Real-time multi-camera stream analysis for player interaction in a 3D world

Conor O'Reilly

A dissertation submitted to the University of Dublin,
in partial fulfilment of the requirements for the degree of
Master of Computer Science in Interactive Entertainment Technology

2008

Declaration

I, the undersigned, declare that the work described in this dissertation is, except where otherwise stated, entirely my own work, and has not been submitted as an exercise for a degree at this or any other university

Conor O'Reilly

Dated: 9th September 2008

Permission to Lend and/or Copy

I, the undersigned, agree that the Trinity College Library may lend or copy this thesis upon request.

Conor O'Reilly

Dated: 9th September 2008

Acknowledgements

I would like to take this opportunity to sincerely thank the following people for their support, help, encouragement and insights.

To my wife Simone and children - Shauna, Simon and Patrick - for putting up with and supporting yet another mad endeavour. Simone's mother for suggesting the impossible was possible - give up work, forget the mortgage become a student for a year. Her father, Simon, for just always being there whenever help was needed. To my mother and father for making the impossible a financial reality and their unwavering belief in my abilities. John Cusack for supporting a poor student's sushi and beer habits for an entire year, the payback will be painless. Paul, Enda, Sean, Tricia, Kieran, Richard, John and the rest of the lads and lassies at Propylon for their 'back to school' support while they took up the burden of travelling to the US in my stead. Pat, Mick, Martin and Padraig for coaching the St John Ballinteer GAA under 11's football team in my absence and winning our first cup. To the gang at Fine Line writers group for their companionship and constant encouragement through the years.

Many thanks to my supervisor Dr. Rozenn Dahyot for illuminating the world of mathematics in a meaningful and practical manner, allowing a beguiled lowly traveller in a foreign land to find his way. And last but not least, to my fellow classmates on the MSc. in Interactive Entertainment Technology in Trinity College Dublin - thanks for opening old eyes to the possibilities of new technology, for all the virtual fun and games and most specifically for the rallying cry when the spectre of assignment deadlines loomed near - "Pints? Did you say Pints? I heard Pints!".

Conor O'Reilly

University of Dublin

September 2008

Abstract

Analysis of the HumanEva I & II [91, 105, 106] datasets as a proof of concept for the implementation of a 3-D pose estimation system for real-time avatar animation within a single player game played in a family room installed with multiple stationary cameras. Reconstruction of the 3-D volume of the player from multiple 2-D video streams. Investigation of an approach to 3-D skeleton recovery from multiple 2-D video streams.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The motivation behind this research is the following question:

— Is computer vision ready to be used as a human computer interface for a single player game played in a family room installed with multiple cameras?

A smaller range of more fundamental questions arise from the enquiry above:

— What vision based techniques are suitable for this type of environment?

— Given multiple camera views can a 3-D silhouette of a single player be inferred?

— Can a 3-D skeleton of the player be derived from the 3-D silhouette and can this skeleton be used to animation, in real-time, an avatar in a game?

1.2 Goal

The goal of this research is to investigate the answers to these questions by reviewing the current techniques, approached and implementations in this research area, loosely referred to as - markerless human motion capture, and to implement techniques appropriate to the environment under consideration. Appropriate in this context refers to the techniques most suited to the dynamics and constraints of the environment as considered under the following headings:

— Efficiency - operational efficiency given the real-time context

— Accuracy - pose estimation accuracy within tolerances that are acceptable to game play rather than actual spacial fidelity to the real world positions

— Perception - acceptable player perception of the avatars movement and interaction within a virtual 3D space.

In perusing this research the intention is to establish the feasibility and usefulness of creating such an environment. An essential component of the analysis is the robust real-time recovery of 3-D dynamics from the 2-D dynamics.

1.3 Types of motion capture applications

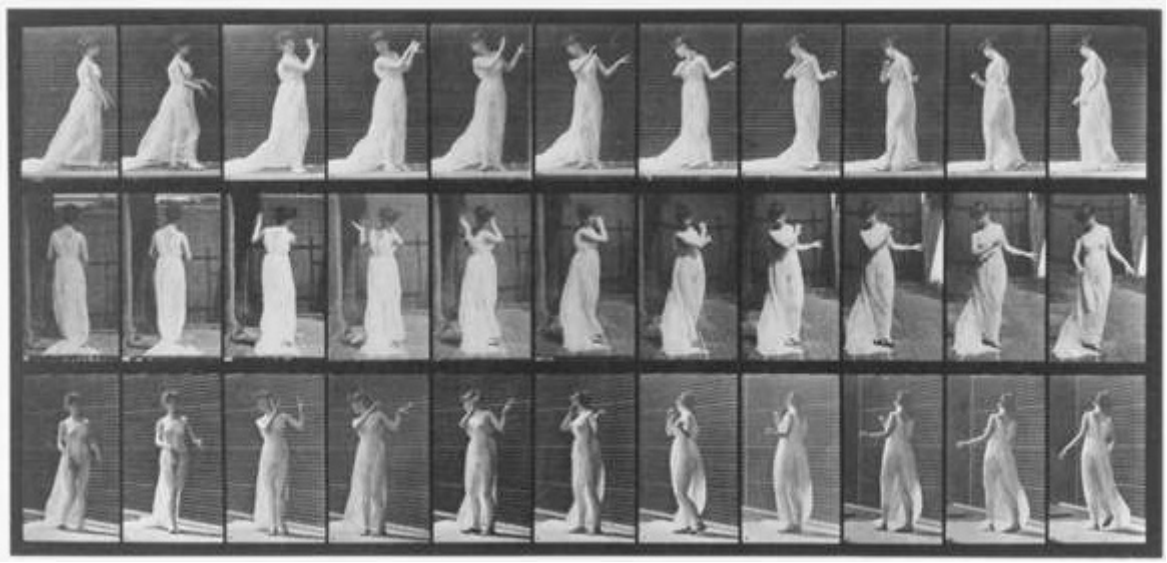

Figure 1.1: *Eadweard J. Muybridge. Walking, turning around, action of aversion. Animal Locomotion Plate 55, 1887*

The term human motion capture is used to describe the analysis of human actions by a computer system, the term has been further defined [68] to refer to the capture of large scale body movements such as the arms, legs, head and torso. This definition allows for the tracking of a human as a single object or to be more specific, as a self-occluding, self-colliding, 3-D articulate skeletal structure with a number of joints about which motion occurs to a high degree of freedom. A simpler definition of motion capture [65] defines it is the problem of extracting the full or partial parameters associated with the motion and actions

of a human subject from video data or a sequence of images.

Some of the earliest Human Motion Capture analysis can be found in the works of Eadweard J. Muybridge (1830 - 1904) who used 36 lenses with 12 to 24 cameras, placed at 30-, 60-, and 90-degree angles to his subjects to photograph the motion of humans and animals [74] [73]. He initially used motion analysis in 1877 to prove that all four of a horse's hooves left the ground at the same time during a gallop. Today the major application areas for human motion capture are as follows:

— Smart Surveillance

— Intelligent Environments

— Virtual Reality, Augmented Reality and Augmented Perception

— Motion Analysis

— Entertainment

— Control - Human Computer / Machine interfaces

This dissertation focuses on the control aspects of human motion capture where the actions of the tracked subject can be used as a human computer interface (HCI) to animate an avatar within the 3D virtual environment of a game.

Surveillance applications are interested in tracking, perhaps even identifying, one or more individuals over time while monitoring for specific actions, activities or behavioural patterns. [83, 53, 26]

Intelligent environments can be viewed as spaces (rooms) where humans and computers interact or collaborate. Such an environment requires a significant level of functionality such as: intruder detection, multiple person tracking, subject identification, human pose - posture and movement analysis. An intelligent environment facilitates the integration of human computer interfaces with intelligent meeting rooms, teleconferencing, or performance spaces. Cars and other vehicle environments are also considered potential candidates for intelligent environment systems[101, 100].

Virtual Reality, Augmented Reality and Augmented Perception. These applications are at the crossroads of computer graphics, computer vision, and human computer interaction. The phrase **mixed reality** has been coined to refer to the spectrum of applications between virtual environments and real world environments that integrate these environments in real-time [38, 6]. An example of an augmented perception application is the translation of a visual environment into an acoustic representation of the same environment in order to guide a blind person[3].

Motion analysis applications may involve the diagnosis of orthopaedic issues through gait analysis [12, 114] or the indexing of sports videos based on motion sequences [61, 60]. The ability to recognise humans and their activities is also relevant to the design of machines capable of interacting in a human inhabited environment.

Entertainment, motion capture applications as used within the film industry for the purposes of placing real actors in virtual scenes, animating virtual characters [34] or synthesising human motion [5] are considered here as being entertainment based applications of human motion capture. Though the animation of an avatar in a game could be considered under the heading of entertainment for the purposes of this dissertation it is being considered under the category of control.

There are many aspects to **Control - Human Computer / Machine interfaces** applications and some of the above categories could be considered control applications or at least vision-based interfaces (VBI). But in the context this research the following applications are most applicable since they employ multiple cameras and are aimed at avatar or agent control within virtual worlds:

— Virtual world interaction via motion capture [67]

— MIMIC, real-time animation of a virtual agent [54]

— Real to virtual world figure control software [120]

— Personalised human avatars and the automatic generation of skeletons [99, 98, 32, 33, 2]

— The use of human motion in games [78]

A very succinct summary of the above has been provided by Sminchisescu [93] on this personal website `Ref:http://sminchisescu.ins.uni-bonn.de/` it reads as follows: "Recognising visual objects, understanding video content and transferring this into 3-dimensional models is the basis of advanced special effects, digital libraries and image indexing systems, or the construction of humanoid robots that can localise objects, recognise people, comprehend actions and interact with the world seamlessly."

One factor stands out in all of the above control applications - the environments used are highly constrained and often specially build for the purposes of simplifying the markerless motion capture process. In assessing the feasibility of implementing this technology in a family room we need to take a close look at the assumptions and constraints being applied by the researchers.

1.4 Motion capture systems in general

Moeslund's survey of the papers in this area from 1980 to 2000 [68] and from 2000 to 2006 [69] considers motion capture systems to consist of two high level components - sensing and processing.
Sensing can be considered in terms of active or passive sensing. Active sensing places devices on the subject that transmit position and motion data. Whilst active markers are an effective means of motion capture in certain contexts, their presence can interfere with the motion of the subject. It also requires specialist equipment and set-up expertise. These factors make them inappropriate in the context of the environment being considered in this research.
Passive sensing is based on the analysis of visible light or other electromagnetic wavelengths. Passive sensing does not require the wearing of active devices, passive markers though maybe used. Passive markers are composed of light reflecting material that are placed on joints and other areas of interest. Vision based human motion sensing is considered to be passive sensing. However these passive markers have to be attached to the skin or attached with Velcro to a special full body suit, again the approach requires special equipment and set-up expertise.
The goal of markerless passive sensing is to allow a subject to enter and interact within the motion capture environment without the wearing of a special suit or the attachment of special markers. It is this approach that is most suitable to providing a game interface within the environment of a family room.
Moeslund's survey further decomposes the elements of a human motion capture system into four functional components - initialisation, tracking, pose estimation and pose recognition.

— Initialisation: Refers to environmental initialisation, such as camera calibration, light levels analysis, environment / background mapping and subject analysis / dimension acquisition.

— Tracking: Following the movements of an object or subject of interest.

— Pose Estimation - the estimation of the relative position and orientation of a 3D object or subject with respect to a reference camera system [86].

— Pose Recognition, is defined [68] as: the analysis of a pose in order to recognise actions performed by the subject.

It is under these functional categorisations that the techniques researched and implemented to-date in this area will be considered with the exception of pose recognition which is beyond the scope of this dissertation.

1.5 Motion capture environment

The motion capture environment of interest is an average size (12 x 16 x 8 feet) family room equipped with multiple stationary cameras. Average room size varies from country to country, with Canada at the top of the scale and Finland at the bottom [111].

The goal of this research is to consider the feasibility of creating a software API for a games engine that would allow the player to be taught to dance, learn yoga, perform exercises or practise karate. To this end the player's motion must be tracked in 2D, a 3D pose ascertained and the pose compared with the correct pose for the particular learning exercise. This is the context in which the current techniques available in this area are being reviewed within the State of the Art chapter of this document, chapter (2).

Whether a computer game is an appropriate facilitator for the learning of these skills is not under review. These particular skills are mentioned because they represent a range of poses that are not your typical standing, walking or running poses but involve the subject lying on the floor or perhaps standing on their head.

The most appropriate configuration and number of the cameras in the room will not be investigated during this research. But these issues will be considered with regard to the available techniques as the poses described above may require particular camera orientations - a ceiling camera, orthogonal camera positions, a camera perpendicular to the body axis of a standing person.

Creating this environment would be expensive without assurances that technology and techniques available made the creation of this application feasible. To this end, the synchronised video and motion capture data from the HumanEva I [8, 105] project will be used as test data.

1.5.1 The HumanEva dataset

There has been significant research in this area in recent years, Moeslund's initial survey [68] referenced only 130 papers in 20 years. His second survey is based on over 350 publications published over a six year period. One issue raised in Moeslund's initial survey was the lack of a normative video data set that could be used by all researchers in the area as a means of quantitatively comparing their algorithms and techniques. The HumanEva [91] dataset is such a dataset. There are in fact two datasets:

HumanEva-I

The HumanEva-I dataset [105] contains 7 calibrated video sequences (4 grey scale and 3 colour) that have been synchronised with 3D body poses obtained from a motion capture system. There are 4 subjects performing 6 common actions e.g. walking, jogging, gesturing.

HumanEva-II

The HumanEva-II dataset [106] is smaller than the first but it has been generated using a hardware synchronised system with 4 colour video cameras and 12 Vicon motion capture cameras. The motion capture was recorded at 120 FPS and the video at 60 FPS. The video has been distributed as a sequenced of PNGs rather than as AVI files. It contains 2 test sequences with subjects S2 and S4 from the HumanEva-I set. The first 3 frames contains motion capture data for initial pose tracking initialisation or for deriving subject skeletal measurements.
More importantly the environment and the subjects have not been highly constrained. For instance, the mat on the floor is a dark red, similar to skin which interferes with colour based segmentation. The subjects are wearing everyday clothes and elements of the background have not remained static through out the filming process. The video sequences therefore provide a good indication (at a smaller scale) of the issues that would be present in a family room environment.

1.5.2 Limitations of this approach

This approach has several issues as it does NOT allow the following to be tested :

— Environmental configurations e.g. the placement of cameras, how many cameras necessary.

— System configurations e.g. what camera equipment to use, how to synchronised camera signals - is synchronisation even necessary, what type of cameras to use - z-cameras, IR cameras, visible-light cameras

— Segmentation issues that are dependent on the potentially cluttered or obscuring background of a family room e.g. its objects, decoration and lighting.

— Segmentation issues dependent on nature of the subject - adult, child, pensioner, handicapped, other structural issues. The clothes they are wearing in relation to the background and how baggy, bulky or obscuring the clothes would effect the system.

— Other movements in the room and their effects e.g. non-player occupants, pets, their shadows.

— Changing light conditions e.g. window light, lights turned on / off, shadow conditions.

Most important of all player perception is not testable without a real-world implementation e.g. avatar motion response time and virtual environment interaction accuracy. That said the time period for this research only allows for an initial analysis of the approaches and techniques that could be used and concentrates mainly on obtaining a 3D volume for a person within this environment.

1.5.3 Advantages of this approach

The HumanEva dataset has been derived within a contrived environment that is suitable for vision based human motion capture. Though this is far from the real life environment being considered, it does provide the following:

— The camera calibration information is available with the data set. Though no details on the size of the subjects, placement of the cameras or size of the room are available.

— HumanEva motion capture data can be used to test tracking algorithms and to represent the pose a player has to mimic.

— HumanEva video can be used as the multi-camera input stream that contains the subject to be tracked.

— The HumanEva motion capture data can be used as the ground truth for establishing the accuracy of the volume reconstruction and pose estimation implementations.

— And most importantly the multi-camera environment does not need to be constructed in order to analyse the video streams.

The scope of this research is quite wide and it extends from the generalist - what techniques could work within the chosen environment? To the specific - can a game avatar be animated by analysing 2-D dynamics from multiple cameras and thus recovering in real-time the 3-D dynamics of a single player? Analysing and using the HumanEva dataset provides a balance between these two questions given the constraints of time, resources and funding.

1.6 Focus of the State of the Art review

The related work review will focus on the multi-camera stream analysis research to-date that is most applicable to the environment under consideration. The review will be structured around the functional areas listed below as these represent the components that would need to be implemented, or require consideration, in an actual implementation.

— Infrastructure

— Initialisation

— Subject Modelling (Human Models)

— Background subtraction (Subject Extraction)

— Model acquisition and reconstruction

— Motion Tracking

— Pose Estimation

— Pose Definition

— Visualisation

— Performance

1.7 Contributions

In terms of leading the field in the establishment of techniques and infrastructures that would allow computer vision to be used as a human computer interface for a single player game played in a family room installed with multiple cameras. The researchers below have made the greatest advances.

— 2002 Mikic [65] - Voxels (University of California, San Diego)

— 2002 Theobalt [99] - Volume reconstruction for video-based human motion capture (Max Planck Institute)

— 2005 Cheung [22, 23] - Shape-From-Silhouette, Visual hulls (Carnegie Mellon University)

— 2006 Corazza, Muendermann [70] - Visual Hulls (Stanford)

— 2008 deAguiar and Theobalt [31] SIGGRAPH2008 - Actors performance capture with multi-view video (MPI Informatik / Stanford)

— 2008 Vlasic, Baran, Matusik and Popovic [116] SIGGRAPH2008 - Articulated mesh animation from multi-view silhouettes.

The contribution of this research are:

— Development and analysis of a new technique for extracting the 3-D skeleton of a subject from the multiple camera views. Whilst other researchers have used silhouettes and voxels to interpret the 3-D volume of a subject their interest has been in the creation of a visual hull for the subject. The volume occupied and the internal voxels are purged in favour of the outer surface voxels. The

method investigated in this research creates a density graded contour map of the subjects volume similar to an MRI scan. This technique potentially allows the inner skeleton of the subject to be reliably obtained whilst also leaving more detail than shape from silhouette techniques.

— Analysis of the current research in the area in terms of using markerless vision techniques to animate an avatar in a game.

— Analysis of the HumanEva I and II dataset with respect to volume based subject reconstruction techniques and real-time subject segmentation techniques. Development of tools and tests in C++ / OpenCV and MATLAB in order to perform the analysis.

— Identification of issues within the HumanEva I data set in relation to the calibration of the colour and grey scale cameras.

  – That do not appear to be calibrated in the same world space and so are not immediately usable for visual hull construction as a 7 camera configuration.

  – The grey scale cameras BW1 and BW4 cameras have calibration issues as also identified by Mundermannhad [71].

— An implementation approach to the design of a system for animating an avatar in a game. Definition of the key modules required to build a system which would allow computer vision techniques to animate the avatar.

— Identification of the potential issues and challenges involved in building a system that would allow computer vision techniques to be used to animate an avatar in a game.

