurch, New Zealand

Supervisor: Dr. Richard Green
richard.green@canterbury.ac.nz

Abstract

The area of computer vision based motion capture and pose estimation has received much research interest. However, most proposed systems are tested in ideal conditions with heavily constricted movement and are thus less than ideal. Furthermore, hardware requirements are often very high with multiple cameras or magnetic sensors reducing the availability of practical solutions.

This paper presents a novel stereoscopic approach to estimating the three dimensional pose of an articulated model through the use of non-contact computer vision based motion capture. The system is robust in cluttered and dynamic environments, performing real-time three dimensional model generation. Algorithms are presented for fundamental steps in computer vision based motion capture: tracking, and pose estimation, with accurate results computed.

Acknowledgments

I would like to thank Dr. Richard Green for his guidance throughout the year.

Contents

List of Figures

Publications

Related work is in press for the Image and Vision Computing New Zealand (IVCNZ) conference, 2006[1].

[1] Full paper published in proceedings of this independently reviewed conference.

1 Introduction

This paper presents a stereoscopic approach to estimating the three dimensional pose of an articulated model. Computer vision based posture estimation is the process of observing human figure(s) from camera(s) and using the data to approximate the pose of the person. From this estimation a three dimensional body model can be constructed and aligned with the person's posture and movements. Furthermore, analysis of the movement can be performed collecting informative statistics.

Systems that have been developed thus far fall into two broad categories, contact and non-contact. In contact systems the user has various joint marker sensors[1] attached to their body in comparison to non-contact in which the user is without any sensors attached and can act freely and more naturally using computer vision based motion tracking. At this point in time, contact based systems have the benefit of being more accurate. However, they are cumbersome and time consuming to use and are limited in their applications.

A non-contact system generally consists of two distinct components; tracking and pose estimation. Tracking refers to the process of computing correct identification of the subject and possibly limbs between successive frames. Tracking algorithms first require the correct segmentation of the subject from the scene, and can be classified as high or low level tracking [15]. An example of low level tracking is that of edges, in comparison to high level tracking which could be of the head and feet for example. After tracking has been performed the process of estimating the pose and aligning it with the subject's body can be performed.

Systems developed thus far have problems dealing with self-occlusion, where one part of the body obstructs another part with regards to the camera's view angle. Furthermore, non-contact systems, whilst reasonably accurate, are still not as accurate as commercially available contact marker based tracking systems.

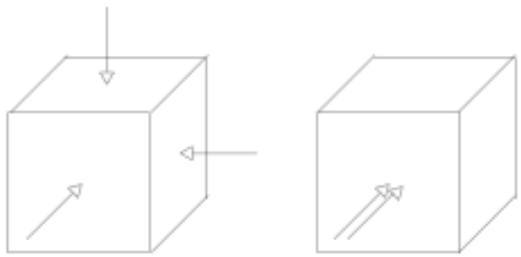

Figure 1.1: Left: A typical setup consisting of three cameras with associated viewing angles. Right: Proposed setup: a stereo camera with its two viewing angles illustrated.

Systems developed so far are often tested in ideal conditions, such as a highly constrained laboratory environment. Often multiple cameras are used from different viewing angles [16] (see Figure 1.1), use static backgrounds, lighting with no shadows, subjects wearing tight fitting clothes and limited movement (mostly gait and frontal posing). Such constraints cause these systems to be fairly limited in their practical use beyond the research laboratory.

[1] A list of commercially available trackers has been compiled at: `http://www.hitl.washington.edu/projects/knowledge_base/virtual-worlds/tracker-faq.html`, November 2006.

The benefits of creating a reliable and robust system that can generate and align a three dimensional articulated model from a stereo camera are such that the technology will be less cumbersome, reduce the burden placed on users and allow more real world users to take advantage of the technology.

To solve the previously mentioned problems I propose a system which will use one stereo camera facing the subject (see Figure 1.1). This configuration will allow the system to be used in less constrained environments. An alternative to using a stereo camera would be to use a single camera facing the subject, which still allows the system to be of practical use, but has limitations overcoming poor background segmentation and depth of field estimates.

This research is proposing a hierarchical structure for three dimensional articulated model generation. The first hierarchical level is a single stick figure of the subject, progressing to higher levels of complexity, modeling more parts of the body such as head and feet. This hierarchical method would allow the system to degrade gracefully if it detected too much noise in the environment for example.

The models (excluding the first level model) are three dimensional articulated models. An articulated model consists of a series of bones connected via transformable joints. The human body is an example of an articulated object, Figure 1.2 shows an example of an articulated model. Furthermore, the system will have the ability to collect and display informative data such as bounding boxes, speed and acceleration.

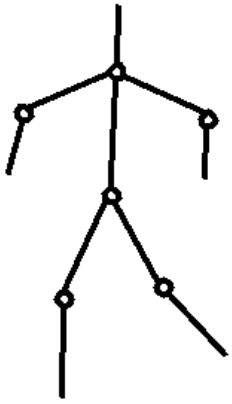

Figure 1.2: An articulated model.

2 Background

2.1 Stereo Vision

A binocular stereo vision system is where two cameras are used, whereby each camera views the scene independently with some area of overlap. This has the advantage over monocular vision, where a single camera is used, in that it can perceive the depth of objects and from this generate three dimensional real-world coordinate data.

Stereo reconstruction, the process of reconstructing the scene in three dimensions, relies on the correct functioning of stereo processing algorithms. Stereo processing algorithms serve the purpose of locating corresponding pixels between images, which are termed conjugate pairs. This is termed the correspondence problem. With many thousands of pixels in each image, stereo processing is a challenging task but well researched.

The most simple case of stereo vision is binocular stereo vision in which the cameras are only displaced horizontally. If this is the case, the stereo processing algorithms can make use of what is termed the Epipolar constraint. The Epipolar constraint states that pixel in the left image is situated on the same horizontal line in the right image. This constraint then reduces the pixel search space required by the processing algorithms.

When the images from each camera are superimposed, the distance between conjugate pairs is termed the disparity. From this disparity it is then possible to compute depth information. The equations for computing depth are defined in [11]. From Figure 2.1 (adapted from [11]), the point P is projected to the points p_l and p_r in each image plane. If the origin of the coordinate system is assumed to be the left lens centre though comparing the triangles PMC_l and p_lLC_l the following is computed:

$$\frac{x}{z} = \frac{x_l}{f} \tag{2.1}$$

Conversely, through comparing the triangles PNC_r and p_rRC_r the following is computed:

$$\frac{x-b}{z} = \frac{x_r}{f} \tag{2.2}$$

When combining the two above equations the depth can be computed via:

$$z = \frac{bf}{(x_l - x_r)} \tag{2.3}$$

2.1.1 Stereo Matching Algorithms

Stereo correspondence can be achieved through the use of edge matching and region correlation. This process performs most accurately in a scene with highly irregular texture.

Correlation Based Methods

Correlation based methods use image windows of a fixed size to test for similarity. A window is taken from the first image, and compared against a series of windows from the second image. The window from the second image that produces the maximum correlation is considered the corresponding window. Two

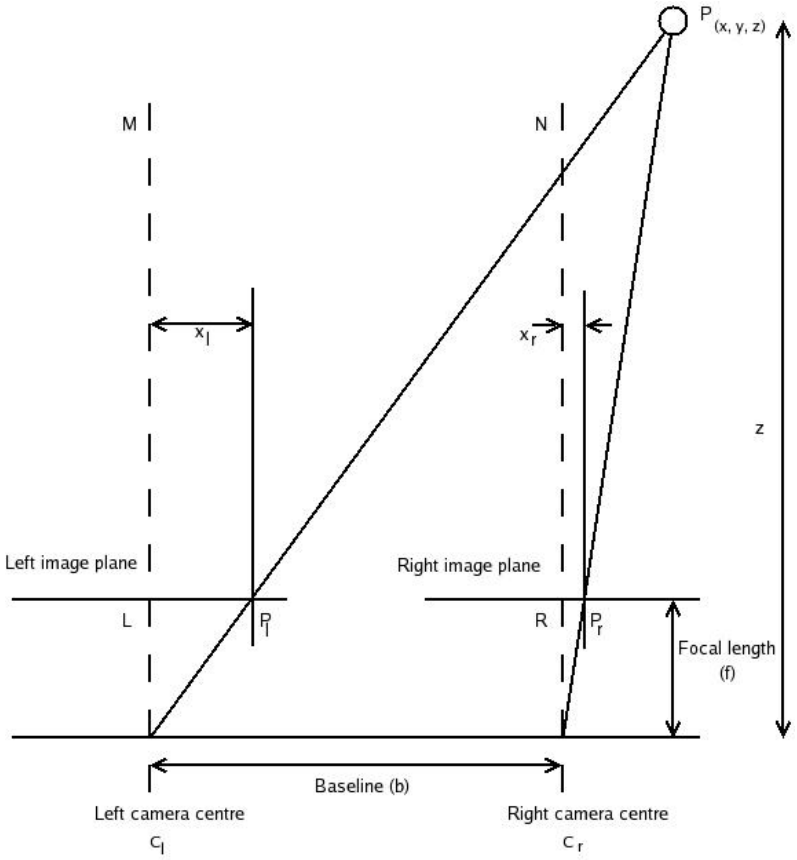

Figure 2.1: Stereo vision diagram.

correlation based methods are the Sum of Absolute Difference[1] (SAD) and the Sum of Squared Differences (SSD) [3]. The SAD correlation method as defined as:

$$\sum_{(u,v)\in W} |I_1(u,v) - I_2(x+u, y+v)| \tag{2.4}$$

While SSD is considered as:

$$\sum_{(u,v)\in W} (I_1(u,v) - I_2(x+u, y+v))^2 \tag{2.5}$$

Where I_1 and I_2 are the image windows.