— Outline of an approach to completing the research above and implementing a mesh based game avatar within a collision detection based game.

1.8 Document Structure

The rest of this document is laid out as follows:

— **State of the Art**, chapter 2 - a review of relevant prior research in this area in terms of their applicability to this research.

— **Theoretical Basis**, chapter 3 - an overview of the techniques and algorithms implemented.

— **Implementation Overview**, chapter 4 - an overview of the analysis / implementation approach, processes, methodology and outputs.

— **Evaluation of Background Subtraction Methods**, chapter 5 - the silhouette generation techniques implemented and evaluated. Includes an Evaluation and Future Work section.

— **Initial Volume Analysis**, chapter 6 - the techniques used to initially explore the HumanEva datasets.

— **Volume Rendering Code Overview**, chapter 7 - explanation of the structure of the main subject volume reconstruction code. Includes an Evaluation and Future Work section.

— **Volume Analysis : HumanEva I**, chapter 8 - a review of the subject volume reconstruction techniques performed on the HumanEva I dataset. Includes an Evaluation and Future Work section.

— **Volume Analysis : HumanEva II**, chapter 9 - a review of the subject volume reconstruction techniques performed on the HumanEva II dataset. Includes an Evaluation and Future Work section.

— **Conclusions**

— **Bibliography**

— **Appendices**

The Conclusions chapter is a standalone chapter but all other chapters are intended to be read in consecutive order. Each builds on the other leaving the Volume Analysis: HumanEva II chapter unfortunately rather short.

Chapter 2

State of the Art

Over the last ten years there has been a huge amount of activity in this research area. Moeslund's initial survey [68] referenced only 130 papers in 20 years. His second survey [69] is based on over 350 publications published over a six year period. The primary surveys in this research area and their taxonomies are presented below:

— 1995 Cedras and Shah [17] surveyed motion based recognition research under the following headings:

  – Motion Information Extraction - trajectory methods, optical flow, region based features

  – Motion Recognition - cyclic motion detection, lip reading, gesture interpretation, motion verb recognition

  – Human Motion Tracking and Recognition

— 1997 Aggarwall and Cal [1] categorise the research papers they reviewed under the following headings:

  – Body structure analysis (pose estimation) - model based and non-model based, where no subject *a priori* information was used

  – Tracking : Single cameras, Multiple cameras

  – Recognition : State-Space, Template Matching

— 1999 Garvrila [40], divided the existing research as follows:

- 2-D approaches without explicit shape models, where pose recovery is not used e.g. body part recognition. Instead human movement is described in terms of simple low-level, 2-D features within an area of interest.
- 2-D approaches with explicit shape models, where knowledge of how a human model would appear in 2-D is used to generate a model that is checked against the current view. Body parts are often segmented, tracked and labelled. Various model types are used - stick, blobs, edges etc.
- 3-D approaches : 3-D articulate pose recovery methods which try to recover joint angles with respect to the subjects co-ordinate system. Model are used to predict events such as self-occlusion and self-collision where a typical framework contains the following components - predict, synthesis, image analysis and state estimation [77]

— 2001 Moeslund and Granum [68] review of the papers in terms of their functional components:

- Initialisation
- Tracking
- Pose Estimation
- Pose Recognition

— 2003 Wang L. , Hu W., and Tan T. [118] review papers from 1989 onwards, but 70% of the references are from 1996 onwards to 2001. A hierarchy of approaches is defined under the following taxonomy:

- Human Motion Analysis
- Human Detection and Tracking
- Human Activity Understanding

— 2006 Moeslund, Hilton and Kruger [69] reviews papers from 2000 to 2006 under the same functional taxonomy as the 2001 survey [68].

— 2006 Poppe [80] reviews papers on markerless pose estimation research as follows:

- Model-based approaches relying on top-down and bottom-up pose estimation. Top-down approaches model a projection of the human body with the image observed, whilst the bottom-up approach matches body parts to assemble a human body.

- Model-free approaches where no explicit human body is available establish a relationship between observation and pose through example-based or learning-based algorithms. Learning-based approaches use training data to establish the relationship between image-space and pose-space. Example-based approaches looks up examples in a database and compute similarity matches e.g. Ren [85].

— 2006 Sigal and Black [91] provide a summary of human motion and tracking algorithms in their technical report on the HumanEva-I dataset [105].

Given the context of this research the following functional taxonomy has been derived as a basis for discussing the current state of the art in this research area. The functional areas have been chosen as they form the basic components of a processing pipeline that would process views from multiple cameras and articulate an avatar in a game.

— Infrastructure

— Initialisation

— Subject Modelling (Human Models)

— Background subtraction (Subject Extraction)

— Model acquisition and reconstruction

— Motion Tracking

— Pose Estimation

— Pose Definition

— Visualisation

— Performance

Infrastructure refers to the research environment, the equipment used and how it is configured e.g. camera synchronisation and the hardware / software architecture. The **Initialisation** category pertains to the environment initialisation procedures and processes once the infrastructure is turned on e.g. camera calibration, light level assessment, environment / background mapping, subject detection.
Many of the algorithms rely on human models for tracking and pose estimation. **Model acquisition** is defined by [65] as the estimation of the parameters of the human model being used based on the specific

size and shape of the person to be tracked so that the model accurately reflects the position and pose of the person at the beginning of the motion capture process. The acquisition of the subjects parameters may be performed manually or automatically in real-time [65, 94]. The subject initialisation may occur within the first few frames and be enhanced with subsequent frames or there maybe an initialisation phase where the subject performs a set of scripted motions that calibrate the subject model. Model acquisition improves the accuracy of the foreground / background segmentation and pose estimation processes.

There are a variety of methods for modelling the subject, **Subject Modelling** reviews the various techniques that have been implemented. Humans have been modelled in 2-D as stick figure, collections of blobs, ellipsoids or rigid body segments attached to each other in articulate chains. Or in 3-D based on volumetric and kinematic models using cylinders, tapered cones, cubes, spheres and skeletons.

Background subtraction refers to the methods and algorithms implemented for separating the subject (foreground) from the background image stream. These methods also take into account shadows, highlights and the movement of objects / subjects that are not the target subject.

Motion Tracking - the tracking of the subject / object or parts of the subject / object between consecutive frames is based on finding feature correspondence between frames based on position, velocity, shape, texture and colour [1]. Tracking is a means of preparing data for pose estimation and recognition [68]. The algorithms associated with tracking are based on a number of criteria: single or multiple cameras - moving or stationary, 2-D or 3-D tracking - whole subject / object or parts of a subject / object, outdoor or indoor environment, environments with a single human or multiple humans. Tracking can be subdivided into model-based, region-based, active-contour-based and feature-based [118].

Pose Estimation is the process of estimating the relative position and orientation of a 3-D object with respect to a particular camera system [86]. In the case of an articulate human subject the pose estimation process consists of two stages modelling and estimation [80, 93]. Modelling involves the construction of the likelihood function that takes into account all the parameters of the infrastructure, the images obtained, the subject model being used and the matching algorithm. Estimation is concerned with finding the most likely pose given the likelihood surface. The problem is that projecting the world into images suppresses depth information and this makes pose estimation fundamental difficulty.

Pose Recognition is defined by Moeslund [69] as the recognition of a subject or subjects actions, activities, behaviours and even their identify within one or more frames. This process is outside the scope of this research but **Pose Definition** which is the encoding of pose information in such a way as to allow further Pose Recognition is of interest to this research.

Visualisation refers to the various methods and tools for analysing and displaying the captured motion, the tracking parameters and the pose estimation models.

Performance deals with the issue of qualifying the robustness and quality of the techniques being used. Asking the questions - what methods have been used to-date? Are they qualitative evaluation or quantitative evaluation methods?

The following diagram summaries the multi-camera input to avatar animation processing pipeline based on the stages described above. see figure 2.1.

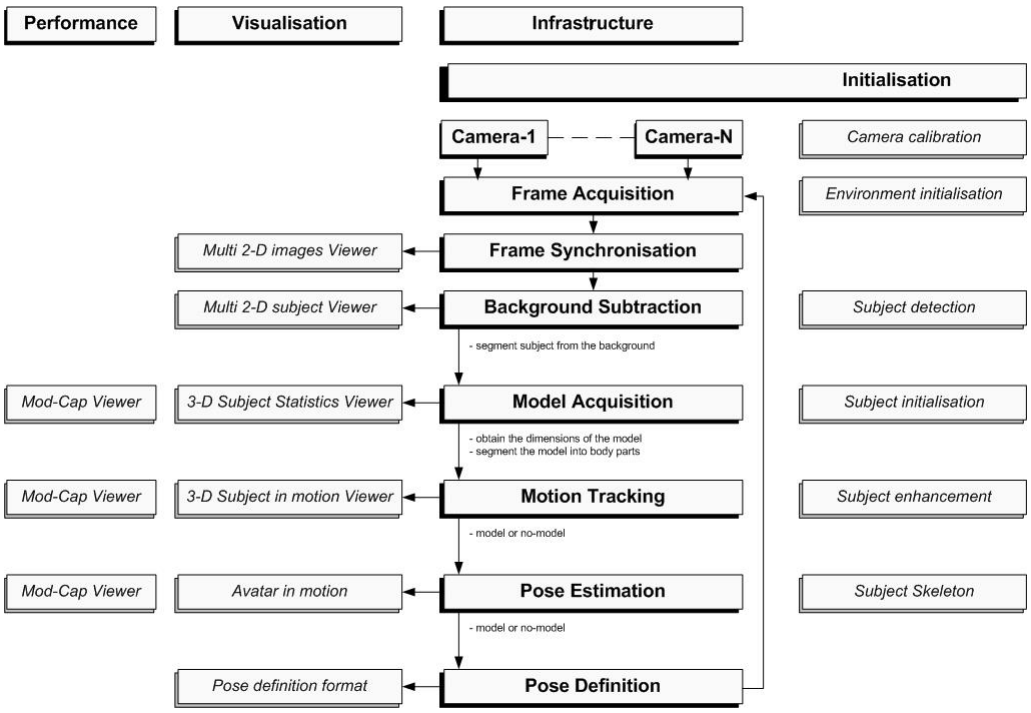

Figure 2.1: *Multiple camera input to avatar animation processing pipeline*

2.1 Assumptions

The goal of computer vision is to recover information from images or image streams. The application potential of this research area is huge as illustrated by the diverse nature of the application areas described in the Introduction 1.3. Key to extracting information from computer vision based processes / algorithms, and interpreting the data, is fully understanding the assumptions, constraints and contexts of the techniques used.

Many of the techniques require unrealistic constraints in order to reduce the complexity of the problem but this make the techniques unusable in a real-time unconstrained environment. In defining a solution

for our environment we need to evaluate the state of the art whilst being cognisant of the potential method limitations, their assumption and the cunning constraints imposed by researchers.
Before listing the assumptions it is worth listing the difficulties inherent in this domain so that the assumptions can be seen in context. Many of the difficulties listed below are discussed by Sminchisescu [93] though mainly in the context of a single camera.

— Projecting the world into images suppresses depth information.

— There are difficulties detecting and tracking humans. The robustness of the algorithms is highly dependent on environmental conditions and the pose/nature of the subject . These issues result in the false detection of a human within indoor and outdoor environments or the breakdown of tracking functions under certain conditions.

— Distinguishing a human subject from the environment may not be accurate or may be noisy due to lighting conditions, shadows, scene conditions and motion blur.

— The human body is an articulate, self-occluding, self-colliding and it can move with a high degree of freedom. Accurate occlusion prediction is important so as not to mislabel body parts or measure them incorrectly. The difficulty is that self-occlusion occurs frequently in humans.

— How to represent a human - a 3-D volume of points, a skeleton, a set of ellipses. The degree of dimensionality to use depends on the level of reasoning, accuracy or reconstruction required. Depending on how many parameters are used within the model, the model may become computationally impractical if real-time performance is required.

— The degrees of motion of a subject is no limitless, volume or kinematic based models for instance must ensure that body parts do not intersect each other. This requires a model that constrains the possible physical movements of a human. This decreases the possible search area but introduces other modelling complexities.

— Human motion is highly rich and occurs at different levels of detail, e.g. walking / running verses hand gestures and facial expressions. Depending on the angle of view certain motions may be missed or ambiguous e.g slight rotational gestures such as occur at the wrists or with the head.

— Clothing worn by humans is deformable and occluding, it produces very strong variations in shape and appearance.

— Models need to take account of the variations in human stature which are dependent on age, health and race.

Many of the assumptions listed below have been augmented based on Moeslund [68] initial survey. Later sections of this document define the environment under consideration in terms of these assumptions.

2.1.1 Movement Assumptions

— The subject remains within the capture area, therefore the volume of interest is stationary and the subject is always within that volume.

— No camera motion or constant camera motion

— Only one person in the capture area at a time

— The subject faces the camera at all times

— Movement is parallel to the camera-plane

— The subject is not occluded by the environment

— Movements are slow and continuous

— Only one or a few limbs moved between frames

— The subjects motion pattern is known

— The ground surface is flat

— The subject is always in an upright position

2.1.2 Environment Assumptions

— The lighting is constant

— The background is static

— The background is uniform

— The camera parameters are known

— Shadow and highlight detection / removal are possible

— Items of a know dimension are in the scene e.g. a checkered floor grid, to aid camera calibration. Mikic [65] calibrates cameras using Tsai's [102] methodology.

2.1.3 Subject Assumptions

— The start pose is known

— The subjects dimensions are known

— The subject is wearing tight fitting clothes

— The subject is wearing clothes of a distinctive colour (or at least distinctive from the background)

— The subject is not wearing either active or passive markers

— The subject is in constant motion

2.1.4 Model dimension estimation assumptions

— The human body is symmetric, left and right side body dimensions are the same

— The person walks into the room standing so that this is the most accurate height measurement as compared to them jumping, sitting or lying. Or an initialisation pose is used to ensure an accurate height measurement

— The aspect ratio and joint lengths of the subject are within the mean and standard deviations for these human attributes as acquired from an anthropometric database[76].

— Subject width measurements can be compared with atypical body-mass-index (BMI) data [94] or anthropometric measures of the human body based on ISO7250

2.1.5 Specialist Hardware Assumptions

— If colour is used [22] then the assumption must be made that the cameras are colour balanced.

— Frame capture by multiple cameras is synchronised so they are all looking at the same volume at the same time.

— Time of flight based z-cameras per pixel distance values have not been washed out be ambient light of the same frequency

2.1.6 Method assumptions

— **Shape-from-Silhouette methods** assume that all of the silhouette images are captured either at the same time or when the object is stationary. This assumption is violated if the object is moving or changing shape thus SFS based approaches treat each time instant sequentially and independently. SFS is best suited to indoor locations where the cameras are static and there there are few moving shadows.

— **The need for accuracy** Accurate reconstruction of a physical space within a virtual space [37] is important for a number of applications such as: interactive visualisation of remote environments or objects, virtual modification of a real scene for augmented reality tasks. But within a game environment perceived accuracy, responsiveness and real-time processing are more important.

2.2 Infrastructure

Most human motion capture research involves the construction of a constrained environment in which the research takes place. Early environments consisted of specialist equipment constructed purely for the purposes of the research. Today commodity cameras and computer hardware can be used to construct the infrastructure required for the markerless analysis of human motion.
Infrastructure is discussed under the following headings:

— Cameras

— Environment

— Lighting

2.3 Camera Configurations

The following camera configurations infrastructures are significant in that they show a progression from specialist hardware to commodity hardware in terms of markerless human motion capture and depth inference with single cameras and multiple cameras:

— Single camera

  – 1997 Bregler [14]

- 2002 Sminchisescu [93] - Three-Dimensional Human Modelling and Motion Reconstruction in Monocular Video Sequences
- 2004 Wagg [117] - Markerless extraction of walking people
- 2005 EyeToy [109] launched for Playstation 2 games, Richard Marks - Manager, Special Projects, Sony Entertainment.

— Stereo camera

- 1991 Chen [18] - computing motion and depth of an object from binocular orthographic views
- 2004 Yonemoto [120] avatar animation based on 2-D blobs and body posture estimation.

— Three cameras

- 1998 Kakadiaris [52] - 3 orthogonal cameras. The subject performs a set of predefined poses that incrementally reveal the structure of the body.
- 2005 Ren [85] - 3 cameras used to controlling animated human characters while they swing dance. A database of dance motions is queried to recover the posture by comparing the silhouette extracted body configuration with motion graphs generated from the database.

— Multiple cameras

- 1995 Kakadiaris [51] early multi-camera research into 2-D segmentation techniques.
- 1996 Garvrila [41] - 4 orthogonal (visible light) cameras - front, back, left, right. Human activity recognition based on body pose recovery, tracking and recognition of movement patterns using chamfer matching with a tapered superquadric model. The system actually tracks multiple people and is much cited by other research papers.
- 1997 Jung [49]
- 1999 Hilton [44]
- 1999 Delamarre [35]
- 2000 Cheung, Kong-man(German) [21] Real-time 3-D voxel reconstruction of human motions.
- 2002 Theobalt [99] - AG-4, Stanford.
- 2002 Mikic [65] - 8 cameras, model acquisition and tracking using multi-camera voxel data, San Diego.

- 2004 de Aguiar, Theobalt, Magnor [30, 33, 32, 99, 98] various papers on skeleton estimation from 3-D voxel data obtained from multi-camera analysis.
- 2005 Cheung, Kong-man(German) [20, 24, 25] Shape-From-Silhouette (SFS) across time.
- 2006 Black and Sigal [105] - HumanEva, Brown University. Synchronised motion capture and video sequences based on four subjects performing a number of activities in a minimally constrained environment for the purposed of comparing the accuracy of markerless human motion capture algorithms.
- 2007 Sundaresan [94] - Hydra project, Maryland.

— Camera arrays

- 2004 Zhang [121] 48 cameras on a mobile platform. `http://amp.ece.cmu.edu/projects/MobileCamArray/`

2.3.1 Camera Calibration

Camera calibration is the process by which the Euclidean relationship between the image and the world is ascertained [43]. The process allows the internal (intrinsic) camera geometry and optical characteristics to be determined, such as the focal length, principle point, lens distortion and the angle between the x and y pixels. It also allows the 3-D position and orientation of the camera to be determined relative to a particular world co-ordinate system. These parameters are known as the extrinsic parameters of the camera. Camera calibrations techniques are based on various mathematical models that describe the characteristics of the prospective projection from the 3-D world to the 2-D image plane. All camera models are a specialisation of the general projective camera and can be categorised into two groups, models with a finite projective centre or with a projective centre at infinity e.g. affine camera. The camera calibration model of interest is the finite camera which at a simple level is based on the pinhole camera model. The web reference `http://www.vision.caltech.edu/bouguetj/calib_doc/` contains a list of camera calibration links with associated software [13].
In the research papers cited above, Mikic [65] used Tsai's [102] method where camera calibration is performed once when the cameras are fixed in their position. A grid of known dimension (chessboard like object) is placed in the scene - often on the floor. The world co-ordinate system for all cameras in the 3-D volume is then defined as x, y on the floor with z being the height. Sundaresan [94] performed calibration based on Tomas Svoboda's algorithm [96] which uses a standard laser pointer to calibrate a multi-camera infrastructure. Sigal and Black use the Matlab camera calibration tool kit developed by Bouguetj [13] to calibrate the cameras used to record the HumanEva videos.

2.3.2 Camera Configuration

In setting up a multi-camera environment there are a number of camera configuration issues that need to be taken into account as regards how many cameras to use and how to orientate the cameras. These considerations are highly dependent on the markerless human motion capture techniques to be applied. In this case we are considering only stationary camera configurations. The issues to be aware of are as follows:

Camera Orientation

Cameras directly facing each other would obtain redundant 2D silhouette data.

How many cameras

Good quality reconstruction of the visual hull of a human body can be achieved with as few as five cameras due to the bodies smooth shape and convex nature [58, 65].

Camera Synchronisation

The accuracy of reconstructing a 3-D volume or 3-D object within a volume are reduced if the cameras are not synchronised in time so that for each clock tick, all cameras in a multi-camera scene are viewing the subjects pose at the same instant. The accuracy is important in certain applications [37] but not necessarily in game based applications where real-time processing and the perception of accuracy is more important.

2.3.3 Example Camera Configurations

The following paragraphs outline some examples of the types of environments that need to be constructed in order to begin research in this area without the choice of the HumanEva dataset as the starting point for this research.

2007 Sundaresan - Hydra Project, Maryland [94]

Ref: `http://www.umiacs.umd.edu/users/aravinds/research/hydra.html`

The portable multi-camera capture facility consists of ten (scalable to 18) Firewire Pixelink A742 cameras attached to two workstations. The cameras can capture colour images at 105 fps with 640x480 resolution or 27fps at 1280x1024 resolution, they are synchronised using a custom-made circuit that takes a trigger signal from one of the cameras or an external source and buffers the signal to trigger all cameras. The cameras are mounted on tripods and can be positioned at heights of 1-14 feet. The cameras are wireless controlled via the 1394-based DC control library from two Linux workstations. Custom code and some Pixelink extensions manage the multiple cameras attached to the multiple PCs and provides a GUI for camera management. It is possible to capture multiple synchronised colour images at 100 fps in a laboratory setting.

2006 Sigal and Black - HumanEva, Brown [8, 91]

Ref: http://vision.cs.brown.edu/humaneva/

There are two datasets. The equipment difference between them, is the number, and type of colour camera used. The HumanEva-II video stream has 4 colour cameras that were hardware synchronised with the motion capture data as compared to the HumanEva-I video stream where the 3 colour cameras and 4 grey scale were software synchronised.

The HumanEva-I data capture was performed with two commercial video capture systems. The four Pulnix (http://www.pulnix.com/) TM6710 grey scale cameras were connected to the Spica Technology Corporation (http://www.spicatek.com/) system. Whilst the three UniQ (http:// www.uniqvision.com/) UC685CL 10-bit colour cameras were connected to the IO Industries (http://www.ioindustries.com/) system. The grey scale cameras have a 644x488 resolution and a frame rate of up to 120 Hz. The colour cameras have a 659x494 resolution and a frame rate of up to 110 Hz, that has been scaled to 640x480. "To achieve better image quality under natural indoor lighting conditions both video systems were set up to capture at 60 Hz."[8] . Optical marker-based motion capture was obtained with a ViconPeak MoCap commercial system.

2002 Mikic and Trivedi - University of California, San Diego [65, 101]

Ref: http://cvrr.ucsd.edu/pa-am/index.html

8 cameras each attached to a PC with synchronisation software in a 2 x 3 meter area. The 640x480 pixels per frame are captured at approx 10Hz and stored as images and then converted to AVI. The voxel size is 25x25x25mm and the segmentation is performed off-line.

2002 Theobalt - AG-4, Stanford [99]

Ref: http://www.mpi-inf.mpg.de/~theobalt/VisualHullTracking/index.html

4 externally triggered Sony DFW-V500 IEEE1394 cameras with a resolution of 320x240 pixels in colour mode. The maximum achievable frame rate with the external trigger is approximately 15fps. Two AMD Athlon 1GHz PC are used, each has two connected cameras.

1901 Muybridge - Stanford [74] [73]

Ref: http://americanhistory.si.edu/muybridge/htm/htm_sec3/sec3.htm

Muybridge used up to 36 lenses with 12 to 24 cameras, placed at 30-, 60-, and 90-degree angles to his subjects. The two cameras placed at 30- and 60-degrees were able to hold up to 12 lenses each. The 90-degree angle was known as the lateral, or parallel, view, while the others Muybridge referred to as the front and rear foreshortenings. With this set-up, a successful session could result in as many as 36 negatives.

2.4 Game console cameras

2008 PlayStation Eye

Ref: `http://www.joystiq.com/tag/Webcam/`

Camera resolution 640 x 480 at 60 frames/second or 320 x 240 at 120 frames/second

2.5 Environment

Particular effort has been taken in the case of the HumanEva video recordings to introduce a problematic environment rather than a clinical research environment. The room is cluttered with camera stands, objects in the room were not static between recordings e.g. the mat. The subjects are wearing normal clothing that is not necessarily distinguishable from the background in some areas. The reddish mat is also close to the RGB colour of skin. The environments in which intelligent space [101, 100] research operates is closer to the family room environment and so research in this area is worth reviewing.

2.6 Lighting

The environment used to date have strived to provide diffuse overhead lighting in order to reduce sharp shadow effects. In real room environments lighting effects from windows and lamps will create sharp deep shadows or large highlighted areas, obscuring colour and texture details. Though there is however the potential to use the shadows as additional subject information [7].

2.7 Subject Extraction and Acquisition

The first stage of markerless human motion capture is detecting the presence of a human, extracting the human from the background and assessing the characteristics of the subject : height, joint locations and body part sizes. Much of the research work assumes the presence of a human and that that human is in motion within the scene. Within a family room environment there may be multiple humans, the game player may be initially standing still waiting for the game to start and other humans, including pets may be in motion. Also there is no guarantee that the objects in the room will remain in there current location during the course of the game or between games. Consistent lighting can also not be guaranteed as curtains may be closed, room lights turned on or off or the room lighting (sunlight) may be time of day or weather dependent. The PlayStation EyeToy [109] was equipped with a light sensor to ensure

minimum light quality before commencing a game. In computer vision terms this environment would be considered a complex background similar in nature to an outdoor scene where objects such as trees are constantly in motion, lighting follows daily cycles and weather conditions occlude or hamper vision. The family room indoor environment is at least free of some weather related issues.
Early work in this area can be found in research by Bobick and Davis in 1997 - The KidsRoom [11]. The most recent environment equivalent would be the research work regarding Intelligent Spaces by Trivedi, Huang and Mikic [100, 101]. Intelligent spaces are environments such as a meeting room or a vehicle, where humans and computers collaborate. The spaces are monitored by multiple audio and video sensors that are hidden within the spaces infrastructure and allow the space to respond to specific events and triggers.
The following section reviews related work under the following headings:

— Detecting Humans

— Foreground Segmentation (Subject Extraction)

— Subject Acquisition

2.7.1 Detecting Humans

Techniques for detecting humans can be divided between techniques that require background subtraction (foreground segmentation) and those that can detect humans based on feature detection directly without preprocessing the image. Segmentation based techniques can be further decomposed into the segmentation approaches and the feature techniques then applied. Some techniques are better tuned to outdoor rather than indoor environments, or rather can be used more efficiently deployed on indoor scenes as they don't have to deal with some of the complexities of the outdoor scenes.
There are basically three feature based approaches to detecting humans within video sequences or images:

— Shape based detection, in the form of contours [83] or other descriptors e.g. Viola - face detection based on classifiers [115]

— Colour and texture based detection, for example skin colour [81]

— Motion based detection, for example gait analysis and other human specific motion patterns e.g. Cutler and Davis - periodic motion similarity [29]

— And combinations of the above, refereed to as multiple cues

2.7.2 Foreground Segmentation, Background subtraction or Matting

Foreground segmentation refers to the methods and algorithms implemented for separating the subject from the background image stream. These methods also need to take into account shadows, highlights and the movement of objects / subjects that are not the target subject.
The most basic form of background subtraction involves obtaining an image of the background prior to a human being present and subtracting the background image from the current image to obtain the differences. Hopefully the human in the scene is the only difference. But this is rarely the case, shadows, illumination changes, highlights and small movements of background objects e.g. trees, introduce additional differences that must be compensated for prior to extracting the contour or silhouette of the subject of interest.
The various approaches to background subtraction can be classified as either adaptive or non-adaptive. Absolute difference or Mean value algorithms can be considered non-adaptive as the background scene is not augmented over time to reflect changes that may have occurred in the background image. Adaptive methods are running average algorithms, alpha blending, Kalman filtering and Gaussian mixture models (GMM). Within the OpenCV [110] library, which will be used with for the purposes of this research, two adaptive segmentation algorithms have been implemented based on the following two papers.

— 2001 Kaewtrakulpong and Bowden - An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection [50], is based on an adaptive Gaussian Mixture Model.

— 2003 Li - Foreground Object Detection from Videos Containing Complex Backgrounds, is based on a Bayes decision rule for classifying foreground and background pixels based on colour and colour co-occurrence features. [62]

Porikli [81] implemented a number of segmentation algorithms with a single camera infrastructure, finding alpha-blending methods to be the fastest in terms of computational complexity. He achieved 1 msec/frame to estimate a 320x240 colour background on a 1.8Ghz machine. GMM with 3 models took 17 msec/frame but with 5 models was very slow. The Kalman filter performed at 25 msec/frame.
During Mikic [65] research into markerless human body acquisition and tracking using multi-camera voxel data she investigated the following approaches to segmentation and shadow removal.

— 2001 Cucchiara - Shadow removal based on colour and motion [82]

— 2001 Cucchiara - Shadow removal based on HSV (Hue, Saturation and Value) colour information [82]

— 1999 Horprasert Background subtraction and shadow detection [45]

Mikic found that Horprasert [45] best suited her indoor environment.
Cheung [21] in his research on real-time 3-D voxel reconstruction of human motions based on a Shape-From-Silhouette approach used colour differences between the background image and the video image to segment silhouette pixels from background pixels. The Silhouette pixels were then tested against the background pixels to assess the angular difference between their colour intensities. If the pixel is a shadow pixel then the difference is slight as the shadow reduces the intensity of the background colour. This method allowed Cheung to extract silhouettes at 30 frames per second.
Other shadow detection and removal techniques can be found in the following research work [87, 46, 39].

2.7.3 Model (Subject) Acquisition and Reconstruction

Model acquisition is defined by [65] as the estimation of the parameters of the human model being used, based on the specific size and shape of the person to be tracked, so that the model accurately reflects the position and pose of the person at the beginning of the motion capture process.
There are a number of possible approaches:

— Manual initialisation [35]

— Off-line acquisition and detailed segmentation [51] with reconstruction [84]

— Automated real-time model acquisition with limited reconstruction detail. The objective being real-time motion tracking. This methods compared the camera projections of a generic 3-D human model with the camera image plane once it has been preprocessed in one of the following ways:

  – Silhouettes [35]

  – Edges [41] or

  – Optical flows [14]

— automated real-time model acquisition and reconstruction within the 3-D domain (voxels) [21, 65]

The purpose of model acquisition is to estimate body part sizes and body orientation. Which side is left and right, back or front, effects the definition of angles at joints e.g. knee joint. The following assumptions are often made regarding the human body: the body is symmetric, left and right side body dimensions are the same, the length of the head, torso, thigh and calf should add up to the persons height.
Human body proportions are assumed to be within standard medical parameters. So given a subjects height the lengths of limbs can be calculated. For instance aspect ratios and joint lengths are used

for probability matching in Ramoser's [83] shape-based detection of humans in surveillance applications. The mean and standard deviations for these human attributes are acquired from an anthropometric database [76]

There are basically two approaches to segmenting the body model into its parts - model based approaches and non-model based approaches

Model Based Approaches

Mikic [65] segments the subjects body into six volumetric areas - the head, the torso and the four limbs by using a simple template matching, growing and fitting algorithm based on knowing average body shapes and dimensions. Bayesian networks methods based on prior knowledge of human body proportions were then used to refine the limb lengths. The width dimensions are apparent from the data that was initially evaluated via the template method. The process starts by finding the head with a spherical crust template. Then the approximate location of the neck is found by averaging over the head voxels looking for at least one neighbour that is a non-head voxel. An average torso template is then anchored like a pendulum at the neck with its axis on the centroid of non-head voxels. It is then rotated and grown into the subjects torso area. The four limb regions are found as the four largest regions of connected voxels. The hips and shoulders are used as hinge points for template fitting and growing procedures to find the upper arms and thighs. The lower arms and calves are found by locating connected components closest to the identified upper arms and thighs.

Non-Model Based Approaches

Sundaresan [94] segments the voxels into different body parts based on their connectivity without using any model information. This is achieved by transforming the voxels into Laplacian Eigenspace where they form a 1-D curves. 1-D spline fitting procedures are used to segment the nodes according to which 1-D curve they belong to, the segmented voxels are then registered to six articulated chains - head, trunk and limbs. Cheung [21] fits ellipsoid to the 3-D volume to reconstruct the subjects volume in 15 FPS.

Body part segmentation issues

Limb measurements are the least accurate as they depend on the borders between body parts which are hard to determine, for instance where does the thigh end and the calf begin. Most issues occur when the limbs are close to the body trunk.

Other errors occur in what Sundaresan [94] refers to as loop-back or self-contact poses e.g. where the hands are resting on the hips or limb ends are in contact with other parts or the body, as would be the case in Yoga or karate type poses [94].

Classical Shape-from-Silhouette (SFS) methods assume that all of the silhouette images are captured either at the same time or when the object is stationary. This assumption is violated if the object is moving

or changing shape thus SFS based approaches treat each time instant sequentially and independently. Thus to improve shape estimates more cameras are required this is an ”across space” approach. Recently [21, 23, 23] algorithms have been devised that use an ”across time” approach where the number of silhouettes collected is increased by taking multiple silhouettes from a camera over time while the object is moving. By compensating for the motion of the object a visual hull can be constructed that is equivalent to the hull produced from a larger number of cameras.

SFS is best suited to indoor locations where the cameras are static and there there are few moving shadows. SFS is also highly sensitive to errors in silhouette contours and the calibration of the cameras.

Dynamic Models and A Model Initialisation Phase

Kakadiaris [51, 52] used 3 mutually orthogonal cameras and had the subject perform a set of predefined motions that incrementally revealed the structure of the subject. A single deformable model was used to fit the apparent body contour, the model deforms over time to fit the changes. Physics based tracking was used to obtain the difference between the predicted and actual image, then forces calculations were applied to the model to deform it as required. This is a dynamic model based approach. Dynamic models yield good results but rely on strong prior assumptions on the type of motion. [90, 35]

2.8 Subject Modelling

Pose estimation is a process of identifying how a subjects body and/or individual body parts are positioned in a scene [67]. There are different approaches to pose estimation and reconstruction some use models others do not. Pose estimation processing is based on the tracking data obtained following segmentation of the subject from the background. The pose can be estimated at various levels of accuracy, the required accuracy is dependent on the application which may just require the subjects centre of mass e.g. tracking applications. Medical applications require the exact measurements of every limb and its precise world co-ordinates. Using a model allows the system to be tuned to the characteristics of the subject e.g. physical attributes, motion characteristics, and to incorporate this knowledge into its processing routines. Thus the need for model acquisition.

An overview of 2-D and 3-D approaches to human modelling can be found in the surveys by [1], [40], [118], these surveys provide a taxonomy of the approaches - models and non-models.

No Model

Many approaches to 2-D or 3-D interpretation of human motion focus on the joints of the body and tracking them between frames. To do this without a priori shape model requires heuristic assumptions to be made in order to establish the correspondence of joints in an image sequence.[1]

Using Models

If the intention is to build a robust motion capture system that produces physically accurate posture estimates then the choice of subject modelling framework is extremely important. The physical constraints associated with the subject e.g. a human, can be used to refine the tracking or constrain the model being used. Without the use of a model it is hard to establish feature correspondence between successive frames [1].

So most of the tracking algorithms depend on a dimensional model of the subject that is initialised to the subjects first pose and compared with subsequent poses to validate the foreground segmentation process and to provide pose estimates. Many different model types have been proposed for this purpose.

Modelling Humans

— Tapered superquadrics [41, 94].

— Spheres, truncated cones and parallelepipeds [35] .

— Ellipsoids, Blobs [119].

— Truncated cones connected to joints by rigid links.

— Ellipsoids, cylinders and twists framework [65].

— Rigid body segments attached to each other in articulate chains [94].

These models rely on knowledge of the human body as defined by [65]:

— Components : head, torso, limbs. Though the definition of a body component may depend on the level of detail required.

— Structure : how the body parts are connected and the location of joints

— Joint rotations : the axes of rotation for different joints e.g. the degrees of freedom.

— Joint angles : the range of possible angles associated with each joint and each degree of freedom

Human Model Types

Aggarwal [1] further decomposes the models into the following types:

— Geometric models: stick figures based on the movement of the bones in an underlying skeletal structure. 2-D contours based on the projection of a human on a 2-D image, 3-D volumes in the form of cones, ellipses and spheres.

— Motion models: these models use points, 2-D blobs or 3-D volumes and the nature of the model depends on whether one or multiple cameras are used.

— Articulated human body models: articulated human body model are constructed of super-quadric body segments that are attached to each other at joints. The segments are connected to each other in a kinematic chain that allows three degrees of rotation at all joints except for the shoulder joint. The shoulder joint is complex and so is allow three degrees of limited translation as well as rotation to better model the complexity of the joint [94].

2.9 Motion Tracking

Tracking is about recognising the same object in multiple frames.
Sminchisescu [93] nicely sums up the different methods as follows: ”Several criteria can be used to classify human body tracking methods: the representation used to model body parts (cylinders, quadrics, cones, deformable), and kinematics (Euler angles, twists, independent parts); the image features used (edges, intensities, silhouettes); or the motion models (noisy Gaussian dynamics, linear auto-regressive processes, dedicated walking or running models); the parameter estimation method (single or multiple hypothesis methods).”
Tracking can be performed in the 2-D domain, based on pixels on the image plane, or in the 3-D domain based on voxels or depth/range information. Pixel based methods suffer from drift as they do not use absolute features when performing tracking. Though they are better able to deal with rotation about the axis of the body segment[94].
The various tracking methods used to-date are summarised below:

— Multi-camera image plane analysis: simplified camera models are used to project the model to the image plain for comparison with expected poses [41, 35].

— Multi-camera Voxel analysis: The subject is reconstructed in 3-D, the body parts segmented and labelled. The position of predefined measuring points is then predicted via an extended Kalman filter (EKF) which updates the model [65]. There is no need to perform comparisons in the image plane as the voxel data is in the same dimension as the subjects real body and is mapped to the subjects body dimensions.[21, 27]

— Optical Flow and Intensity: Human body contours are computed based on on optical flow and intensity. The 3-D model is composed of shapes - spheres, truncated cones. The model is projected to the image plane and aligned to the contours via forces.[35]

— Expectation Maximisation with valid Kinematic Structure [47].