Feature Based Methods

Feature based methods use a set of sparse features to constrain the correspondence search. Features of an image that may be used include edge points, lines and corners. Unlike correlation based methods, instead of image windows, the numerical and symbolic properties of feature descriptors are used.

2.2 Computer Vision Based Motion Capture

The functional structure of a computer vision based motion capture system as defined by [15], is shown in Figure 2.2. It is divided into four dependent components; initilisation, tracking, pose estimation and recognition. Initilisation generally refers to system setup such as camera calibration, scene adaptation or

[1] Used by the Point Grey Research Bumblebee stereo camera.

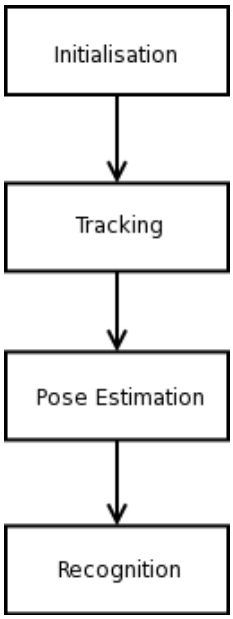

Figure 2.2: Steps in computer vision based motion capture.

model initilisation. Tracking refers to both segmenting the human from the image data, and/or keeping correspondence between points of interest through successive frames. The final stage, recognition, means to determine what a user is doing, for example sign language [20]. Recent tracking and pose estimation research is outlined below.

2.2.1 Tracking

Segmenting an image is the process of identifying regions of similar properties, for example, regions of similar texture or new objects in a scene. Segmentation as a subset of tracking, is thus vital for being able to interpret what is in the scene. Examples of high level tracking are discussed below.

Background Subtraction

Background subtraction is perhaps the most simple method to segment an image for motion. Unfortunately it is also the least robust of the presented methods. The process involves first taking a reference or *background* frame of the scene. This background frame is then subtracted from the current frame to detect any differences. As the background frame is static, this method cannot cope with modifications of a scene such as illumination changes. As no two frames are exactly the same, to cope with noise, the two images are subtracted, and only those pixels whose value is above a specified threshold are considered areas of movement. Functionally, this is defined as:

$$|frame_n - background| > threshold \tag{2.6}$$

Adjacent Frame Difference

Adjacent frame difference, discussed in [22], is similar to the background subtraction method. Two images are subtracted and those pixels whose resulting value is above a predefined threshold are considered areas of movement. The two methods only differ in that instead of using a static background frame, adjacent frame difference compares the current frame with the prior frame. This method is more robust as it can recover from dynamic changes in the scene. However, it has failings that slow moving objects in the scene can lose part of their figure or fast moving objects can gain parts of the background. Furthermore, a fast moving object within the scene can *double up*, that is, appear twice in the segmented image. Consider if

a human was on one side of the frame, and then before the next frame was captured by the camera had already moved to the other side of the scene, the subtraction process would then incorrectly identify both regions as movement. Similarly, adjacent frame difference is defined as:

$$|frame_n - frame_{n-1}| > threshold \tag{2.7}$$

Double Frame Difference

Double frame difference [27] builds on adjacent frame difference and is an improvement over both the previously defined methods. The double frame difference algorithm works by maintaining two adjacent frame difference images, one that is the result of the frames n_{-2} and n_{-1} and the other n_{-1} and n (where n is the current frame). The result of the intersection of these two adjacent frame difference images then forms the final segmented image. Whilst this method is computationally more expensive, it removes the associated problem of objects appearing twice that can happen in adjacent frame difference and is adaptive to dynamic scene changes unlike background subtraction. Double frame difference is then defined as:

$$d_{n-1} \cap d_n \tag{2.8}$$

Where d_x is the double frame difference resultant for frame x and $x-1$.

Adaptive Background Models

An example of an adaptive background segmentation technique is to model each pixel's distribution as a mixture of Gaussians (MOGS). Pixels are determined as background or foreground via the persistence and variance of each mixture. MOGS have advantages that they are adaptive, time-efficient and robust. However, they have difficulties with slow or large moving objects, shadows, and segmented regions may contain holes [28]. An example from [21], illustrated below, computes the probability of observing the current pixel as:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}\Sigma_{i,t}) \tag{2.9}$$

Where K is the number of distributions, $\omega_{i,t}$ a weight estimation (what portion of the data is attributed to this Gaussian), the i^{th} mixture, at time t, $\mu_{i,t}$ the mean value and $\Sigma_{i,t}$ the covariance matrix.

Chroma-Key

Chroma-Key algorithms segment the image based on colour regions [19]. This can be completed by having either a static colour background, such as a green-screen, or requiring the user to wear clothing of a specific colour that is not located within the scene. Chroma-Key algorithms are relatively simple. However, they have the disadvantage that they place constraints on the scene, or what the user is required to wear. Furthermore, the chroma-key method is not dynamic to scene changes. For example, if a user were to walk into the scene and place an object down, the object would always appear in the segmented image.

Tracking Over Time

Tracking over time algorithms serve the purpose of estimating points of interest in successive frames. This is advantageous when constructing a human body model. Consider if the process were tracking the position of the foot, and the foot became occluded behind the opposite leg for several frames. Without tracking, the model generation algorithms will fail. However with tracking, the algorithm can estimate the position of the foot before it reappears.

Tracking over time algorithms generally have a recursive process. There are constant predictions being made ahead of time and measurement update corrections adjusting the predictions by actual measurements as shown in Figure 2.3 from [23]. Several mathematical models exist for predicting successive locations such as the Kalman and Particle (Condensation) Filters which are outlined below.

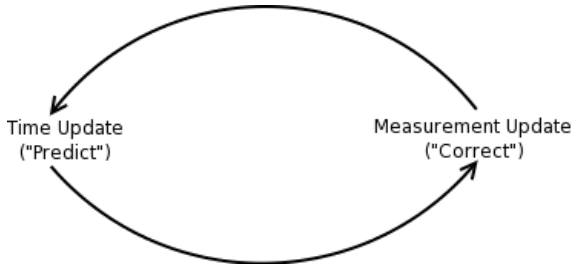

Figure 2.3: Tracking over time algorithms.

Kalman Filter

The Kalman Filter relies on a recursive set of equations to estimate the state of a process whilst minimising the mean of the squared error. The filter is uni-modal and based on the Gaussian distribution. The disadvantage of the filter is that it only predicts a single position.

The time update equations for a form of the discrete Kalman Filter can be defined as [23]:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (2.10)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2.11)$$

Whilst the measurement update equations can be defined as:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (2.12)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (2.13)$$

$$P_k = (I - K_k H)P_k^- \quad (2.14)$$

Where $\hat{x}_k^-$ is the *priori* estimate, $\hat{x}_k$ is the *posterior* estimate, z_k is the difference between the actual measurement, Q is the process noise covariance, R is the measurement noise covariance, P_k^i the priori estimate error covariance, P_k the posterior estimate error covariance. A, B and H are matrices, A relating the state at previous time step $k-1$ to the current state k, B relating the optional control input to the state x, H relating the state to the measurement z_k.

Particle Filter

The Particle Filter [9], unlike the Kalman Filter, is non-Gaussian, and rather than being a single estimate of position and covariance, is an entire probability distribution. This makes it robust to heavy image clutter. Furthermore, the Particle Filter is a simple algorithm when compared to the Kalman Filter.