— Hidden Markov Models and the EM algorithm [14].

— Principle Component Analysis: used to find the principle axis of a person. A constant acceleration kinematic model predicts the positions of body parts in the next frame. The projected models contour locations are adjusted using undirected normalised chamfer distance between its contours and the image contours[41].

— Physics based Synthesis: A single deformable model is used to fit the apparent body contour of the subject as they move. Forces are calculated that deform the model to the new pose.[64, 51, 52]

— Twists and Exponential Maps: The products of twists and exponential maps are used to describe the correctness of body parts and relative motion of parts connected by joints [14].

— Dynamic Bayesian Network.

— Probability [119].

— Condensation Algorithm (Particle Filtering) [36].

— Stereo Depth Data [28, 79].

— Tracking using multiple cues: Motion and structural cues [94].

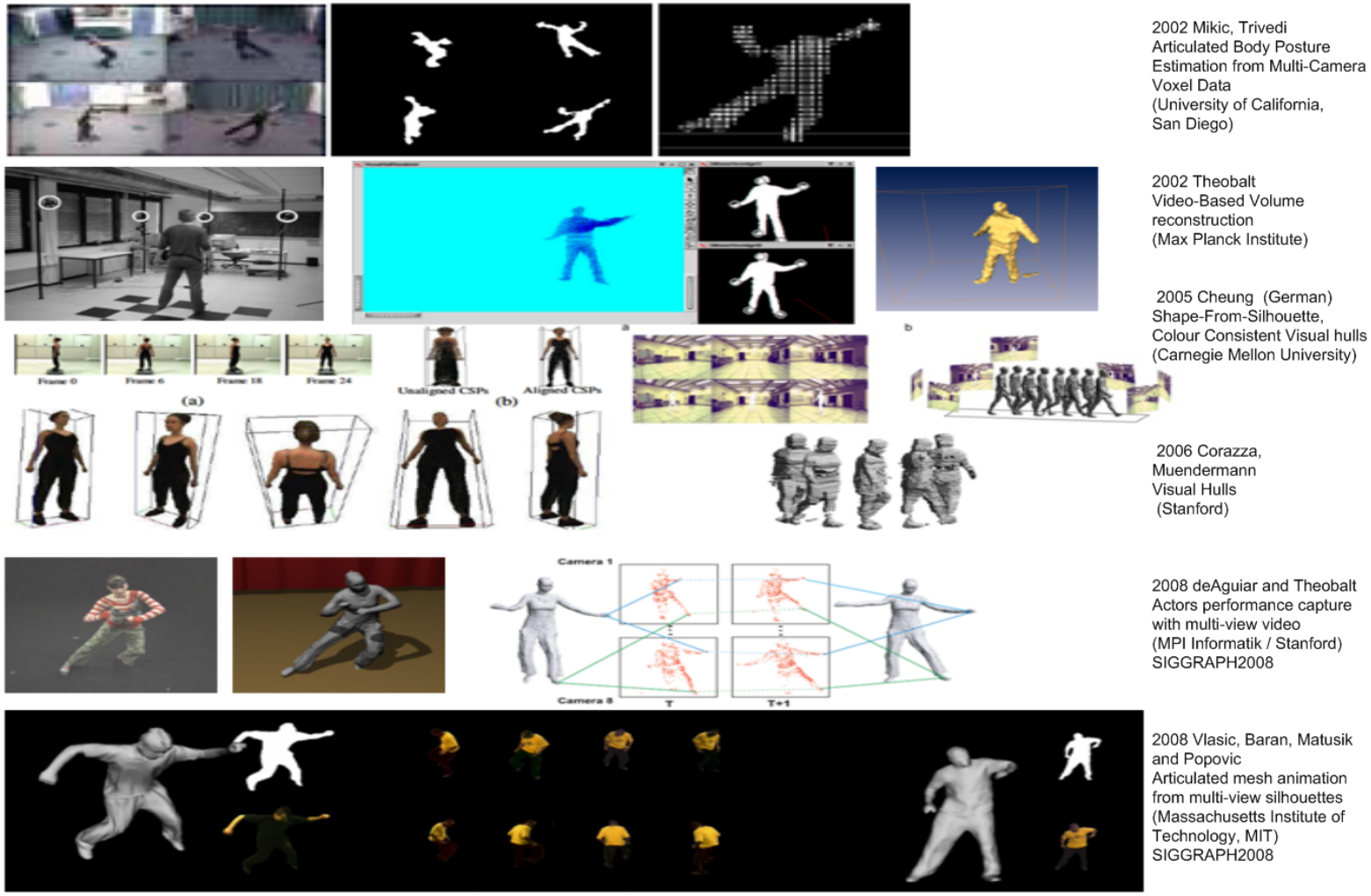

Figure 2.2: *The state of the art markerless human motion capture systems*

2.10 Summary

Many techniques, algorithms and man years of research are associated with the multi-camera human motion capture research area. The key papers and researchers that have incorporated this research into working systems is summarised below:

— 2002 Mikic [65] - Voxels (University of California, San Diego)

— 2002 Theobalt [99] - Volume reconstruction for video-based human motion capture (Max Planck Institute)

— 2005 Cheung [23, 22] - Shape-From-Silhouette, Visual hulls (Carnegie Mellon University)

— 2006 Corazza, Muendermann [70] - Visual Hulls (Stanford)

— 2007 Sundaresan [94] - Maps each voxel into the Laplacian Eigenspace (LE) (University of Maryland)

— 2008 deAguiar and Theobalt [31] SIGGRAPH2008 - Actors performance capture with multi-view video (MPI Informatik / Stanford)

— 2008 Vlasic, Baran, Matusik and Popovic [116] SIGGRAPH2008 - Articulated mesh animation from multi-view silhouettes.

Chapter 3

Theoretical Basis

Research work [65, 99, 23, 70, 94, 31, 116] in this area is extensive. It has evolved out of years of work, serial collaboration and extensive studio facilities. This research and its associated implementation is only the tip of the iceberg in terms of the full processing pipeline required to create a game avatar from multiple cameras in real-time. That said, the state of the art research builds on, refines and optimises the fundamental principles, processes and knowledge infrastructure presented here.
This research takes a multi-camera shape from silhouette approach to subject shape reconstruction and 3-D skeleton estimation based on the following primary material:

— The Camera Model

— Camera Calibration

— 3-D Voxel Reconstruction (Shape-From-Silhouette)

— Voxel Colouring

— The 3-D to 2-D projection calculation

— Background Subtraction (Foreground Segmentation)

— Shadow Removal

The approach taken was to build a simple end-to-end processing pipeline based on the above which exercises and validates the implementation of the basic algorithms and principles. This pipeline will take the multiple video streams and camera calibration data from the HumanEva I and II datasets [91], segment

the subject from the video, generate silhouettes, construct a 3-D visual hull for the subject and infer 3-D shape and skeleton. Once the basic pipeline has been modelled in MATLAB and implemented in C++/OpenCV [110] more sophisticated algorithms and optimisations can be implemented. It is intended that MATLAB and/or the OGRE [112] graphics engine will be used for visualisation.

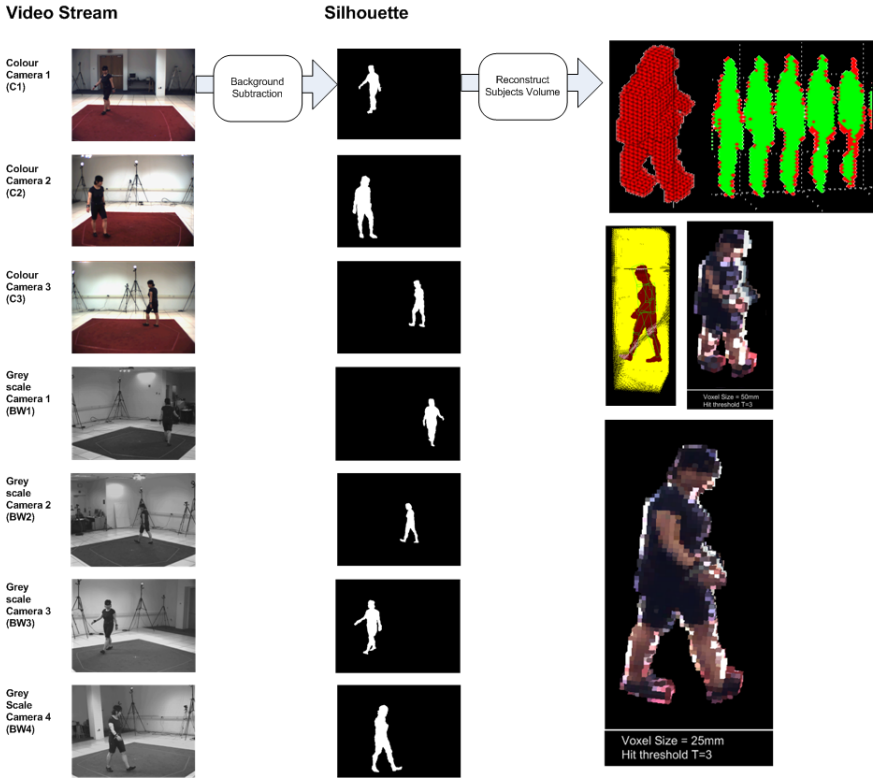

Figure 3.1: *HumanEva I, Subject S1, Walking, Frame 91*

The skeleton inference mechanism is unique to this research. The basic concept is metaphorically similar to the way an inverse 3-D Radon transform would reconstruct the internals of a body from an MRI type scan. Let me explain:

A visual hull is constructed based on all cameras in the scene concurring that a particular world volume (voxel) as occupied by the subject is within the silhouette image of the subject as seen by that camera. So if a cubic volume of space, as defined in world [X;Y;Z] co-ordinates relative to the cameras location, is projected back to a pixel location [x;y] on the 2-D image plane captured by a camera, and the pixel is part of the subject then a hit count for that voxel location is augmented by one. If four cameras are uses then a voxel with a hit count of four is considered to be within the volume of the subject. Hit counts of

3 or less are background voxels. So in the case of the HumanEva dataset the spacial volume is 8 x 8 x 4 meters, within this space there is a subject occupying approximately a volume of 1.8 x 1.8 x 1.8 meters (arms out wide, a persons width equals their height). The space occupied by the subject is carved out of the volume seen by all cameras. The visual hull is a representation of the maximum space that the subject occupies. If the voxel size is small and a highly refined algorithm for occupancy testing is used, a fine grain visual hull can be obtained. The smaller the voxel size and the more refined the occupancy test the higher the computational expense. The occupancy test is often an approximation e.g. Cheung (German) SPOT algorithm [21] since the volume cube projects to a diamond shape in 2-D. Thus there is occupancy ambiguity at silhouette edges as regards what percentage area of the pixel footprint, as cast by the voxel within the silhouette, constitutes a hit. The greater the number of cameras, at various angles the better the potential subject volume reconstruction but again the greater the computational expense, though the research area is prime for multi and many processor optimisations [42].

The images from X-ray Computerised Tomography (XCT), Positron Emission Tomography (PET) and Magnetic Resonance Imaging (MRI) scans are cross sections of a body there the darker regions of the images are areas of greatest ray absorption, corresponding to the core organs / bones of the body. Metaphorically speaking there is a similarity between the hit count for a voxel, which represents how many cameras concur that this voxel is part of the subjects volume and the volume occupancy density maps generated from MRI data. The voxel hit density could be considered a probability counter as regards the subjects occupation of that space rather than just a yes/no answer as regards occupancy or not. The denser hit counts voxels would therefore represent the areas of greatest certainty as regards occupancy which would equate to the core skeleton of the subject. Whilst one of the goals of this research is to review the potential of using multiple cameras to animate an avatar in a game, a secondary goal is the recover of the 3-D skeleton from multiple 2-D video streams.

3.1 Notation

The inline matrix notation used is as per MATLAB(R) programming syntax:
[x;y] refers to a 2 by 1 matrix as used to represent pixel 2-D co-ordinates.

$$[x;\ y] = \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.1}$$

[X;Y;Z] refers to a 3 by 1 matrix as used to represent 3-D XYZ co-ordinates.

$$[X;\ Y;\ Z] = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \tag{3.2}$$

3.2 The Camera Model

The perspective camera model provides a mathematical model which relates how 3-D locations in world space co-ordinates [X;Y;Z] are projected to pixel co-ordinates [x;y] in a 2-D image plain [43, 13]. The mapping is an approximation and the degree of error depends on the camera model used. The simplest model is the pinhole camera model. This model forms the basis of the model used in this implementation.

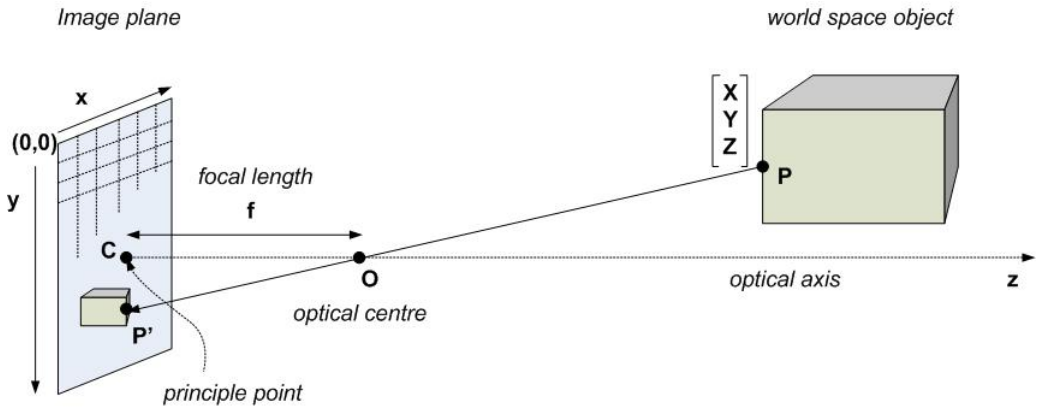

Figure 3.2: *pinhole camera model*

O is the camera pinhole or optical centre; point *P* is an object point that is represented by a pixel point *P'* on the image plane. From the cameras point of view the *Z* co-ordinates are along the optical axis, either *Y* or *X* can represent the camera up axis, it depends on the co-ordinate system in use. *I* is the image plane that is perpendicular to the optical axis.The focal point, f , is the distance between the optical centre and the principle point *C*. The principle point is the point where the image plane touches the optical axis.
If the focal length f and the world co-ordinates *X*,*Y*,*Z* of a point *P* are known then it is possible to calculate the pixel co-ordinates x , y that represent its image *P'* on the cameras image plane. P' projection onto the image plane in a simple pinhole camera model is given by:

$$x = f\frac{X}{Z} \tag{3.3}$$

$$y = f\frac{Y}{Z} \tag{3.4}$$

And so the vector representing the pixel co-ordinates is:

$$\vec{p} = \begin{cases} x = f\frac{X}{Z} \\ y = f\frac{Y}{Z} \end{cases} \tag{3.5}$$

By switching to homogeneous coordinates the equations can be written in matrix form. Homogeneous coordinates are a method of representing 2D points by 3D vectors and 3D points by 4D vectors. Thus allowing points at infinity to be represented and manipulated. Also some non-linear perspective projection transformations become linear in this over parameterised space and are hence easier to manipulate mathematically [43]

To conversion to homogeneous coordinates an extra scaling coordinate w is added to a point and all other coordinates x, y, z need to be divided by w to get normal 3D coordinates out of a 4D homogeneous vector. So the vector [x;y] becomes:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{3.6}$$

Equations 3.3 and 3.4 in matrix form would therefore be represented as follows:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.7}$$

Extending this model to account for how a real camera projects world co-ordinates to it image plane means adding other factors to the model. In a real camera the pixels are never exactly square hence there is a separate focal length in the x direct and y direction. Two factors k and l expressed in $\frac{pixels}{cm}$ are multiplied by the focal length f to obtain f_x and f_y where $f_x = kf$ and $f_y = lf$. Another property of a real camera is that the origin of the co-ordinate system is generally by convention placed at one of the corners of the image not at the image centre (principle point). The offset factors u_0 and v_0 translate the principle point to the required origin location, normally the top left corner of the image. So the vector representing the pixel co-ordinates becomes;

$$\vec{p} = \begin{cases} x = f_x \frac{X}{Z} + u_0 \\ y = f_y \frac{Y}{Z} + v_0 \end{cases} \tag{3.8}$$

Another feature of a real camera is that the angle between the *x* and *y* axes on the image plane is not 90 degrees. This is refereed to as *skew*, it occurs when the camera CCD (the charge-coupled device or image sensor, note - there are a CMOS photoelectric sensors) is not perpendicular to the optical axis. A skew co-efficient that defines the angle between the *x* and *y* pixel axes is normally denoted as α. The resulting matrix that takes account of these factors is known as the intrinsic camera matrix M_{int} it has the following form:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x & \alpha * f_x & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.9}$$

The intrinsic matrix is often refereed to as a *3x3* matrix but the 3x4 version is required for any type of matrix manipulation. The intrinsic parameters of the camera do not depend on the cameras position in the world.

The origin of the world co-ordinates [0;0;0], as the model now stands, lies at the pinhole *O*. It is more useful to represent the location of the objects in the world with respect to the location of the camera. To do this a rotation matrix (3x3) and a translation vector (3x1) are introduced to convert the world co-ordinates to camera co-ordinates. The combined rigid body transformation matrix is refereed to as the extrinsic camera matrix M_{ext}. The 3-D to 2-D projection formula thus becomes:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x & \alpha * f_x & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.10}$$

The translational vector $[t_x;\ t_y;\ t_z]$ is equal to the position of the camera in world co-ordinates and the rotational matrix specifies the orientation of the camera.

There is however one other additional consideration in terms of the properties of a real camera - radial distortion due to the shape of the lens used. The distortion is modelled in 2-D and chosen such that a distortion applied to co-ordinates *x* and *y* can be undistorted again. The HumanEva dataset uses the Bouguet[13] tool set for camera calibration and therefore it is modelled on the 1966 Brown 'Plumb Bob' or "Brown-

Conrady" camera distortion model [16]. The model has the following parameters $r^2 = x_n^2 + y_n^2$ and a 5x1 vector of parameters - $kc_1, kc_2, kc_3, kc_4, kc_5$. Where x_n and y_n are the normalised *x* and *y* co-ordinates as follows – if *P* is a point in space as defined from the cameras reference frame to be at co-ordinates $[X_c;\ Y_c;\ Z_c]$, then the normalised (pinhole) image projection is:

$$\vec{p_n} = \begin{cases} x_n = \frac{X_c}{Z_c} \\ y_n = \frac{Y_c}{Z_c} \end{cases} \tag{3.11}$$

The x and y co-ordinates, x_d,y_d produced by the application of the distortion formula are as follows:

$$\vec{p_d} = \begin{cases} x_d = (1 + kc_1 * r^2 + kc_2 * r^4 + kc_5 * r^6) * x_n + 2kc_3 x_n y_n + kc_4(r^2 + 2x_n^2) \\ y_d = (1 + kc_1 * r^2 + kc_2 * r^4 + kc_5 * r^6) * y_n + kc_3(r^2 + 2y_n^2) + 2kc_4 x_n y_n \end{cases} \tag{3.12}$$

The application of these formula's is described below in section 3.6.