2.2.2 Pose Estimation

Pose estimation as defined by [15], is the process of identifying how a human body and/or individual limbs are configured within a scene. The paper divided pose estimation processes into three categories: model free, indirect model use, and direct model use. Model free is when no a priori model is used. Indirect model use and direct model use both utilise a priori model in the process of determining the configuration of the human. In indirect model use the model is used to restrict and interpret the observed data, whilst in direct model use, the model is maintained according to the observed data.

Model Free

A model free approach is that proposed by [25]. This system uses a multi-class statistical model to find colour blobs and shape to find the position of the hands and head. The system first requires an empty scene for which it builds a colour background model. It then detects regions of change and uses two dimensional

contour shape analysis to find points of interest such as the head. After this, blobs corresponding to the points of interest are created and tracked. Whilst the system proved to be quite robust, it expects a fairly static background. Furthermore, it could only handle a single human within the scene. If this condition breaks down the system will try to model the two moving humans as a single human.

Similarly, [14] uses blobs to model the head and hand of a human, which are found through detecting human flesh coloured regions. The system uses kinematic constraints to determine where the elbow joint is located and creates a silhouette of this structure for which it compares with the silhouette of the human created via background subtraction. This system uses two cameras to then generate three dimensional data for the found points. Again, the system is accurate, however, it assumes a frontal pose with regards to the camera and that the head position is fixed with regards to the shoulder. Furthermore, the use of background subtraction may make the process fail if dynamic scene modifications were encountered.

A stick figure representation is created in [10] using distance transformations. The system uses thermal imaging to construct a silhouette of the human, for which it then uses a heuristic contour analysis method to determine several points of interest. The thermal imaging used to detect the human is said to be robust in difficult background and illumination conditions. However, this system had the constraint that the human remained stationary with regards to the depth from the camera. This constraint allowed the system to infer three dimensional positions, however it is significant constraint.

Indirect Model

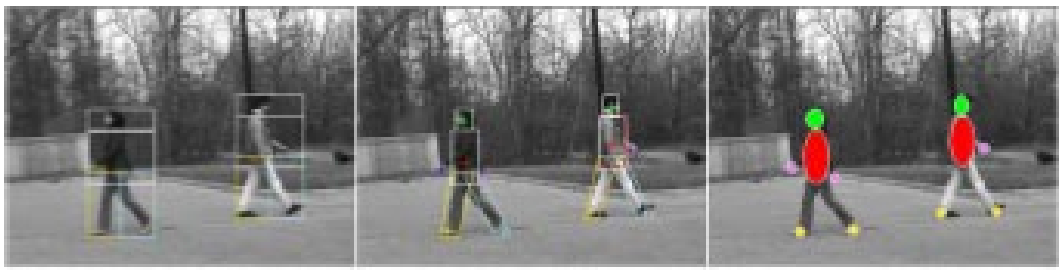

Figure 2.4: Cardboard model segmentation.

Indirect model methods use an a priori body model beforehand to determine the pose of the human. An example of this is using a Cardboard Model [12] which contains the relative positions and sizes of body parts to determine the pose of the human. An example of cardboard model use is shown in Figure 2.4 from [8]. The left most image shows the segmentation of the detected region to determine posture.

A further example of indirect model is through the use of Principal Component Analysis (PCA) as shown in [4]. Here again the model is constructed through the use of geometric constraints on the length ratio of limbs. The system further uses heuristic methods to identify significant points.

Direct Model

Direct model use is the case where an a priori model is used and is maintained continually to correspond to the updated human data. An example is shown by [17] which uses four cameras and a series of equations to map three dimensional homogeneous image points to four dimensional homogeneous model data. The system is shown to be very accurate at modeling the upper body. However processing time is low at two frames per second and requires four cameras.

Work by [26] (shown in Figure 2.5) uses colour blobs to find the hands and head, for which it then uses a recursive estimation framework to model the upper body. The system encodes kinematic constraints and uses the Extended Kalman Filter for low-level tracking. Similarly, this research proved to be accurate at modeling the upper body.

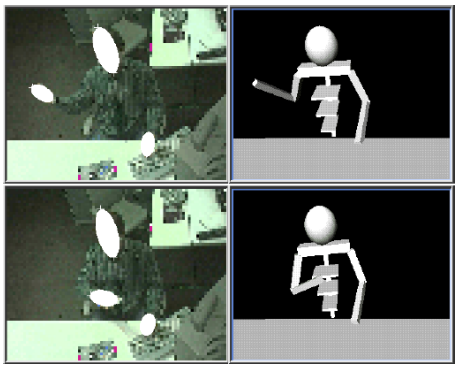

Figure 2.5: Colour blob tracking.

A further example is that by [18], here a graph represents the human body. This research uses occlusion-sensitive local likelihoods that approximate the global likelihood which accounts for occlusion in the image. The research works well at constructing a whole-body model that deals with occlusions but no data is provided on the speed required for the method.

The research by [29] is of particular interest as they use a stereo camera to estimate the upper body and limbs (shown in Figure 2.6). The system utilises a method based on the iterative closest point registration algorithm (ICP) integrated with an unscented Kalman Filter. Processing time for this method is below real-time at approximately 1-2 seconds per frame and uses four stereo cameras.

Figure 2.6: Upper body estimation using stereo vision.

3 Design & Implementation

3.1 Overview

Libraries

The following libraries with their associated descriptions aided in implementation:

Triclops/Digiclops: The Triclops and Digiclops libraries are provided by Point Grey Research for use with their stereo cameras. They provide facilities to retrieve images and perform rectification and stereo image processing through a `C` programming interface. Processing parameters like correlation mask size, transformation matrix, and others are fully configurable.

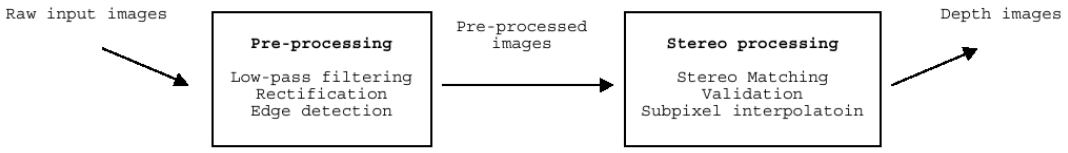

Figure 3.1: The workflow of the Digiclops/Triclops API.

Figure 3.1 from [1] shows the workflow of the Digiclops/Triclops libraries. Below is a description of the key steps.

1. Raw images are acquired via the Digiclops library and prepared for use in the Triclops library which provides the functionality for the remaining steps.
2. Pre-processing is carried out which allows the specification of resolution and low-pass filtering.
3. Low-pass filtering smooths the images before rectification, without which the rectified images will exhibit aliasing.
4. Rectification corrects the input images for distortions in the camera lenses. These can cause straight lines in the scene to appear curved, particularly near edges. The rectification process further causes the rows of the input images to be aligned for horizontally displaced cameras, which allows the stereo processing algorithm to make use of the epipolar constraint.
5. Edge detection, an optional feature, causes stereo matching on the differences in brightness rather than absolute values. This is desirable as the cameras have auto gain functionality and the auto gains between cameras may not match causing issues to occur in stereo processing.
6. Stereo processing is carried out via the Sum of Absolute Differences correlation based method. The various stereo processing parameters are discussed in Section 3.3.

OpenCV: The Open Source Computer Vision library by Intel provides hundreds of many common image processing algorithms, such as contour processing and image statistics, through a `C` programming interface. OpenCV also performs GUI operations, from displaying two dimensional images to accepting user input. Images are retrieved via the Triclops/Digiclops API and converted to standard OpenCV data formats for subsequent processing.

OpenGL: The Open Graphics Library provides facilities for three dimensional rendering and user input through a `C` programming interface. In this system it is used to render the articulated human model in a three dimensional world.

Hardware

Figure 3.2: The Bumblebee stereo camera by Point Grey Research.

The Bumblebee stereo vision camera (Figure 3.2) developed by Point Grey Research was used in this implementation. The Bumblebee features two Sony Charge Coupled Devices (CCD) displaced by a baseline of 12 centimeters. It can retrieve 640x480 resolution images at 30 Hz, and 1024x768 resolution images at 15 Hz. However after stereo processing, which is performed on the host PC, the frame rate at 640x480 would drop to approximately 15 frames per second (FPS). At a resolution of 320x240 with stereo processing, frames could be retrieved at approximately 30 FPS, so to keep performance as close as possible to real-time this resolution was utilised. The long and short range accuracy of the Bumblebee stereo camera is shown in Figure A.1.

The system was implemented in `C` and tested on an Intel Pentium IV 2.4Ghz personal computer with 512MB of main memory, the Bumblebee interfaced with the PC via a IEEE-1394 (FireWire) connection.

3.2 Segmentation

In order to segment the detected area of motion disparity image filtering was used in conjunction with double frame differencing. This is a four step process outlined below:

1. Perform the double differencing algorithm.
2. Calculate the centre of mass of the segmented region.
3. Calculate the disparity for the centre of mass.
4. Filter the disparity map based on this disparity value. To ensure the whole body is segmented correctly, for which there could be a range of disparity values, the filtering process includes values that are within a specified range of the disparity value.
5. Compute the centre of mass for disparity filtered segmented region.

Double frame difference was chosen as the basis of the disparity filtering algorithm as it is a simple method that is robust to dynamic scene changes. There are several advantages to disparity filtering over sole double frame difference. The stereo processing algorithms generally will not confuse areas of different depth, and thus a moving object will not be attached to part of the background as often happens with double frame differencing.

Furthermore, when using double frame difference, it is a requirement that the system implement a thresholding scheme, where the current human model is only modified if sufficient movement is detected, or remembering the current configuration of the model across frames. Consider if a full body model had been constructed in the previous frame, then in the current frame the human only makes a small movement, such as moving their arm. If the system records this movement it must compute that it is just the arm moving, and adjust the model accordingly, or implement the first approach and disregard the movement.

As the disparity filtering algorithm always segments the whole human body this issue is negated. See 4.1 for an example of disparity filtering.

3.3 Three Dimensional Data

Disparity Map

A disparity map created from the disparity values returned by the Bumblebee stereo camera is shown in Figure 3.3, the depth of the pixel determines its colour, with white being nearest, to black for which no valid disparity value could be determined. The Bumblebee's stereo matching process allows several key parameters to be modified which are outlined below.

Disparity Range: The disparity range is defined as the range of pixels that the stereo algorithm will search to determine the best match. The minimum disparity is how far away an object can be viewed, 0, for an infinitely far away. Whilst the maximum disparity determines how close an object can be.

Correlation Mask: The correlation mask is the window size that is used for stereo correlation. Larger masks produce disparity data that is more dense and smooth, however precision is lost at regions of depth discontinuities. Smaller masks produce sparser and more noisy data, however perform better at regions of depth discontinuities. The correlation mask size can range from 1x1 to 15x15. In this implementation 9x9 is used as it produces the most desirable data.

Validation: In images occlusion can occur making it not possible to find a valid correspondence between the left and right images. To avoid the incorrect correspondence from being chosen, texture and uniqueness validation can be turned on. Texture validation determines whether correspondences are valid based on the amount of texture inside the correlation mask. Uniqueness validation determines whether the correspondence match is significantly better than any other correspondence, and if not, marks it invalid. Both are turned off in this implementation as it caused the returned data to be too sparse.

Surface Validation: Surface validation removes spike noise from the disparity data. The method segments the disparity data into regions of connected components, with components that are less than a specified size being removed. Surface validation is turned on with a size of 255 as it removes most of the spike noise that can be produced via correlation based stereo matching methods.

Figure 3.3: A disparity map calculated from the Bumblebee.

Individual Coordinates

To cope with the Bumblebee's inability to reliably determine the disparity of an individual pixel, a simple yet robust algorithm was implemented. Instead of checking for the disparity of the exact required location,

$$\begin{pmatrix} -0.01 & 0 & 0 & 0 \\ 0 & -0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 3.4: Bumblebee transformation matrix.

a region of interest (ROI) is constructed and surveyed around that pixel. This method surveys disparity pixels within the constructed square and then returns the average of the detected values. This algorithm relies on the notion that pixels of close proximity will be of similar properties.

Transformation Matrix

The Triclops/Digiclops libraries allow the transformation matrix to be modified by the user. This is useful as it is possible to specify the matrix such that the libraries automatically return three dimensional data that is compatible with OpenGL's coordinate system and can be rendered without performance degradation.

3.4 Model Generation

The model generation is constructed on a frame by frame basis through a series of constraints. The constraints are constructed such that they search for key body points where they would be under normal human movement through a scene.

Movement Direction

A movement direction is associated with the moving figure in the scene. There are 5 possible directions the moving object can take and are listed below along with their defining characteristic.

Stationary: Movement less than pre-specified value.

Front to back: Z value decreasing, X, Y relatively constant.

Back to front: Z value increasing, X, Y relatively constant.

Left to right: X value decreasing, Y, Z relatively constant.

Right to left: X value increasing, Y, Z relatively constant.

Where X, Y, Z are the real world coordinates of the centre of mass. When multiple movement directions are detected, front to back and left to right for example, the greater of the two is selected as the main movement.

Face Recognition

Face detection capabilities are provided by OpenCV via the use of Haar classifiers [24]. Haar classifiers are provided for frontal and profile poses. Haar classifiers perform object detection via Haar-like features, Haar-like features utilise the change in contrast between rectangular groups of pixels.

Centre of Mass

The centre of mass of a detected area of movement is the most robust measurement and forms the basis for all model generation algorithms including principal axis, upper/lower body segmentation, and background segmentation. The centre of mass is considered the waist of the human. The centre of mass is calculated from the first order moments of the segmented region of movement. It is as follows:

$$\bar{x}\sum_{i=1}^{n}\sum_{j=1}^{m}B[i,j]=\sum_{i=1}^{n}\sum_{j=1}^{m}jB[i,j] \tag{3.1}$$

$$\bar{y}\sum_{i=1}^{n}\sum_{j=1}^{m}B[i,j]=\sum_{i=1}^{n}\sum_{j=1}^{m}jB[i,j] \tag{3.2}$$

where $\bar{x}$ and $\bar{y}$ are the coordinates of the centre of mass.

Principal Axis

To correctly determine the principal axis of a shape it must be elongated, the principal axis is thus considered the axis of least inertia. From [11], [13], the principal axis is described as:

$$\tan^2\theta+\frac{\mu_{20}-\mu_{02}}{\mu_{11}}\tan\theta-1=0 \tag{3.3}$$

Where the second order moments μ_{20}, μ_{11} and μ_{02} are considered as:

$$\mu_{20} = \sum_{i=1}^{n}\sum_{j=1}^{m}B[i,j](x_{ij}-\bar{x})^2 \tag{3.4}$$

$$\mu_{11} = \sum_{i=1}^{n}\sum_{j=1}^{m}B[i,j](x_{ij}-\bar{x})(y_{ij}-\bar{y}) \tag{3.5}$$

$$\mu_{02} = \sum_{i=1}^{n}\sum_{j=1}^{m}B[i,j](y_{ij}-\bar{y})^2 \tag{3.6}$$

Solving equation 3.3 yields:

$$\theta=\frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}\right) \tag{3.7}$$

Head Position

The upper body is segmented from the lower by using the centre of mass as a reference point. Once this has been done, the principal axis of the upper body is calculated. The (x,y) position of the head is classified as the pixel with the highest y component that follows the angle of the principal axis originating from the centre of mass. The head position then has a pre-specified value subtracted from it to ensure the head position is as close to the centre of the real head as possible.

Neck Position

The neck is estimated using simple anatomical reasoning at 75% of the way between the centre of mass and the top of the head, although this parameter would need to be reconfigured for different body types.

Feet Position

The feet are assumed to be below the centre of mass, with a foot on either side of the centre of mass. The feet are calculated as the furtherest pixels from the centre of mass satisfying the defined two assumptions using the Euclidean distance measurement. The first foot is mathematically considered as:

$$\max_{x\in[0,\bar{x}],y\in[0,\bar{y}]}\sqrt{((x-\bar{x})^2+(y-\bar{y})^2)} \tag{3.8}$$

Similarly, the second foot is defined as:

$$\max_{x\in[\bar{x},w],y\in[0,\bar{y}]}\sqrt{((x-\bar{x})^2+(y-\bar{y})^2)} \tag{3.9}$$

Where w is the width of the image, and h the height.

To then determine which foot corresponds to the left and right legs, the detected motion direction and orientation of the moving object are taken into consideration.

Hip Positions

The hip positions are determined as the pixels furtherest away from the centre of mass on the left and right hand sides respectively, as this is where they are generally situated for an upright human. To ensure they are as close to the real hips as possible, pixels are scanned orthogonally to the principal axis and each hip has a pre-specified value added to it to ensure the hip position is inside the human silhouette.

Model Descriptions

Listed below are the generated models with associated descriptions of their estimated parameters. The are levels of models generated in increasing complexity. The motivation for doing so is that graceful degradation could be built into the system, for example, if too much noise was detected the system could estimate the simplest model instead of making an incorrect estimate. The models 4 and 5 take into consideration the movement direction associated with the segmented region and the results of face recognition. The justification is that the system can then estimate the orientation of the human and from here correctly the left and right legs or hips. For example, when walking parallel to the camera view plane, from right to left, the furtherest foot away would be considered the right foot.