3.3 Camera Calibration

Camera calibration is the process by which the intrinsic matrix, extrinsic matrix and distortion parameters of a camera are determined. With the camera parameters it is possible to model the translation of an object from 3-D world space to the 2-D image plane as illustrated in the figure below.

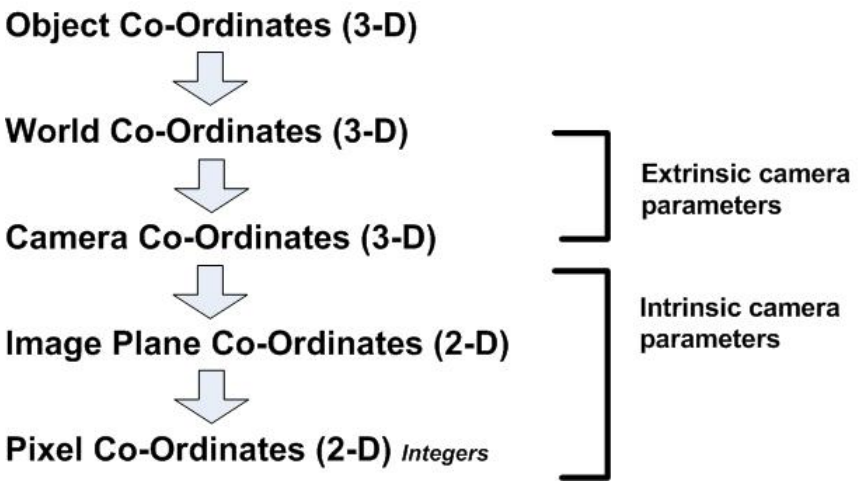

Figure 3.3: *The five reference frames required for 3-D scene analysis [10]*

The HumanEva cameras were calibrated based on Bouguets [13] MATLAB toolkit. The calibration parameters are available in a .CAL file in the 'Calibration Data' directory under each subject. An example and the associated format is provided below:

$Value$	$Variable$	$Name$	$Description$
833.719953	$fc(1)$	f_x	$focallengthX(2x1)$
834.599420	$fc(2)$	f_y	$focallengthY(2x1)$
309.976478	$alpha_c$	$\alpha * f_x$	$skewco - efficient(1x1)$
246.584829	$cc(1)$	u_0	$principlepoint(pointoforigin)$
0.000000	$cc(2)$	v_0	
-0.150146	$KC(1)$	kc_1	$distortion(5x1)$
0.475480	$KC(2)$	kc_2	
-0.001676	$KC(3)$	kc_3	
-0.005027	$KC(4)$	kc_4	
0.000000	$KC(5)$	kc_5	
-0.637589	$RC_EXT(1)$	r_{11}	$Extrinsicrotationmatrix(3x3)$
-0.769520	$RC_EXT(2)$	r_{12}	
0.036325	$RC_EXT(3)$	r_{13}	
-0.193474	$RC_EXT(4)$	r_{21}	
0.114306	$RC_EXT(5)$	r_{22}	
-0.974424	$RC_EXT(6)$	r_{23}	
0.745686	$RC_EXT(7)$	r_{31}	
-0.628310	$RC_EXT(8)$	r_{32}	
-0.221762	$RC_EXT(9)$	r_{33}	
297.644243	$TC_EXT(1)$	t_1	$Extrinsictranslationvector(3x1)$
720.990963	$TC_EXT(2)$	t_2	
5241.870276	$TC_EXT(3)$	t_3	

3.4 3-D Voxel Reconstruction

A Voxel is essentially a three-dimensional space within the world volume observed by a camera. The world volume can be considered to be composed of voxels each of which contributes to a pixel on the cameras image plane. The size of a cube that defines a voxel's spacial enclosure is determined by the needs of the application - the level of granularity required and the associated computational overhead that

the application is willing to expend on processing the volume that consists of the unit voxels. Mikic [65] used voxel sizes of 50mmx50mmx50mm and 25mmx25mmx25mm in her research. The term voxel is arrived at by considering these unit spaces to be volume pixels.

Voxel Reconstruction is a technique for reconstructing a 3 dimensional representation of an object based on images of the object obtained from multiple cameras. The volume imaged by the the multiple cameras is divided into unit volumes or voxels. In order to reconstruct a 3-D volume, a model of the imaging process as it relates to each of the cameras is required e.g. the camera calibration parameters. This allows the positions in 3-D space to be mapped to image positions in 2-D space (the image plane). Many voxels will contribute to a pixel, the closest voxel to the camera will occlude the other voxels.

The general method used by the predominant researchers [65, 99, 23, 70, 31, 116] in this area is shape from silhouette. From each camera the silhouette image of the subject is extracted. The silhouette is the occluding contour of the subject in the image and it contains some information about the 3D shape of the subject. With a single silhouette image of the subject, we know that the 3D shape of the subject lies inside the volume generated by back projecting the silhouette area using the calibration properties of the camera. The problem with back projection is that each pixel will return a multiple of voxel positions, defined by a cone within the volume, that contributed to that pixel. If the voxel world co-ordinates are projected into the the 2-D image plane there is potentially an area of pixels that the voxel will project to.

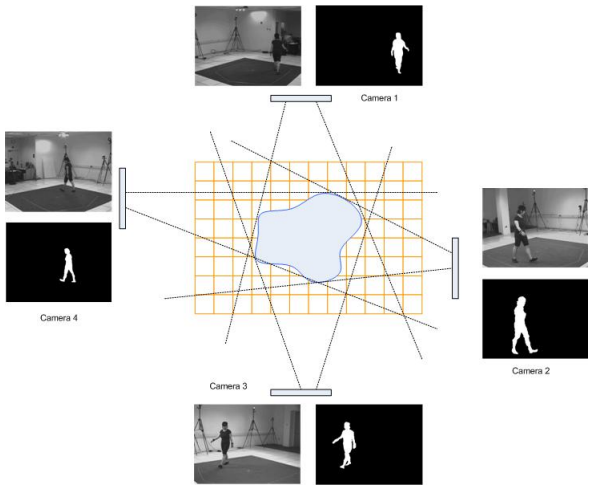

Figure 3.4: *Shape from Silhouette*

So the volume of interest is split into voxels and each voxel 3-D vertexes are projected into the 2-D image of each camera, if they are within the pixel range of the cameras sensor then they form a diamond shaped

footprint. If the footprint in each image is within the silhouette of the subject in all camera images the voxel is marked as occupied, if not, it is marked as un-occupied. The volume of the subject is thus carved out of the volume creating a visual hull [57, 59] of the subject.

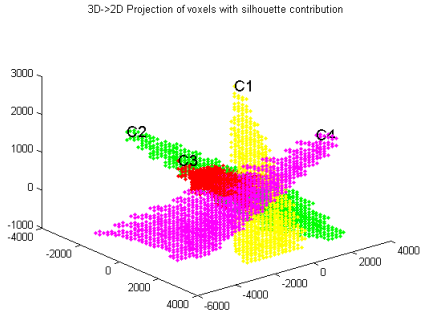

Figure 3.5: *A plot for each camera, showing the voxels that contribute to the silhouette of the subject*

Volumetric modelling of scene space assumes that there is a known, bounded area in which the object of interest lies[37]. There are many techniques and optimisation for volume based reconstruction, see survey [92]. If stationary cameras are used then the voxel to 2-D projection does not have to be computed each time for each camera instead a lookup table can be created which decreases the computation required considerably. It is also possible to construct a lookup table for pan/tilt cameras [65] where a part of the projection calculation is precomputed and stored in a lookup table for each camera.

In general it is impossible to accurately reconstruct an arbitrary volume using a finite number of cameras. The visual hull is the closest approximation that can be obtained by volume intersection. Where the visual hull is defined as the (maximal) object that can reconstruct the original objects silhouette from any viewpoint. [58, 56, 57, 59]. Good quality reconstruction of the visual hull of a human body can be achieved with as few as five cameras due to the bodies smooth shape and convex nature[58, 65].

Instead of iterating through each unit voxel in the volume of interest another approach is the use of an octree data structure for defining the volume and its voxels [37, 97]. The volume commences as a single cube, it is then subdivided into eight cubes. Each cube is checked to see if contains the object to be reconstructed, if not, then no further subdivisions are performed on that cube space. When all cubes have stopped dividing because the algorithm has reached the minimum voxel size specified then the resulting data tree contains a representation of the 3-D shape being reconstructed.

There is a probabilistic approach to assigning voxel in the volume of interest to ensures that no holes are carved in the model. Each voxel is assigned a probability based on comparing the likelihoods for the

voxel belonging or not belonging to the object. [15]

3.5 Voxel Colouring

Voxel colouring is a method for reconstructing the shape and colour of a 3-D object. It was first introduced in 1999 by Seitz and Dyer [88]. The method is similar to voxel reconstruction but with an additional step to reconstruct a photo-hull where the colour in the silhouettes of each image are compared in order to find an appropriate colour for the voxel. If a Lambertian reflection model is assumed then the colours from the various camera angles can be compared. The Lambertian reflection model says that all objects in the scene reflect the light equally in all directions.

In order to handel occlusion there is a special voxel traversal order. Voxels closest to the camera are visited first to ensure that a visited voxel cannot be occluded by an unvisited voxel. Where a voxel is found that is part of the subjects volume and its colour across all images is found to be consistent, the mean colour of the collection of pixels is used to colour it and the pixels that contributed are marked off from further use. The voxels projected footprint into the image plane is often bounded by a bounding box to approximate the footprints shape rather than using scan-conversion techniques to obtain the actual area and properties of the 8 projected points (vertices) of the voxel.

Voxel colouring is one technique for obtaining the a textured representation of the subject. The more advanced techniques in this area can be found in the work by Vlasic [116] where a full textured 3-D subject pose can be obtained at a speed of 16 seconds per frame.

3.6 The 3-D to 2-D projection calculation

The basis of reconstructing the volume of the subject from multiple camera views is the 3-D to 2-D projection calculation which is composed of the following steps (see MATLAB code in Appendix B):

Step 1 - Transform the World XYZ co-ordinates to the camera co-ordinate frame using the parameters in the extrinsic camera matrix as follows:

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.13}$$

$$T = \begin{pmatrix} t_x \\ t_y \\ t_z \\ 1 \end{pmatrix} \tag{3.14}$$

$$XYZ_{world} = \begin{pmatrix} x_{world} \\ y_{world} \\ z_{world} \\ 1 \end{pmatrix} \tag{3.15}$$

$$XYZ_{camera} = R * XYZ_{world} + T \tag{3.16}$$

Step 2 - Convert to 2-D pinhole camera co-ordinates:

$$\begin{aligned} x_{normalised} &= \frac{X_{camera}}{Z_{camera}} \\ y_{normalised} &= \frac{Y_{camera}}{Z_{camera}} \end{aligned} \tag{3.17}$$

Step 3 - Apply the distortion equations 3.12:

$$\begin{aligned} x_{distorted} &= (1 + kc_1 * r^2 + kc_2 * r^4 + kc_5 * r^6) * x_n + 2kc_3 x_n y_n + kc_4(r^2 + 2x_n^2) \\ y_{distorted} &= (1 + kc_1 * r^2 + kc_2 * r^4 + kc_5 * r^6) * y_n + kc_3(r^2 + 2y_n^2) + 2kc_4 x_n y_n \end{aligned}$$

Step 4 - Convert to pixel co-ordinates by applying the parameters in the intrinsic camera matrix:

$$\begin{aligned} x_{pixel} &= f_x * (x_{distorted} + (\alpha * f_x) * y_{distorted}) + u_0 \\ y_{pixel} &= f_y * y_{distorted} + v_0 \end{aligned} \tag{3.18}$$

3.7 Background Subtraction

Background methods and techniques are discussed in detail in section 2.7.2. In order to perform voxel based subject reconstruction we need to generate clean subject silhouettes in real-time. Shadow removal is only one of the problems. Depending on the complexity of the environment, the cloths being worn by the subject and the degree of movement involved the silhouette images can be extremely noisy. Whilst morphology and connected component analysis can be used to clean up these noisy images they incur a computational overhead. Whilst the exactness of silhouette extraction is important where measurements are to be performed on the subjects motion, in games perceived accuracy and responsiveness is more important.

In the HumanEva dataset the environment and the cloths worn by the subject are deliberately intended to be noisy. The mat on the floor of the room changes position throughout the span of the video recording sessions. The dataset therefore includes three empty background sequences, one from the start of the session, one from the middle, one from the end.

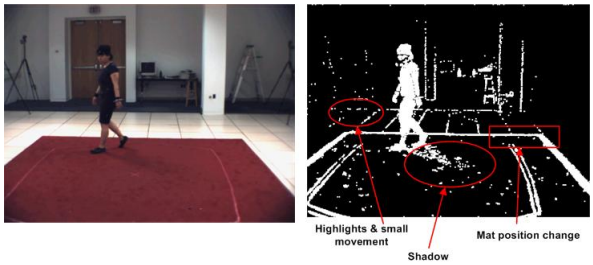

Figure 3.6: *A HumanEva subject silhouette derived from a simple absolute difference between the current video frame and the first frame of the first empty background video*

The simplest means of background subtraction is to take an image of the background without the presence of the subject or any moving object and then to subtract this image from the current image containing the subject. A threshold is then applied so that only the pixels with a significant difference between the background and foreground are selected as the potential foreground pixels. There are many more sophisticated methods [62, 81, 50, 89]. The key objective is to obtain a good silhouette in an acceptable time frame. With this objective in mind some of the basic background subtraction algorithms will be implemented in order to assess the potential issues in the HumanEva data prior to implementing more sophisticated algorithms. The methods to be tested are:

— Absolute difference based a static background frame.

— Mean and standard deviation difference test in RGB space based on a static background obtained over a number of video frames.

— Running Average based on a dynamically changing background.

— Running Average based on a dynamically changing background but with initial averaging performed from the background video.

There are two background subtraction functions implemented in OpenCV which will also be implemented.

— CvGaussBGModel() - based on P. Kaewtrakulpong and R. Bowden research [50].

— CvFGDStatModel() - based on Li et la. research [62].

OpenCV also contains a running average function - cvRunningAvg(). This algorithm learns the background model which makes it very suitable for environments where the background constantly changes or is different each time the cameras are turned on e.g. a family room. The background image is first initialised to all zero values. Then, for every pixel in each of the selected video frames, the average background image is updated using the update equation as shown below:

$$B(x,y) = (1-a) * B(x,y) + aI(x,y) \tag{3.19}$$

Where *B* is the accumulated background image and *I* is the current image, *0¡=a¡=1* is the learning rate and *(x, y)* are the 2D image co-ordinates.

3.8 Shadow Removal

In segmenting the foreground from the background by accumulating an image of the background or by using a static background model, the result is the same. To a lesser or greater extent there is noise. Shadows, light reflection highlights, small background movements and non subject movements. Shadows produce the largest amount of noise and so to achieve a good subject silhouette they need to be removed. Once the basic background subtraction has taken place the extracted foreground needs to be tested for shadow pixels[81, 21]. Shadows do not change the hue of the objects they project on to, instead they decrease the luminance and change the saturation. So the luminance and the colour saturation difference between a background pixel and a foreground pixel can be tested based on a threshold to establish if the pixel is in fact a background pixel in shadow rather than being part of the foreground.
Given an image composed of *i x j* pixels and a background image also composed of *i x j* pixels. A colour pixel on the image is denoted as $cp_{image}[i,j]$ and a colour pixel on the background is denoted as $cp_{background}[i,j]$. The colour difference between the two pixels is given by:

$$\Theta = \cos^{-1}\left[\frac{cp_{image}[i,j] \cdot cp_{background}[i,j]}{\|cp_{image}[i,j]\|\|cp_{background}[i,j]\|}\right] \tag{3.20}$$

if $\Theta > T^c$ where T^c is the colour angle threshold defined for the difference between a shadow background pixel and a background pixel.

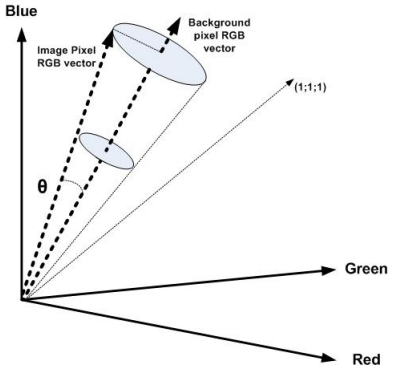

Figure 3.7: *Shadow as a conic around a background pixel's colour vector in the RGB colour space*

Chapter 4

Implementation Overview

4.1 Overview

The implementation presented here focuses on examining the HumanEva I / II datasets and building a simple multi-camera processing pipeline to allow the 3-D reconstruction of the subject based on volume scene reconstruction techniques [37]. The algorithms and methods to be used were first implemented in MATLAB and then in C++ / OpenCV [110]. MATLAB and a free voxel viewer (vxlview) [107] were used to visualise the subjects following reconstruction. The initial intention was to use the OGRE graphics engine for visualisation and the vxlview as the ground truth. But extensive time spent on the analysis of the HumanEva dataset meant that the OGRE [112] graphics engine work was not completed in time.

An agile [108, 103, 95] development approach was taken to the software implementation. Functionality was developed along side tools and tests for verify the results. With each iteration the code was refractored and retested with the developed tools and tests.

4.2 HumanEva Code and Ground Truth

The HumanEva dataset contains MATLAB software for viewing the multiple video streams and superimposed motion capture data.

HumanEva I and II datasets and the software were installed on an Intel Quad Core 2.40GHz CPU PC running Windows Vista Home Premium. The PC was equipped with 2GB of RAM and an NVIDIA GeForce 8600 GTS, Graphics Processing Unit (GPU), 256 MB, 32 shaders. The implementation does

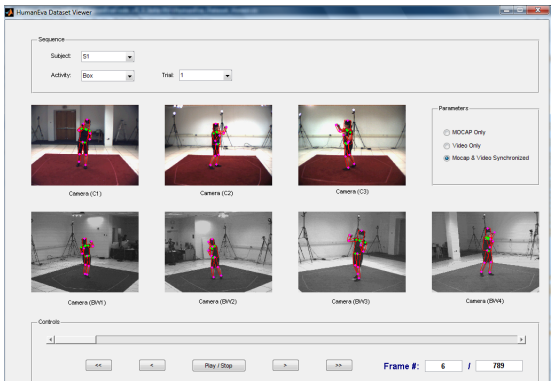

Figure 4.1: *HumanEva I, MATLAB Viewer*

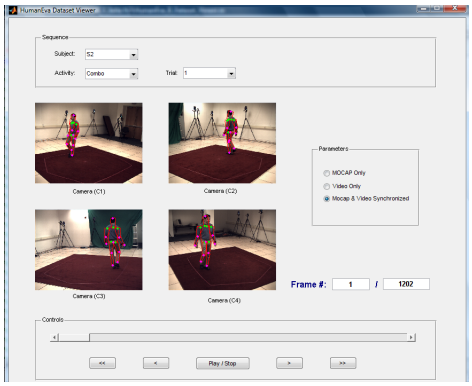

Figure 4.2: *HumanEva II, MATLAB Viewer*

not avail of the multiple processors or the GPU.

The installation required a XVID [113] codec v. 1.1.0 in order for the HumanEva I videos to be viewed within MATLAB and by OpenCV. The code and data must be installed on the root directory e.g. C://HumanEva I and C://HumanEva II or the HumanEva demo viewer will not work. The ”HumanEvaCode v1 1 beta” directory contains a read me which details the installation process.

The cameras in the HumanEva recording environment were calibrated using Jean-Yves Bouguet [13] camera calibration toolkit for MATLAB. The HumanEva MATLAB code makes extensive use of that codes implementation of the camera matrix for 3-D to 2-D projection of the motion capture data. With the end goal of this research being to implement in C++ / OpenCV it was necessary to write a stand alone 3-D to 2-D projection function. The HumanEva code was therefore used as a ground truth for that function where motion capture data (joints and cylinders) were projected using both projection functions

to ensure compatibility.

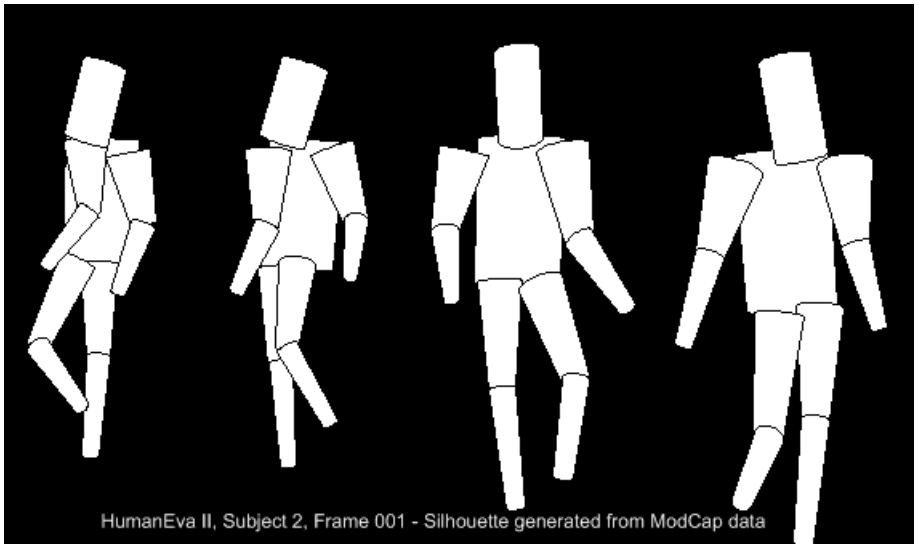

Figure 4.3: *HumanEva II, Silhouettes generated from the motion capture data*

An additional ground truth for the volume reconstruction was the creation of fake silhouettes from the motion capture data. These silhouettes were then used to construct a subject volume. The resulting .vxl files (the file format used by the vxlviewer) were then compared with the .vxl files generated by the actual subjects. The differences provided an indication of the accuracy of the volume reconstruction.

4.3 Build layout

The Code directory contains three primary directories - Data, MCAvatar and Results.
The Data directory contains the actual HumanEva I / II data and equivalent directories HumanEva_I_subset, HumanEva_II_subset. The subset directories contain the extracted data used for the analysis e.g foreground segmented frames, ground truth frames etc. This ensures that the original HumanEva data are not changed by the analysis process.
The Results directory is broken down into four primary directories, each signifying the four major versions of the software. Within these directories there are directories for each analysis run. The later analysis run directories all contain report files that record the settings used e.g. volume co-ordinates, voxel size and the name of the program that created them.
The MCAvatar directory contains the C++ and MATLAB code developed through out the analysis. The

four phases of the MATLAB analysis code have their own directory and the C++ segmentation code is in the MCAvatar directory. The MCAvatar/MCAvatar directory is intended as the final C++ source code directory. All of the above directories are under source code control via subversion (SVN) [104]. A lib directory contains all libraries used, in particular OpenCV.

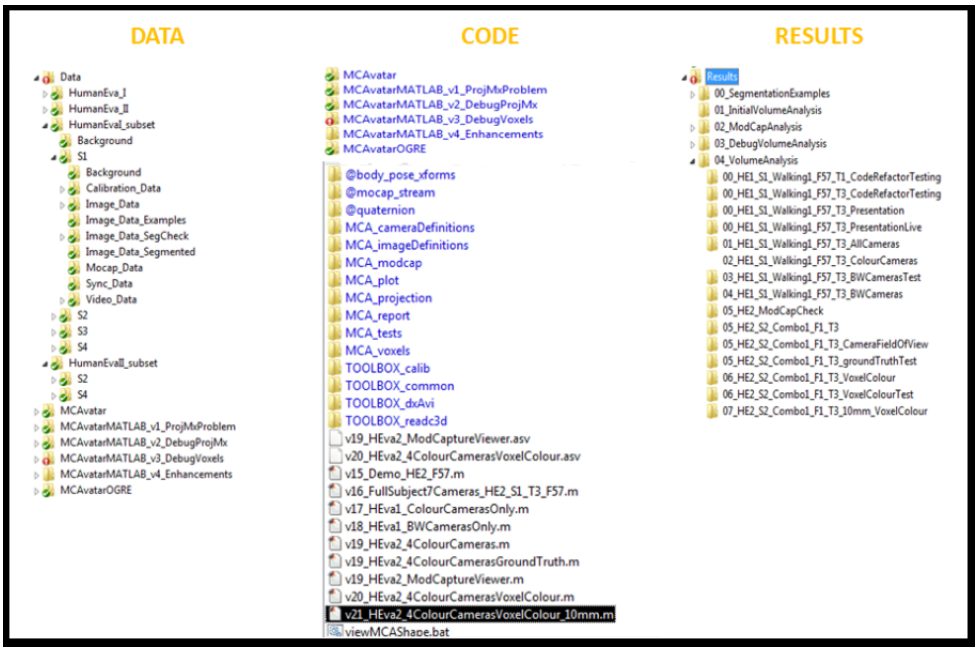

Figure 4.4: *Build directory structure*

4.4 Analysis Overview

The implementation steps were as follows:

— (1) Installation of the HumanEva I and II datasets and code.

— (2) Implementation of the various foreground segmentation algorithms in C++ / OpenCV in order to extract silhouette images of subject 1 walking in the HumanEva 1 dataset.

— (3) Reviewing of the performance of the foreground segmentation algorithms in terms of performance and quality. Creation of manual silhouettes for initial volume analysis.

— (4) Analysis of the HumanEva II dataset to establish the world volume the HumanEva dataset were

videoed in, the co-ordinate system in use and to write the initial 3-D to 2-D projection matrix in MATLAB. Voxel reconstruction of the subject was viewed in two third party voxel viewer.

— (5) Analysis of the HumanEva I dataset following extraction of the video feeds into images. Initial volume reconstruction, debugging of the 3-D to 2-D projection function based on the motion capture data. A MATLAB voxel viewer was written to visualise the subject volumes and the .vxl writer was created.

— (6) Analysis of issues encountered with the HumanEva I 7 camera creation of subject volumes via the shape-from-silhouette technique.

— (7) OGRE 3-D engine installed, compiled and tested.

— (8) Implementation of a fake silhouette generation program in MATLAB based on the motion capture data.

— (9) Implementation of basic voxel colouring.

— (10) Further sampling of the HumanEva data set, creation of subject volumes with voxel sizes of 50mm and 25mm, with various thresholds on the voxel silhouette hit rate. Four primary MATLAB programs were created for volume reconstruction:

  – HumanEva I, 3 colour camera, volume reconstruction
  – HumanEva I, 4 grey scale camera, volume reconstruction
  – HumanEva I, 7 cameras 3 colour camera, 4 grey scale, volume reconstruction
  – HumanEva II, 4 colour camera, volume reconstruction
  – Motion capture fake silhouette generator and image-ModCap sync checker

  The outputs from these programs are described below.

4.5 Primary outputs

The following plots, images and files are automatically generated and saved by the programs in a run directory that is specified by the program being executed.

4.5.1 Plots

— **Plot of occupied voxel centres overlayed on the motion capture data** for the subject. The MATLAB Plot3(X,Y,Z,colour) function is used to generate this data view, .PNG and .FIG file formats are saved. The hit rate (number of silhouette images this voxel projects into) for the voxel are displayed by colour. This plot confirms that the world volume about the subject and the motion capture data for the frame are in sync. This plot will be important for tracking as it will allow provide a ground truth for the tracking algorithm. File name(s): plot_centres.png, plot_centres.fig. See figure 4.6.

— **Plot of occupied voxel centres** for the subject. This plot forms the basis of the MRI like contour skeleton investigation. The contour slices through the subjects volume are taken at half voxel size intervals. The MATLAB Plot3(X,Y,Z,colour) function is used to generate this data view, .PNG and .FIG file formats are saved. File name(s): plot_centresOnly.png, plot_centresOnly.fig.See figure 4.5

— **Voxel plot** creates a 3-D view of the subject in MATLAB. The origin (0,0,0) is also presented as a future position reference point. The voxels cubes are coloured based on the hit rate. The MATLAB patch('Faces',cubeFaces, 'Vertices',voxelPoints, 'FaceVertexCData',faceColour,'FaceColor','flat') function is used to generate this data view, a .FIG format of the plot is saved. The purpose of this plot is to provide a visualisation of the subjects volume that can then be used for comparison with ModCap ground truth data and for the generation of contour slices through the subjects volume as obtained from an MRI scan. File name(s): plot_voxels.fig, plot_voxels.png.See figure 4.7.

— **Per camera voxel view map**. This plot maps the cone of voxels that contribute to the silhouette image of a camera. This plot allow the integrity of the visual hull to be assessed. The problems with the HumanEva I BW1 and BW4 cameras are evident in this view. The view is useful when plotted for the entire volume space e.g. the room rather than for a volume about the subject.The MATLAB Plot3(X,Y,Z,colour) function is used to generate this data view, .FIG file formats are saved. File name(s): plot_cameras.fig, plot_cameras.png.See figure 4.8.

— **A 3-D convex hull** of the voxels was also generated but this proved not to be useful as the arms and leg detail disappears into an egg shaped multi-facaded 3-D mass. This plot represents the initial stages of mesh generation but work in this area has not been completed. The marching cubes algorithm [63] was the intended initial approach.

4.5.2 Images

— **A coverage image for each camera** displays the voxel centres and vertexes that projected back into the images. A yellow coloured pixel represents a voxel centre or vertex that mapped to the image but did not project into the silhouette. Red pixels are centres or vertices that are within the silhouette. The degree of redness (tone) represents the number of times that pixel has been mapped by a 3-D to 2-D projection. The MATLAB code writes directly to the image BUT Y,X co-ordinates are used as MATLAB sees images in this co-ordinate system rather than the X,Y co-ordinate system it uses for its own plotting functions. The purpose of this plot is to assess the coverage area in terms of the image of the volume being projected into 2-D. The plot was useful in identifying issues between the world space co-ordinates used by the 3 colour cameras and 4 grey scale cameras used for recording in HumanEva I. File name(s): coverage_C1.png, coverage_C2.png, coverage_C3.png, coverage_C4.png, coverage_BW1.png, coverage_BW2.png, coverage_BW3.png, coverage_BW4.png. See figure 4.9 and figure 4.10.

— **A coverage image with motion capture data superimposed**. This is the above image with a cylinder representation of the subject as generated from motion capture data superimposed. This image is currently a snap shot of the plot window instead of a get truesize(gcf) with getframe, so it is useful only as for visual inspection of the silhouette
motion capture correspondence. 4.9 and figure 4.10.

— **Fake Silhouette images.** These images are generated for the purposed of performing ground truth analysis on the volume reconstruction methods. The joints points are taken from the motion capture data and cylinders are constructed for the main body parts. The cylinder is projected into 2-D and its convex hull is computed. The fill (x(listOfPoints) , y(listOfPoints), colour.white) function is used to plot the subjects fake silhouette. In this case the truesize(gcf) with getframe functions are used to save the plot image so that the dimensionality of the images is not lost. The generated images are stored in the DATA directory with a subjects subdirectory under the directory, Image_Data_GroundTruth. Their file name depends on the frame number they were generated from. File name(s): frXXXXX_gt.png. See figure 4.12 and figure 4.11

4.5.3 Files

— **Voxel volume reconstruction file** as used by vxlviewer.exe. File name:shape.vxl. The viewer provides a 3-D view of the subjects voxel volume and allows the representation to be rotated around.

See figure 4.13 and figure 4.14.

— **Voxel data reconstruction analysis file**. File name: volume.vra

— **Settings file**, containing the parameters used by the program during its execution. File name: settings.txt

4.5.4 Files formats

Voxel volume reconstruction file

Line 1: Integer Number of voxels (N)
Line 2: Float 9999.99 Voxel size
Line 3: 9999.99 9999.99 9999.99 999 999 999 X, Y, Z, R, G, B
:
Line N+2: 9999.99 9999.99 9999.99 999 999 999

Where X,Y,Z are the co-ordinates of the voxels centre and R,G,B are the red, green and blue colour values within the range 0-255. N equals the number of voxels.

Voxel data reconstruction analysis file
Line 1: ”9999.99” Voxel size
Line 2: ”9999.99”,”9999.99”,”9999.99”,”99”,”999”,”999”,”999”,”99”, ”9999:9999”,...,”9999:9999”
:
Line N+1: ”9999.99”,”9999.99”,”9999.99”,”99”,”999”,”999”,”999”,”99”, ”9999:9999”,...,”9999:9999”

Position 1: X co-ordinates of the voxels centre (float)
Position 2: Y co-ordinates of the voxels centre (float)
Position 3: Z co-ordinates of the voxels centre (float)
Position 4: Voxel hit count (integer)
Position 5: R red colour value (integer - 0-255)
Position 6: G green colour value (integer - 0-255)
Position 7: B blue colour value (integer - 0-255)
Position 8: The camera ID e.g. C1, BW1 etc
Position 9: The pixel position of the voxel centre if in range where if not ”:”

position 10 - X: The pixel positions of any other in range (in image) vertices ”X:Y” (integers)

X equals the no. in range pixels and N equals the number of voxels.

Settings file

The information in the settings file is as follows:

```
CONTROL REPORT

.............................................................
FILES USED
.............................................................

Program:  C:\__MScThesis2008\Code\MCAvatar
                    \MCAvatarMATLAB_v4_Enhancements
                    \MCA_report\Report_HEva1_Colour.m

imagename_C1:  ../Data/HumanEvaI_subset/S1
                    /Image_Data_SegCheck
                    /Walking_1_(C1)/fr00057_seg.png

imagename_C2:  ../Data/HumanEvaI_subset/S1
                    /Image_Data_SegCheck
                    /Walking_1_(C2)/fr00057_seg.png

imagename_C3:  ../Data/HumanEvaI_subset/S1
                    /Image_Data_SegCheck
                    /Walking_1_(C3)/fr00057_seg.png

calibration_C1:  ../Data/HumanEvaI_subset/S1
                    /Calibration_Data/C1.cal
```

```
calibration_C2:  ../Data/HumanEvaI_subset/S1
                    /Calibration_Data/C2.cal

calibration_C3:  ../Data/HumanEvaI_subset/S1
                    /Calibration_Data/C3.cal

ModCap to Image sync_C1:  ../Data/HumanEvaI_subset/S1
                    /Sync_Data/Walking_1_(C1).ofs

ModCap to Image sync_C2:  ../Data/HumanEvaI_subset/S1
                    /Sync_Data/Walking_1_(C2).ofs

ModCap to Image sync_C3:  ../Data/HumanEvaI_subset/S1/
                    Sync_Data/Walking_1_(C3).ofs

static C3D MoCap data:  ../Data/HumanEvaI_subset/S1
                    /Mocap_Data/Static.c3d

subject measurement data:  ../Data/HumanEvaI_subset/S1
                    /Mocap_Data/S1.mp

dynamic C3D MoCap data (the actual motion data):
                ../Data/HumanEvaI_subset/S1
                    /Mocap_Data/Walking_1.c3d
.............................................................
VOLUME
.............................................................
worldPointScalePos_X:  -400.000000
worldPointScaleNeg_X:  -1400.000000
worldPointScalePos_Y:  -100.000000
worldPointScaleNeg_Y:  -800.000000
worldPointScalePos_Z:  2100.000000
worldPointScaleNeg_Z:  -300.000000
```

```
Voxel Size          :  25.000000
Voxel hit threshold      :  3.000000
No. voxels  visited      :  107520.000000
No. subject voxels       :  53658.000000
voxelInThresholdCount:  0
No. subject voxels 1 hit :  53658.000000
No. subject voxels 2 hit :  50179.000000
No. subject voxels 3 hit :  49394.000000
.............................................................
Time taken      :  31383.251784
```

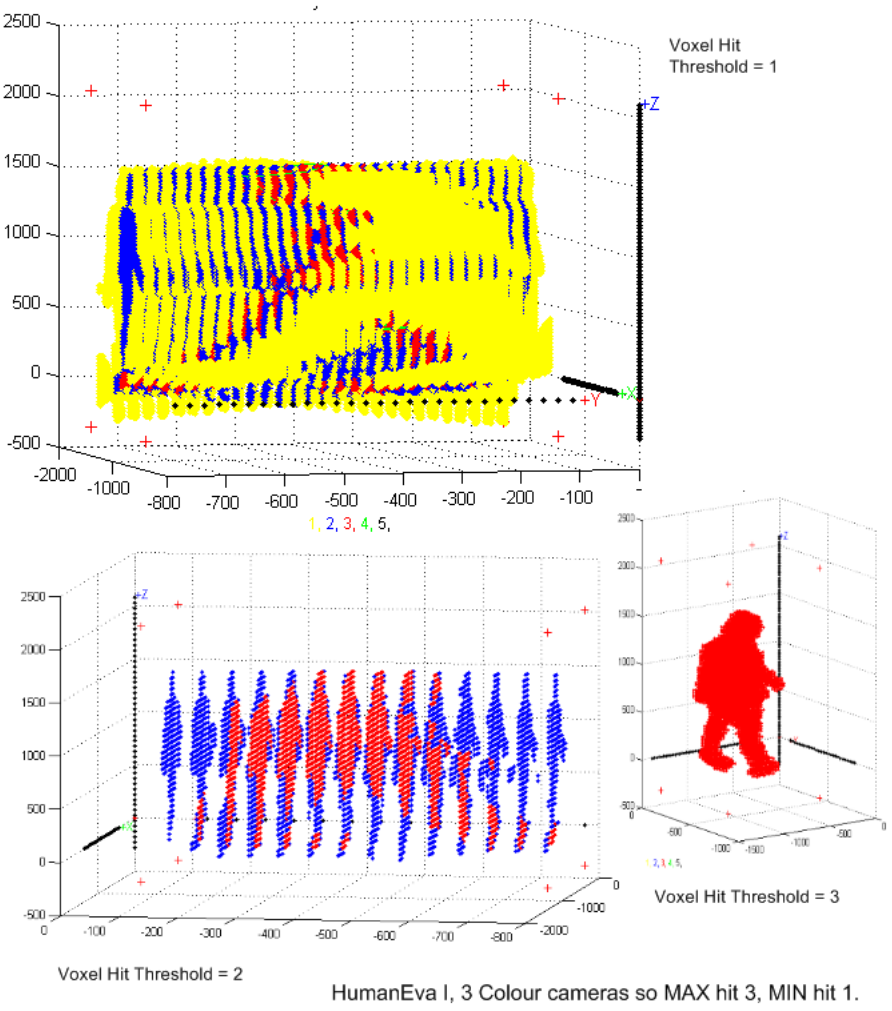

Figure 4.5: *Plot of occupied voxel centres with different hit thresholds. Creating MRI like slices through the data. A hit of 1 means the voxel is only represented within the silhouette of one image. With three cameras the maximum hit is 3, where the voxel project back to the subjects silhouette in all three camera image. With more cameras the core axis (skeleton) of the body should be visible as the voxels with the maximum hit values.*