Model Level 1: A single vertical bar. The centre of mass is first calculated along with the highest and lowest pixels, the end points of the bar are then $(\bar{x}, y_l)$ and $(\bar{x}, y_h)$ where y_l and y_h are the lowest and highest y pixels of the segmented area respectively. Shown in Figure 3.5.

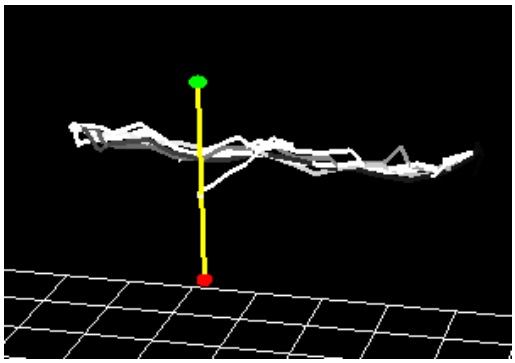

Figure 3.5: Model 1. Also shown is the past movement through the scene. The line interpolates from black (least recent movement) to white (most recent movement). The white grid is a rendered wire frame floor used for reference purposes.

Model Level 2: A single principal axis aligned bone, covering the extent of the segmented region. Shown in Figure 3.6.

Model Level 3: Two articulated bones. The upper bone is principal axis aligned, to the centre of mass. The second bone is constructed from the centre of mass, to the lowest point combined with the centre point of the lower body. Shown in Figure 3.7.

Model Level 4: Three articulated bones joined at the centre of mass. The upper bar is the same as in model 3. The other two bars are connected to the detected feet positions. Shown in Figure 3.8.

Model Level 5: Five articulated bones. The centre of mass is connected to the head and each hip. Each hip is connected to the corresponding foot. Shown in Figure 3.9.

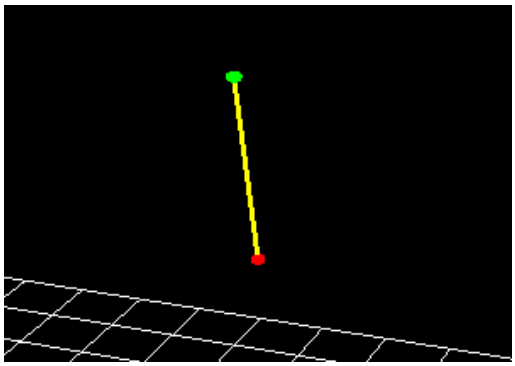

Figure 3.6: Model 2

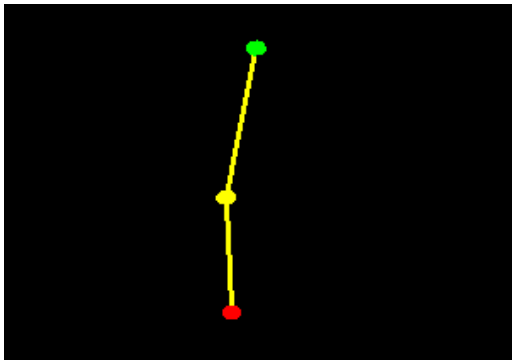

Figure 3.7: Model 3

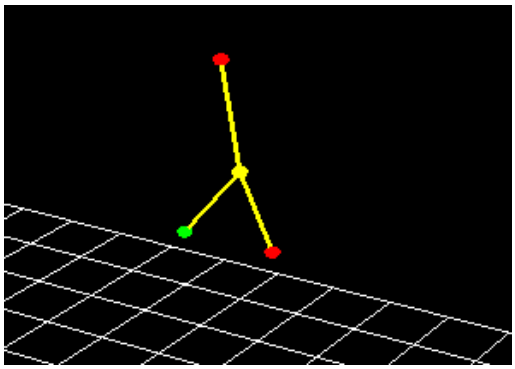

Figure 3.8: Model 4

3.5 Performance Measurements

Several performance measurements were also calculated to enable easy analysis by interested parties, such as a high performance sports coach. These measurements include two and three dimensional bounding boxes, cyclic motion, speed and acceleration.

Principal Axes

Further to the whole body principal axis constructed for model generation, upper and lower body principal axis are also constructed. The motivation for doing so as it would allow a viewer to more easily gain an understanding at what angles the body is configured at when performing a required action, such as jumping.

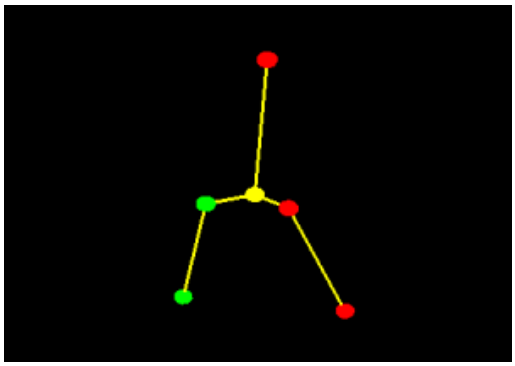

Figure 3.9: Model 5

Bounding Boxes

The system can construct and display a two dimensional bounding box around the detected area of movement. To produce the most accurate bounding box, it is aligned with the principal axis and, thus the accuracy of the bounding box depends on the accuracy of the principal axis. Generating an axis aligned bounding box involves calculating the maximum widths of the human at angles orthogonal to the principal axis. It also generates axis aligned bounding boxes for the upper and lower segments of the human as determined by the centre of mass. A three dimensional bounding box can also be constructed enclosing the human as shown in Figure 4.12.

Cyclic Motion

To determine the cyclic motion of a human as they walk in front of the camera the visible area of their lower body is calculated. The lower body is considered as anything below the centre of mass. Other methods to determine cyclic motion were also tested; the distance between each foot, and the principal axis of the lower body, but neither performed accurately. The calculation of cyclic motion allows a sports coach, for example, to more easily compute where the human is in their movement cycle. Mathematically, the cyclic motion, c, is represented as:

$$c = \sum_{i=\bar{x}}^{n} \sum_{j=\bar{y}}^{m} B[i, j] \tag{3.10}$$

Speed

With three dimensional data and the centre of mass located, calculating the speed, s, of the moving object is a trivial task. The Euclidean distance between the objects centre of mass position is calculated from the current frame i and the previous frame $i-5$. This distance then enables the speed of movement to be calculated as well as acceleration. In this implementation speed is updated every 5 frames of input images.

$$x_d = (x_i - x_{i-5})^2 \tag{3.11}$$
$$y_d = (y_i - y_{i-5})^2 \tag{3.12}$$
$$z_d = (z_i - z_{i-5})^2 \tag{3.13}$$
$$s = \frac{\sqrt{x_d + y_d + z_d}}{5} \tag{3.14}$$

Acceleration

Once speed is calculated, acceleration, a, is then considered as:

$$a = s_i - s_{i-5} \tag{3.15}$$

4 Results

4.1 Performance

The system could perform model generation, which involves tracking and pose estimation, in real-time. However, the process of estimating performance measurements would degrade the system to below real-time. This is because there are several more calculations that need to be made when constructing the principle axes of the upper and lower body for example.

4.2 Segmentation

An example of the segmentation is shown in 4.1. This method performed better than double frame difference by itself as the DDA algorithm would often lose parts of the body and gain parts of the background. For example, when walking towards the camera with a uniform coloured shirt, the stomach region of the user would not be segmented correctly. This would not happen with DDA and disparity filtering, as the disparity filtering only relies on the DDA algorithm correctly segmenting at least part of the moving object.

Figure 4.1: An example of disparity filtering.

4.3 Three Dimensional Data

Due to the heavy reliance the performance algorithms place on accurate three dimensional data this has been evaluated. The centre of mass is used as the pixel of interest in Figures 4.2, 4.3 and 4.4. Of a series of 335 frames captured the average approach was required to recover from the 12% of frames that the exact world coordinates of the point could not be determined.

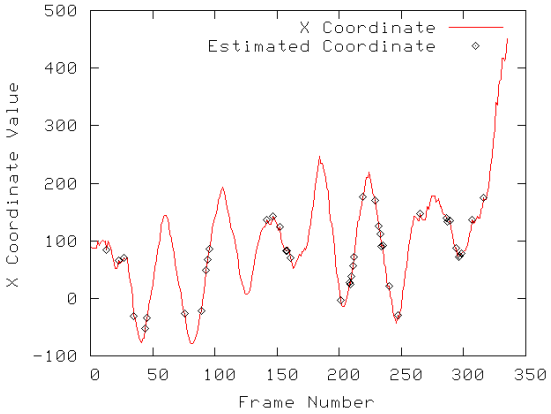

Figure 4.2: X coordinate of a moving object.