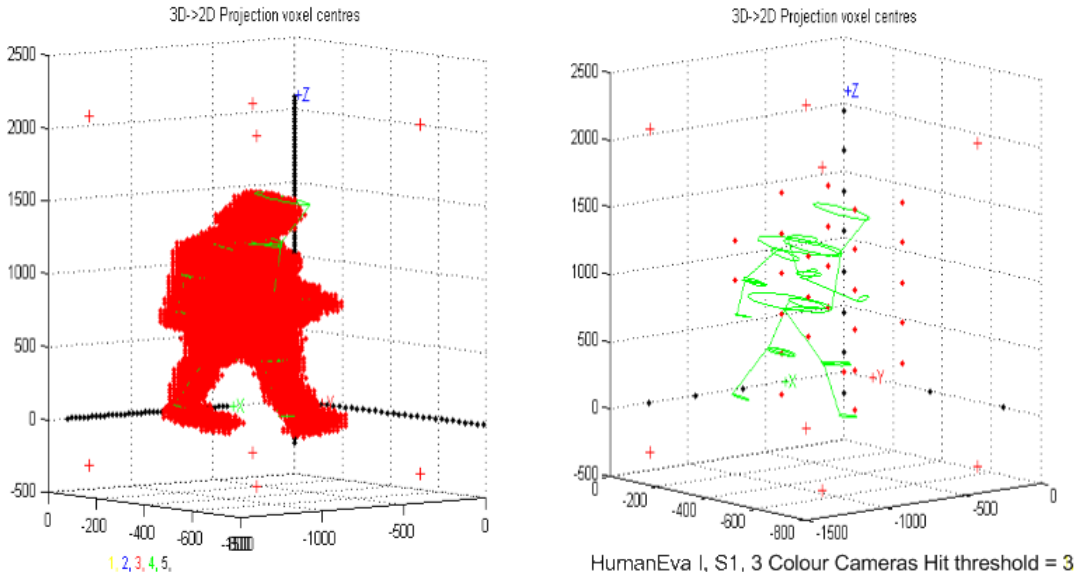

Figure 4.6: *Plot of occupied voxel centres with motion capture data overlay*

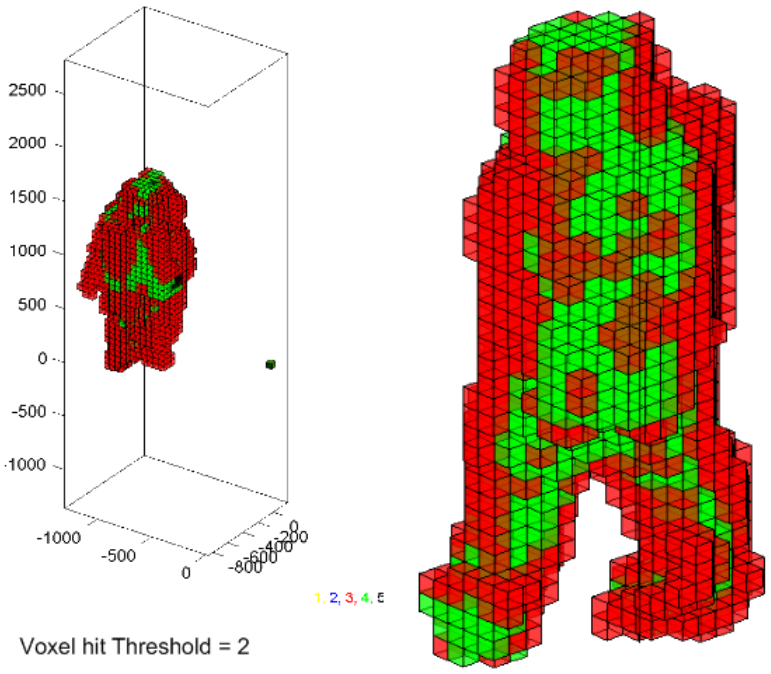

Figure 4.7: *Plot of the individual voxels, that reconstruct the subjects volume*

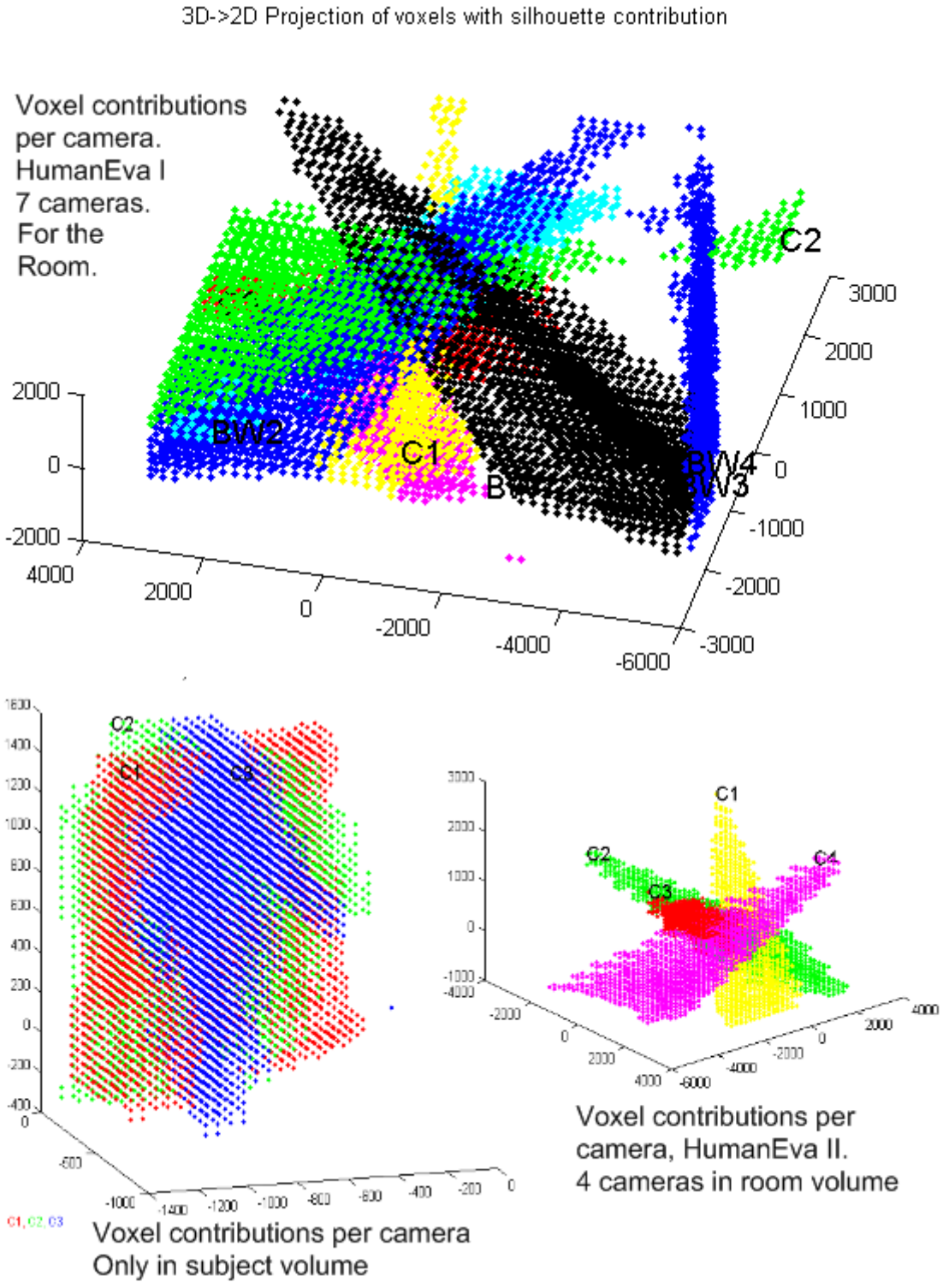

Figure 4.8: *Plot of the cone of voxels that contribute to the silhouette image in each camera. The plots have been generated for the room volume and the subject volume.*

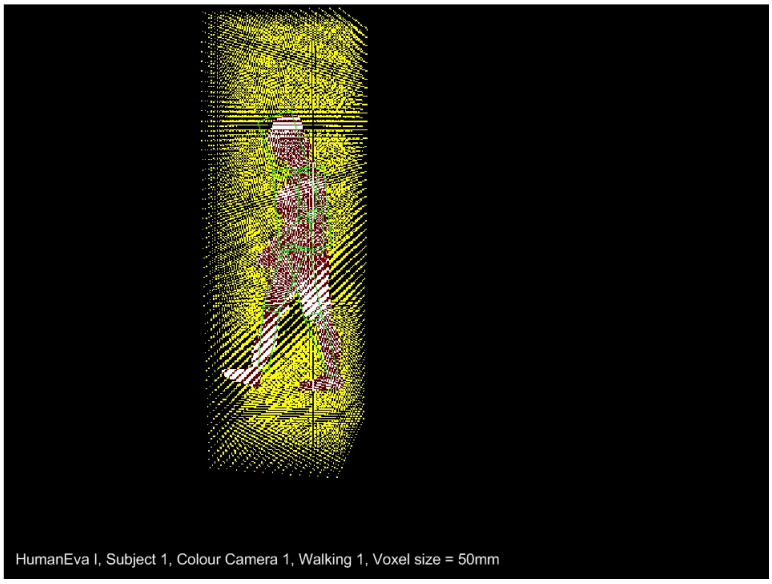

Figure 4.9: *Coverage image with motion capture data overlay, voxel size 50mm*

Figure 4.10: *Coverage image with motion capture data overlay, voxel size 25mm*

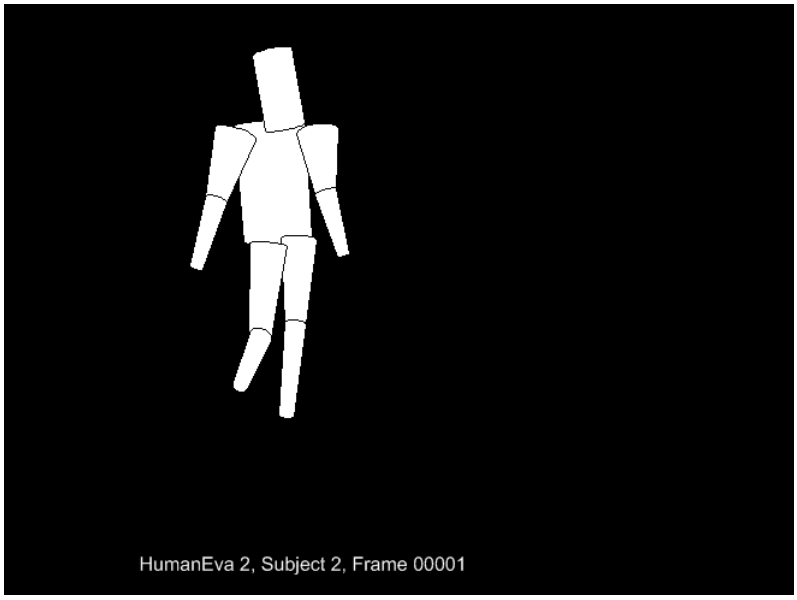

Figure 4.11: *A fake silhouette image*

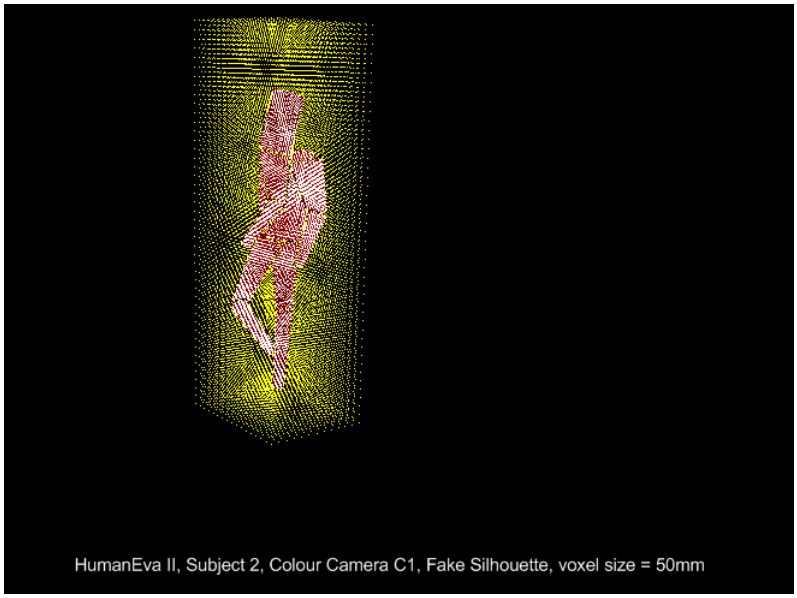

Figure 4.12: *Coverage image with motion capture data overlay, voxel size 50mm as performed with the fake silhouette data*

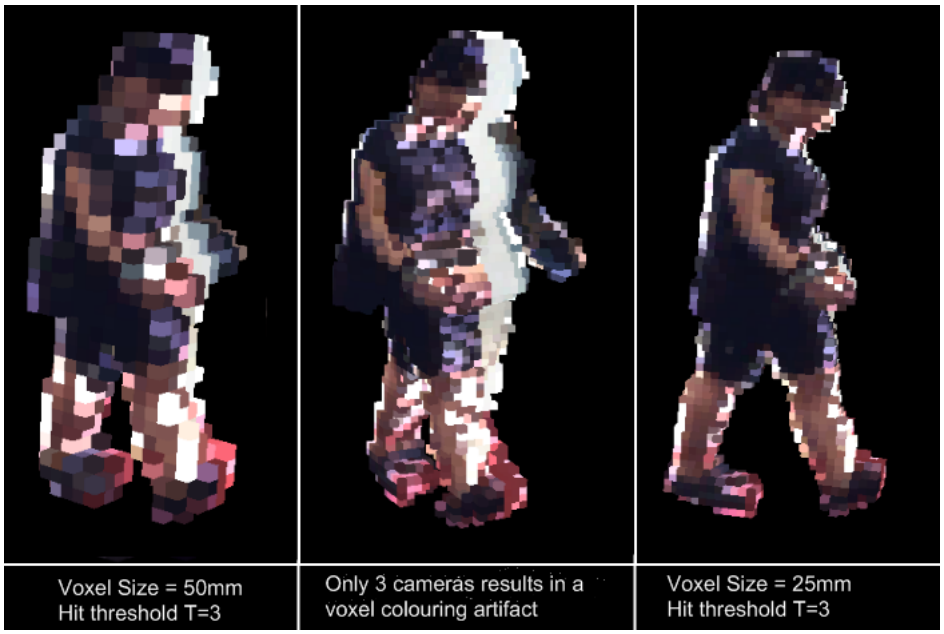

Figure 4.13: *HumanEva I, Subject 1, voxel representation of the subject where basic voxel colouring has been applied to each voxel based on the colour of the pixel of the nearest camera at the voxels centre*

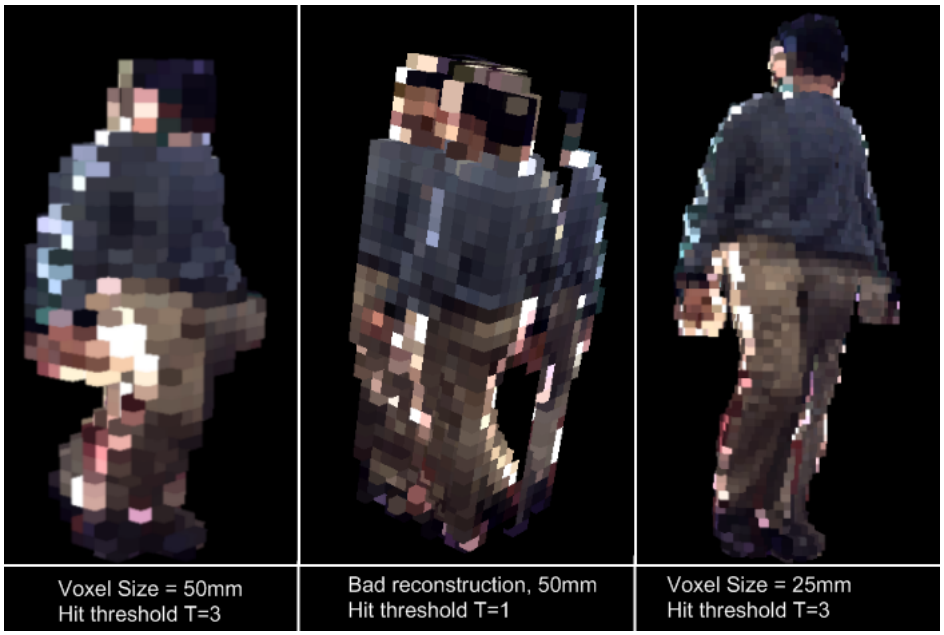

Figure 4.14: *HumanEva II, Subject 2, voxel reconstruction*

Chapter 5

Evaluation of Background Subtraction Methods

There are a number of volumetric scene reconstruction techniques [37]. The approach in this research has been to use the shape-from-silhouette approach as an initial foray into the possible techniques that could be implemented in order to achieve the goals of this research.

Shape from silhouette requires a complete and clean image of the subject of interest to be segmented from the background and rendered in white against an all black background. The cameras must also be calibrated to the same world space and their fields of view must be aligned such that a visual hull [56] of the subject can be obtained. But the first and most important stage of this process is background subtraction.

In considering the possible background subtraction methods that could be implemented the following considerations were taken into account.

— The background subtraction process must be applicable to the constraints of real-time processing as required by a game. The newer consoles typically expect a game to perform at a frame rate of 60 frames per second (FPS) with an image size of 1280 x 720 (high definition video). Therefore it must be possible to obtain silhouette images from a minimum of four cameras at a rate of at least 24 FPS, so every second frame refresh. But the real problem is input lag - the lag between the player moving and the movement being reciprocated within the game. The interval of lag allowed depends on the nature of the game. In jay's [48] study on latency effects for haptic devices it was found that latencies below 25ms have little effect on performance, but the user does not become

aware of the issues until the latencies reach 50ms. So there is potentially 25ms to 50ms available for processing the full pipeline depending on the nature of the game.