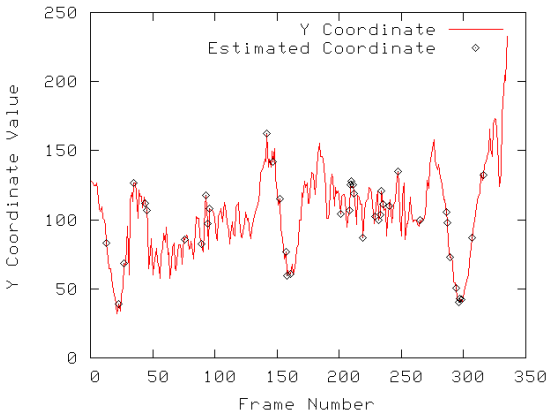

Figure 4.3: Y coordinates of a moving object.

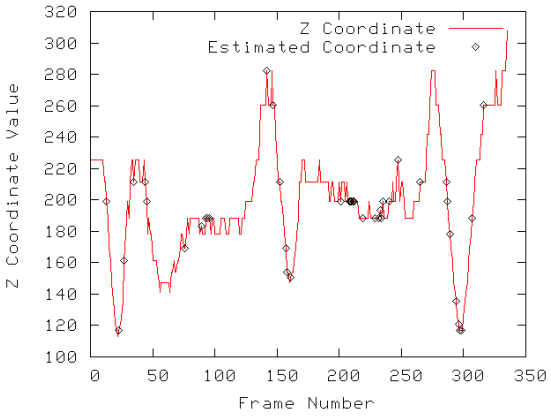

Figure 4.4: Z coordinates of a moving object.

4.4 Model Generation

Face Recognition

The face detection capabilities provided by OpenCV proved to be insufficiently robust. To accurately recognise a face the face must be sufficiently clear and large within the captured image. This is generally not the case when a user is moving through a scene as their face will often only take a minute area in the image. Due to this the face detection was removed from the process of model construction.

Movement Direction

The movement direction detection is accurate as it relies on the robust model generation parameter, centre of mass. However it would not aid in model generation when the user changed orientation without moving through the scene, for example, being stationary and turning around.

Significant Points

An evaluation on how well the process located significant points of the body was carried out. A single person walked through a scene naturally, capturing 100 frames of data. For each significant point a user would manually locate the point and the system would then calculate the Euclidean distance to the estimated location in pixels. These values were accumulated and their corresponding statistics calculated as shown in Table 4.1. The Figures 4.5, 4.6, 4.7, 4.8 and 4.9, show the pixel error rates for the significant points across the 100 captured frames. Graphs are on separate figures for clarity.

Significant Point	Average Error	Min. Error	Max. Error	Std. Dev.
Head	5.81	1.00	26.93	3.21
Neck	5.40	1.00	24.52	3.50
Waist	5.52	0.00	15.03	3.14
Hip 1 *	7.47	1.00	17.03	3.34
Hip 2 *	7.22	1.00	17.80	3.90
Foot 1 *	13.94	4.12	65.55	7.93
Foot 2 *	14.40	2.24	79.38	9.21
Left Foot	32.74	4.00	88.01	24.14
Right Foot	33.90	4.47	86.83	24.33

Table 4.1: Significant Point Error Statistics (in Pixels).
* Correct identification of positions ignoring whether correctly identified as left and right.

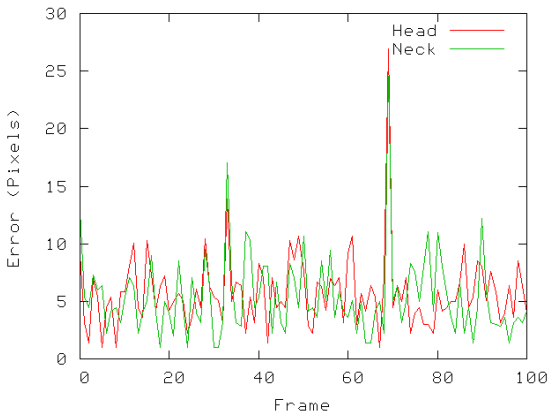

Figure 4.5: Head and neck coordinate pixel error.

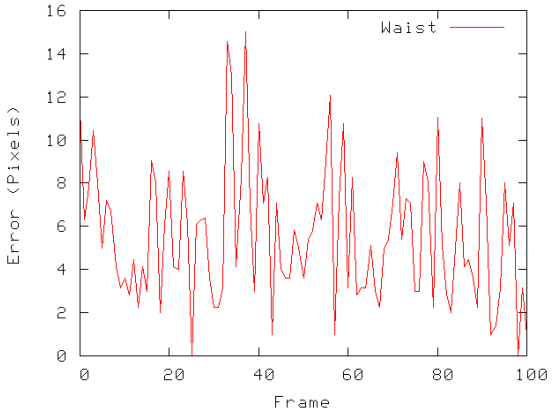

Figure 4.6: Waist coordinate pixel error.

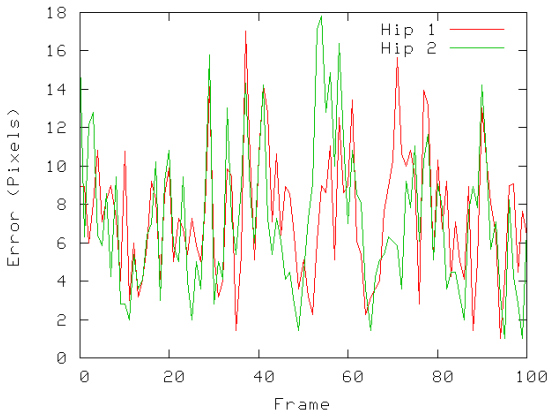

Figure 4.7: Hip coordinate pixel error.

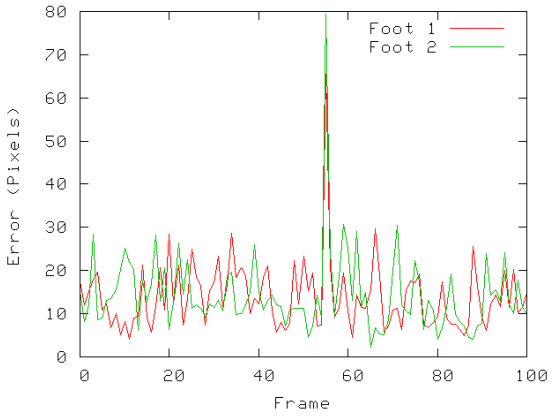

Figure 4.8: Feet coordinate pixel error.

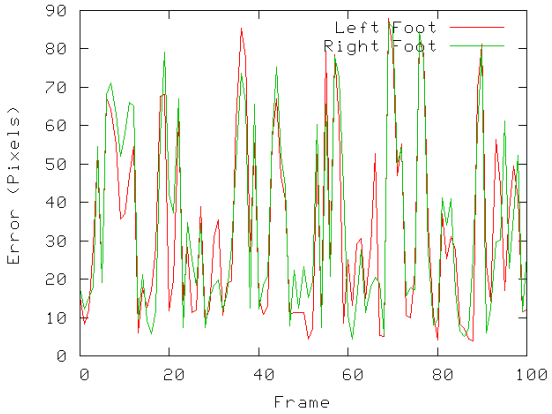

Figure 4.9: Left and right foot coordinate pixel error.

4.5 Performance Measurements

Bounding Boxes

Two bounding boxes around the human are shown in Figure 4.10. The upper and lower segments are each enclosed in a two dimensional axis aligned bounding box with principal axis and centre of mass overlaid. A whole body two dimensional axis aligned bounding box with centre of mass overlaid is shown in Figure 4.11 and a non-aligned three dimensional bounding box is shown in Figure 4.12.

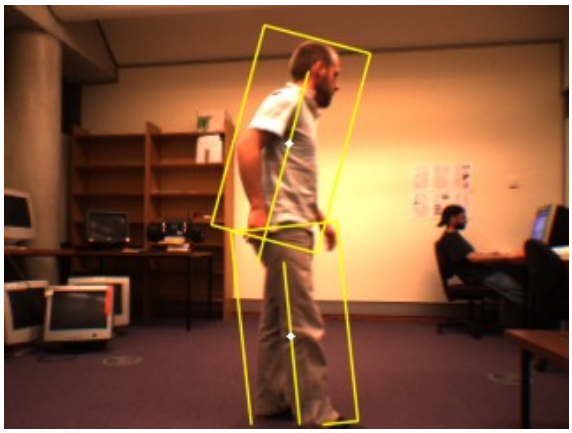

Figure 4.10: Upper and lower bounding boxes with principal axes and centre of mass also shown.

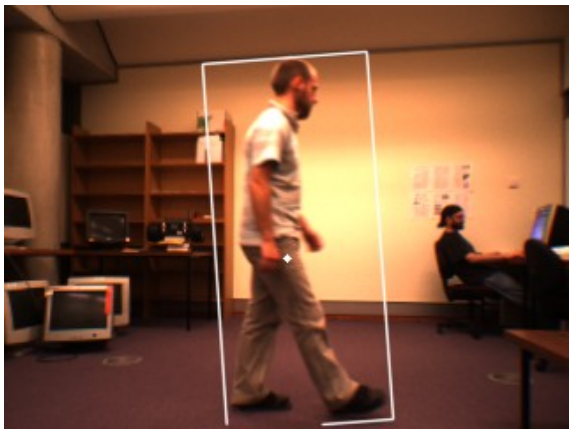

Figure 4.11: Whole body bounding box with centre of mass.

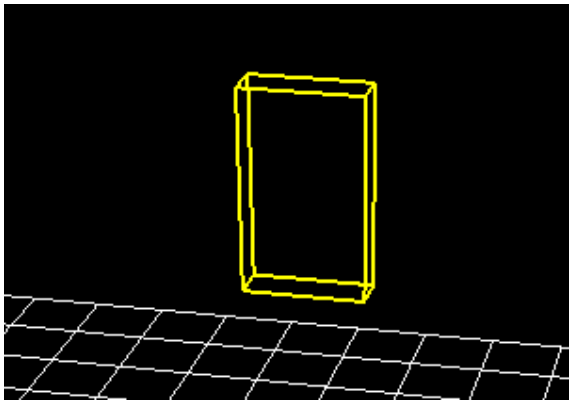

Figure 4.12: A three dimensional bounding box.

Cyclic Motion

A graph showing the cyclic motion of a human via the area below the centre of mass is shown in Figure 4.13.

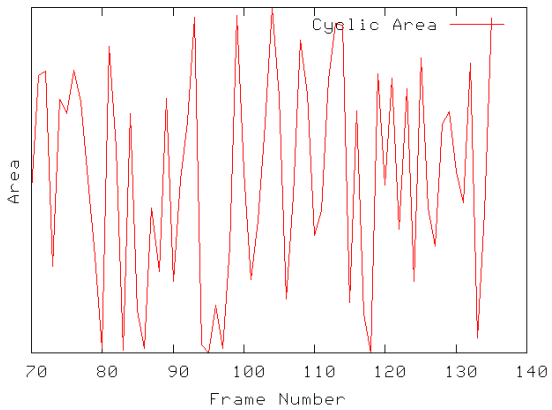

Figure 4.13: Graph of cyclic motion of gait moving parallel to the camera view plane.

Speed & Acceleration

Speed and acceleration are shown in Figure 4.14 calculated over a series of 335 frames.

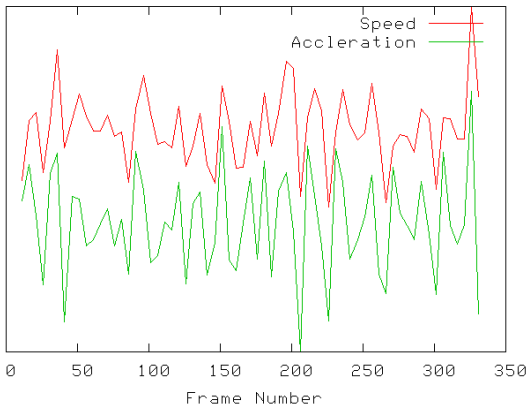

Figure 4.14: Graph of the speed (red) and acceleration (green) of a moving object. Y axis values have been compressed to fit on the same graph.

5 Discussion & Limitations

5.1 Segmentation

The process of using DDA, combined with disparity filtering, performed well. It is a relatively simple segmentation method that yields good results. The disparity filtering method would generally segment more of the moving object than DDA by itself. This segmentation process performs well as it can cope with dynamic scene changes, for example, illumination differences. It would take a few frames to recover from such a change, however this is acceptable as it is a quick recovery process. An example of this is shown in Figure 4.5 where at approximately frame 66 the model generation algorithms fail due to incorrect segmentation but are then corrected several frames later.

However, on occasion, the disparity filtering process would fail if the moving object got too close to the background, which would be corrected several frames later as the user moved away. Furthermore, on occasion shadow regions would be segmented as part of the moving object. This is a common problem in background segmentation and tracking, for which solutions already proposed could be incorporated to make the process further robust [6].

The segmentation process currently relies on there only being a single moving object within the scene for model generation. Future work could involve incorporating a clustering algorithm to further segment regions of movement as captured by the DDA algorithm for subsequent processing. This would allow multiple models to be generated. Furthermore, a system would have to be implemented that maintained identification of multiple objects to allow the speed and acceleration calculations to perform correctly.

5.2 Depth Data

The depth data calculated from the stereo camera was generally adequate for the required purposes. However, there would often be holes in the segmented region that the depth could not be computed which has the potential to cause the model generation process to fail. Through simple experimentation it should be noted that the stereo camera can generate far more dense disparity data when used outside in bright lighting conditions when compared to the laboratory setup.

The region of interest approach to estimate three dimensional coordinates aided in removing this problem. The Figures 4.2, 4.3 and 4.4 illustrate that the estimated values generally fit on the line to the next detected coordinate. Further improvements in stereo camera hardware and stereo processing algorithms will aid in negating this issue.

5.3 Model Generation

It is difficult to compare model generation results across computer vision research, often because results are not reported in enough detail or experiments are highly constrained. Results often contain minimal detail because the process of producing accurate and comprehensive results is very time consuming and some times very difficult.

The average error for finding the hips, neck, centre all performed well being under 10 pixels. The system is robust at finding these significant model points in a cluttered scene. The feet average error, whilst worse than the previously mentioned points, still performed well with both being below 15 pixels. This implies that the constraints hold true for common human motion through a scene.

Whilst the identification of the feet positions is acceptable, both below an average error of 15 pixels, the average error statistics for left and right foot identification did not perform well. There are two reasons

for this, firstly, there are cases when the detected movement direction is incorrect, when turning around for example, and secondly, unless the user walks exactly parallel or perpendicular to the camera view plane, the system could confuse which foot is further away. Furthermore, as the foot finding algorithm is not perfect the depth could be established for a point that is not equal to where the foot really is.

5.4 Performance Measurements

The computation of cyclic motion performed correctly when the user follows a path parallel to the camera view plane as shown in Figure 4.13. However, the process would produce incorrect results when the user walked backwards or forwards with regards to the camera. This is primarily because this motion does not cause the legs to occlude on another allowing the computation of the cycle motion via area.

The bounding boxes enclosing the whole, upper and lower body performed accurately. This is because they primarily relied on the segmentation process to perform correctly. These boxes were aligned correctly as computing the principal axis of a region is a well understood computation.

The speed and acceleration calculations are accurate. The justification is that they are based on the centre being estimated at the waist, which holds true with an average pixel error rate of 5.52, and the accurate estimation of the three dimensional position, which is shown to be accurately computed even if there is no depth data associated with the required pixel at that particular time.

6 Conclusion & Future Work

This research has presented a novel stereoscopic approach to esimating the pose of a human with three dimensional model generation. Two of the fundamental steps in computer vision based motion capture have been addressed; tracking and pose estimation.

Tracking, or segmentation, is performed through the use of the double frame differencing algorithm in conjunction with disparity filtering. The tracking algorithm has shown to be robust in cluttered and dynamic environments preparing the data for pose estimation. Future work could further involve clustering, as at the moment only a single segmented region is identified. If a clustering algorithm were implemented it would allow multiple users to have their associated models and performance measurements accurately computed. Furthermore, a method of identification of users between frames would be required for correct performance measurement computation.

The pose estimation process has shown to be accurate at modelling a human moving naturally through a scene, identifying significant body parts such as head, waist, hips and feet. Future work could involve estimating other parts of the body such as knees, shoulders and hands. Furthermore, the current model generation has no knowledge of how noisy the data is and thus has no capabilities of automatically switching between model generation levels. Methods to measure computer vision noise could be further investigated to allow such capabilties. A successful approach is presented that constructs regions of interest around required three dimensional points allowing estimation of the required data. Without such a scheme the model generation processes would fail.

Several movement parameters were established allowing easier interpretation of the user's movement through a scene. Such parameters included the principal axes of the whole, lower and upper bodies, as well as associated two dimensional axes aligned bounding boxes. A simple three dimensional whole body enclosed bounding box was also presented. The speed and acceleration of the moving object are also computed. Cyclic motion is computed as the visible area of the lower body which has shown to be accurate with movement parallel to the camera view plane.

Bibliography

[1] TRICLOPS Sofware Development Kit (SDK) Version 3.1 User's guide and command reference. Tech. rep., Point Grey Research, 2004.

[2] AGGARWAL, J. K., AND CAI, Q. Human motion analysis: A review. *Computer Vision and Image Understanding 73*, 3 (1999), 428–440.

[3] BANKS, J., AND CORKE, P. Quantitative evaluation of matching methods and validity measures for stereo vision. *International Journal of Robotics Research 20*, 7 (July 2001), 512–532.

[4] COHEN, I., AND LEE, M. W. 3d body reconstruction for immersive interaction. In *Proceedings of the Second International Workshop on Articulated Motion and Deformable Objects* (London, UK, 2002), pp. 119–130.

[5] FORSYTH, D. A., AND PONCE, J. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[6] GAMBA, P., LILLA, M., AND MECOCCI, A. A fast algorithm for target shadow removal in monocular colour sequences. In *Proceedings of the 1997 International Conference on Image Processing* (Washington, DC, USA, 1997), IEEE Computer Society, p. 436.

[7] GAVRILA, D. M. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding 73*, 1 (1999), 82–98.

[8] HARITAOGLU, I., HARWOOD, D., AND DAVIS, L. Who, when, where, what: A real time system for detecting and tracking people. In *Proceedings of the Third Face and Gesture Recognition Conference* (1998), pp. 222–227.

[9] ISARD, M., AND BLAKE, A. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision 29*, 1 (1998), 5–28.

[10] IWASAWA, S., EBIHARA, K., OHYA, J., AND MORISHIMA, S. Real-time human posture estimation using monocular thermal images. In *Proceedings of the 3rd. International Conference on Face & Gesture Recognition* (Washington, DC, USA, 1998), IEEE Computer Society, p. 492.

[11] JAIN, R., KASTURI, R., AND SCHUNK, B. G. *Machine Vision*. McGraw-Hill, United States of America, 1995.

[12] JU, S. X., BLACK, M. J., AND YACOOB, Y. Cardboard people: A parameterized model of articulated motion. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition* (Killington, Vermont, 1996), pp. 38–44.

[13] LEE, K., AND GREEN, R. D. Temporally Synchronising Image Sequences using Motion Kinematics. In *Proceedings of Image and Vision Computing Conference of New Zealand 2005* (2005).

[14] MOESLUND, T., AND GRANUM, E. Multiple cues used in model-based human motion capture. In *Proeedings of The fourth International Conference on Automatic Face and Gesture Recognition* (2000), p. 362.

[15] MOESLUND, T. B., AND GRANUM, E. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding 81*, 3 (2001), 231–268.

[16] OHYA, J., UTSUMI, A., AND YAMATO, J. *Analyzing Video Sequences of Multiple Humans*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[17] ROSENHAHN, B., BROX, T., KERSTING, U. G., SMITH, A. W., GURNEY, J., AND KLETTE, R. A system for marker-less motion capture. *Künstliche Intelligenz*, 1 (January 2006), 45–51.

[18] SIGAL, L., AND BLACK, M. J. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 2041–2048.

[19] SMITH, A. R., AND BLINN, J. F. Blue screen matting. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), ACM Press, pp. 259–268.

[20] STARNER, T., AND PENTLAND, A. Real-time american sign language recognition from video using hidden markov models. In *Proceedings of the International Symposium on Computer Vision* (Washington, DC, USA, 1995), IEEE Computer Society, p. 265.

[21] STAUFFER, C., AND GRIMSON, W. E. L. Adaptive background mixture models for real-time tracking. In *Proceedings of 1999 Conference Computer Vision and Pattern Recognition* (1999), pp. 2246–2252.

[22] TOYAMA, K., KRUMM, J., BRUMITT, B., AND MEYERS, B. Wallflower: Principles and practice of background maintenance. In *Proceedings of the International Conference on Computer Vision* (1999), pp. 255–261.

[23] WELCH, G., AND BISHOP, G. An introduction to the kalman filter. Tech. rep., 2004.

[24] WILSON, P. I., AND FERNANDEZ, J. Facial feature detection using haar classifiers. *Journal of Computing Sciences in Colleges 21*, 4 (2006), 127–133.

[25] WREN, C. R., AZARBAYEJANI, A., DARRELL, T., AND PENTLAND, A. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 7 (1997), 780–785.

[26] WREN, C. R., AND PENTLAND, A. Dynamic models of human motion. In *Proceedings of the 1998 Conference on Automatic Face and Gesture Recognition.* (1998), pp. 22–27.

[27] YOSHINARI, K., AND MICHIHIKO, M. A human motion estimation method using 3-successive video frames. In *Proceedings of the International Conference On Virtual Systems and Multimedia* (1996), pp. 135–140.

[28] ZANG, Q., AND KLETTE, R. Robust background subtraction and maintenance. In *Proceedings of the 17th International Conference on Pattern Recognition* (Washington, DC, USA, 2004), vol. 2, IEEE Computer Society, pp. 90–93.

[29] ZIEGLER, J., NICKEL, K., AND STIEFELHAGEN, R. Tracking of the articulated upper body on multi-view stereo image sequences. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 774–781.

A Appendix

A.1 Stereo Algorithm Accuracy

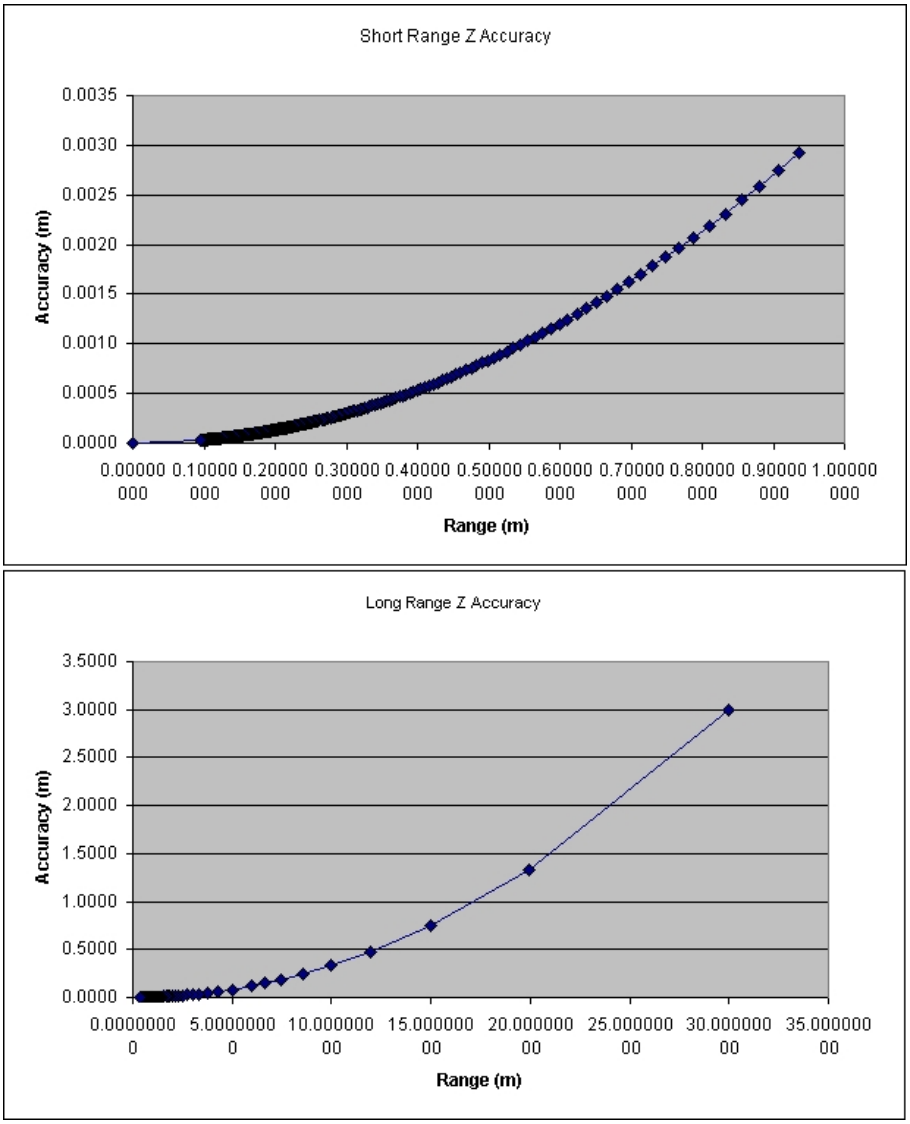

Figure A.1: Accuracy of the stereo algorithm.