— The environment under consideration is a noisy environment as objects other than the player may be moving or objects may be moved intermittently. It is closer to an outdoor environment than an indoor environment, just without the weather and the movement of trees - hopefully.

— Lighting in a family room may cause sharp shadows due to the use of lamps. Light from windows will vary throughout the course of the day. Lights may suddenly be turned on.

— The subject may be indistinguishable from the background, may not be moving at all, mat lie down or may suddenly leave the room.

— The more cameras used the higher the computational load.

See the theoretical basis section 3.7 as regards further details on the methods implemented here.

5.1 Implementation

The goal of the implementation was to provide a C++ / OpenCV background segmentation pipeline that allowed the multiple video streams to be split up into synchronised image files and silhouette image files. This was necessary for the evaluation of the HumanEva I dataset as it consisted of AVI files. The HumanEva II dataset was distributed as PNG files.
The background segmentation implementation focused on the HumanEva I dataset as it contained feeds from 7 cameras which would provide good results as regards the skeleton reconstruction method being investigated.
The C++ / OpenCV code was implemented in Microsoft Visual Studio 2005 Professional edition with the SP1 service pack installed. The source code for the application can be found in the code directory - Code/MCAvatar/MCAvatar. Executables can be found in the Code/MCAvatar/MCAvatar/Debug directory the executable is McAvatar.exe. On execution the following user interface is displayed, see figure 5.1.
The directory Results/00_SegmentationExamples contains snap shot images of the more interesting artifacts of the background segmentation process.
The images and silhouette images generated by these programmes are placed in the Image_Data_Segmented directory under the subjects directory. The Image_Data directory contains the images and the Video_Data directory contains the video files. These directory structures are held within the Code/Data/HumanEvaI_subset

Figure 5.1: *Background subtraction evaluation UI*

and Code/Data/HumanEvaII_subset directories. These directories only contain copies of the HumanEva data set that are being analysed. This is to ensure that the original data is not modified accidentally and to ensure a normative reference exists that can be referenced to if there is any doubt as to the nature of the data or if miscopying is suspected.

The HumanEva I dataset contains three background video files for each subjects session. These video files were recorded at the beginning, middle and end of the recording sessions. The background videos show the scene without the subject present. The HumanEva II dataset contains MATLAB .mat files for the backgrounds.

Note: Background subtraction code is included with the HumanEva dataset. This code was not investigated as it was left for future work once a sense of the datasets background subtraction issues were

assessed via other methods.

5.2 Evaluation

A summary of the methods implemented and their timings is shown below in figure 5.2.
The methods breakdown into those that require an initial background image or set of images (video) to be interrogated prior to commencing silhouette creation (non-adaptive) and those that build up the background model as the capture process commences (adaptive). The preferred approach is be to build up a model of the background and the subject in real-time thus no assumptions about the environment are made. In a family room no assumptions can be made on the state of the room between each play of the game. Though a model of the room could be updated and maintained between plays and with in play much like the running average algorithm but with greater weighting on prior play pixel values e.g. the door or window frame isn't going to go away (though they might get painted!). A statistical model like that would only have to spend computational time rectifying the background model in hot spots. An approach like this would stop the reaffirmation of certain background pixels in a running average type algorithm where the colour of the subjects clothing and the background are similar, see figure 5.3.
It is obvious from viewing the evaluation implementations that each technique has its own advantages but no one of them has the desired effect - fast high quality silhouette generation.

— A basic absolute difference approach has the fastest frame rate of 54 FPS with the fullest subject silhouette but the surround is cluttered with noise and shadows.

— The running average approach runs at 32 FPS but provides an incomplete silhouette with background ghosting. It does however quickly compensate for the difference in the mats position between the empty background image and the current video stream.

— The other implementations - Li, Gaussian Mixed Models (GMM) and basic statistical background - run at 6 / 7 FPS. The GMM implementation provides the best complete silhouette with the only noise being in the form of shadows, though the edges are jagged. See figure 5.4.

— The usage of Li's [62] algorithm was not impressive as compared to the paper on the subject which billed the algorithm as being very good for backgrounds with a high degree of movement. the issue maybe down to parameter tuning as only the default values were used as with the GMM algorithm. The issues we not investigated further as both the performance and quality were not good.

Once shadow removal, morphology and contour finding were added to the implementations the computational overheads increase and still the quality of the silhouette for the faster methods was not a great improvement for the loss in performance.
The following morphology structuring elements were tested to assertion the effect on silhouette quality and process performance. The structured elements in OpenCV are defined by the function cvCreateStructuringElementEx(3, 3, 1, 1,CV_SHAPE_ELLIPSE, NULL) and so the four figure 3311 notation is used to specify the structuring element used during the investigation. The menu option number associated with the function is also included. Images of the results can be found in the directory: Code/Results/00_SegmentationExamples.
See figure 5.5

— 11 - Absolute difference: 8833 - 16 FPS.

— 11 - Absolute difference: 4411 - 32 FPS

— 11 - Absolute difference: 5522 - 54 FPS

See figure 5.6

— 14 - Mean:3311, opening and closing - 12 FPS

— 14 - Mean:3311, opening only - 32 FPS

— 15 - Running Average: 3311, opening, closing, smoothing - 24 FPS

— 15 - Running Average: 3311, opening and closing - 32 FPS

— 15 - Running Average: No morphology - 42 FPS

— 10 - Mean: Shadow removal full image - 6 FPS

— 10 - Mean: No shadow removal - 21 FPS

5.3 Future Work

Whilst these implementations are basic and devoid of optimisation they still provide a qualitative evaluation of the pending issues involved in implementing real-time silhouette generation. The implementations achieved their primary goal which was to educate the researcher on the issues, scope the background

subtraction requirements and provide silhouette images for the next stage of the research - volumetric analysis.

The following background segmentation algorithms need to be implemented and investigated. These approaches are used by the main researchers in the area:

— The HumanEva II dataset contains a background subtraction algorithm [105] based on a mixture of Gaussians. The algorithm creates the background distribution model by learning a mixture density at every pixel by collecting the background images using EM. The subject is assumed to have uniform distribution over all colours. It is noted in the paper that because the background is ambiguous obtaining good segmentation from the grey scale cameras is challenging.

— Cheng and Trivedi [19] (of Mikic fame) perform background subtraction in the HSI colour space followed by connected component analysis that retained only the largest components.

— Cheung (German) [21] describes an algorithm for silhouette generation and shadow removal within RGB space that achieves 30 FPS.

— Theobalt [99] uses a mean and standard deviation approach based on several consecutive video frames with shadow removal based on hue difference between background and foreground pixels, as implemented here. But he further segments the binary silhouette of the person standing in an initialisation position, into segments, using a Generalised Voronoi Diagram (GVD) decomposition.

— Porikli and Tuzel [81] propose the fusing of a number of methods to achieve background subtraction and tracking.

— Friedland [39] uses a colour signature method that is worth investigating.

But it is not just the quality and performance of the algorithms that need to be investigated further. It is obvious from this initial investigation that the use of multiple cues is required in order to achieve a quality silhouette, also the computational overheads need to be optimised by improving the algorithms or the implementation of the algorithms. There is another approach, the deployment of the algorithms across multiple processors or on a GPU.

The gaming computer infrastructure has an edge over general computing. The game consoles - XBOX360 [4] and PS3 [55] contain multiple processors and a GPU (Graphics Processing Unit). This represents significant processing power that could be harnessed to improve the performance of the background subtraction methods. If this is the approach then the techniques that provide the best quality silhouettes should be implemented first, irrespective of their performance implications. Once implemented then research

should turn to optimising these techniques for multi-processor and / or GPU deployment. Changing the technique used, only where it leads to a more efficient deployment within a multi-processor or GPU environment.

A key constraint within gaming is to have processes that execute within a certain bounded time frame. This ensure a consistent game frame rate and maintains player immersion. There are a number of ways that the multi-processor environment of a games console can be used to ensure that the silhouette generation process occurs in a bounded time frame.

— A processor could be dedicated to the image / vision processing components of the game.

— Each camera could perform its processing on a separate processor.

— Where multiple cues are used in the generation of the silhouette e.g. GMM, skin detection, Haar face detection and photo-consistency for instance, then each process could be performed on a separate processor for all cameras. If one process doesn't return in the required period of times the returned cue information from the other processes maybe sufficient.

— Having access to additional processors could allow the image / vision processing components of the game to spawn off additional algorithms to cope with edge conditions without affecting the time bound processing.

Subject initialisation and parameterisation is a key component of many of the state of the art systems, de Aguiar [31] performs a full body laser scan. This allows the processing to be constrained as the dimensions of the subject are known. An initialisation phase could be used not only as an opportunity to obtain the dimensions of the subject but to create a photo-consistent model of the subject thus improving the foreground / background segmentation process. Having a photo-consistent model of the subject would be important for maintaining / obtaining a silhouette of the subject if they are standing still. It is also a good example of a image / video processing edge condition that could take advantage of a multi-processor environment. It might be necessary to introduce a object of known dimension or colour into the initialisation phase e.g. a florescent orange ball, in order to more easily identify the subjects body parts or joints. For instance have the player touch the ball to their nose, hold in left hand, right hand, touch belly button, and each knee. Then the ball is discarded.

An additional approach that requires investigation is to process the camera images at a low level of detail in order to identify the subject general presence in the volume. Then the more computationally intense methods can be applied on a smaller image space e.g. shadow removal, connected component analysis, morphology, contour finding.

This researcher would have a preference for non-model approaches to human motion capture where the requirements for tight clothing or other subject constraints e.g. body symmetry, are not necessary.

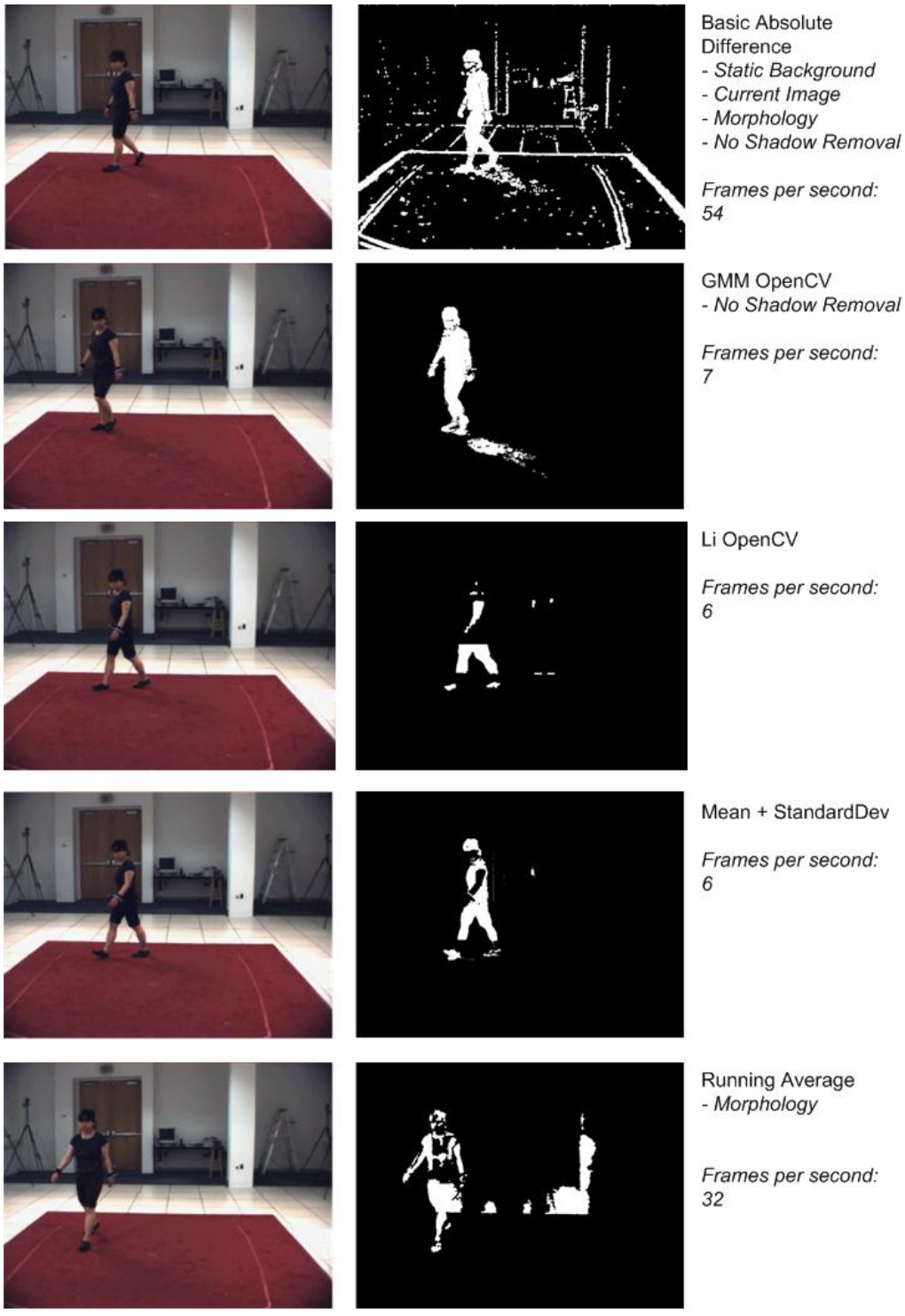

Figure 5.2: *Background subtraction results*

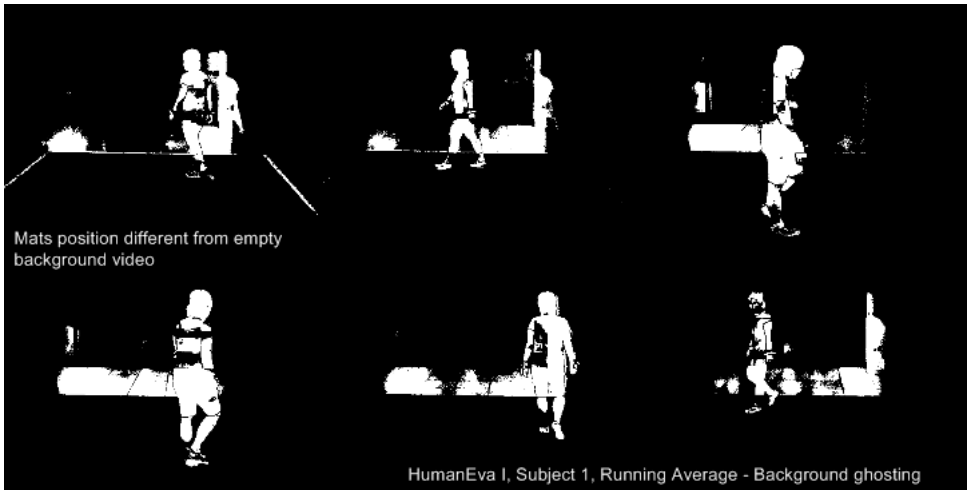

Figure 5.3: *Running Average, background ghosting as the subjects clothes are close in colour to the background colour*

Figure 5.4: *OpenCV's GMM background segmentation results - sharp edges*

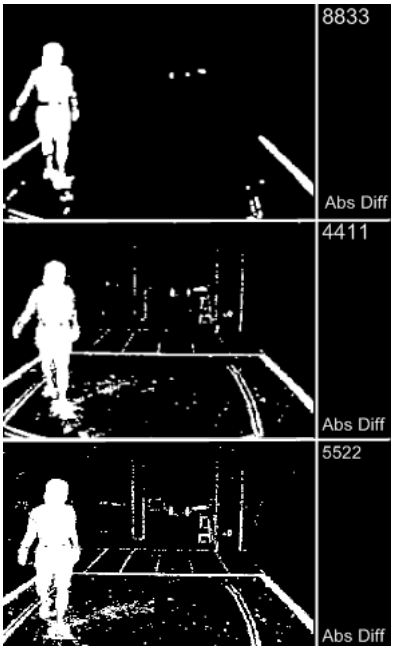

Figure 5.5: *Morphology investigation - Absolute difference*

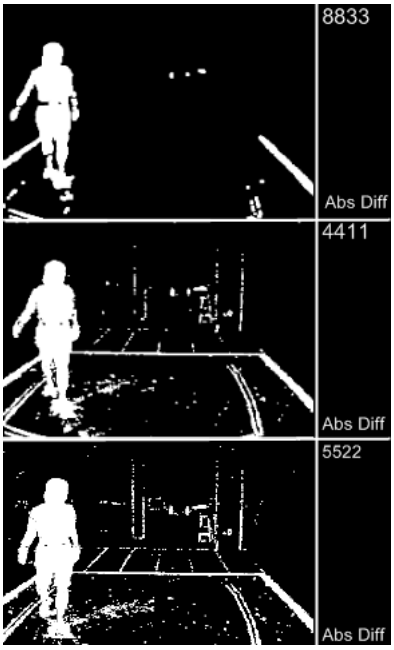

Figure 5.6: *Morphology investigation - Running Average*

Chapter 6

Initial Volume Analysis

The HumanEva II dataset is distributed as a series of PNG images rather than as an AVI video stream. The dataset contains only four colour cameras, as opposed to HumanEva I's seven cameras and the motion capture data has been hardware synchronised with the images.
Initial analysis of the dataset commenced with manually generated silhouettes. The goal was to:

— Establish the co-ordinate system being used e.g. which axis was up.

— Establish the scale of the scene, how big was the area seen by the camera and what were the units of measurement.

— What were the fields of view of the cameras and how did they overlap in the scene volume. Knowing the field of view overlap allows an optimised room volume to be established which defined the volume of interest occupied by the subject. This being the maximum volume search space if the world space co-ordinates (location) of the subject was not known.

Essential to this initial analysis was the implementation of a 3-D to 2-D projection function. The HumanEva demonstration code is written in MATLAB and relies on Bouguet's [13] camera calibration toolkit for its 3-D to 2-D projections. The objective of this research is to develop software in C++ / OpenCV that can be implemented on a games console. Therefore a 3-D to 2-D projection function was written from scratch in MATLAB, see appendix B (section 10.9). Whilst a C++ / OpenCV implementation is desired it was important to model this implementation in MATLAB first in order to compare the results with the ground truth implementation in the HumanEva / Bouguet code. Modelling the code first in MATLAB allowed the following ground truth approaches to be developed:

— The HumanEva / Bouguet 3-D to 2-D projection function could be compared with the projection function implemented within this research for the projection of joint positions and body parts modelled as cylinders.

— Motion capture data could be used to render fake silhouette images of the subject. The subject volume reconstruction could then be compared with a reconstruction based on the fake silhouettes.

— Motion tracking results could be compared with motion capture data.

— It allows qualitative / quantitative tools and tests to be developed that can be ued to verify future code.

All code referenced is relative to the following directory: Code/MCAvatar/ . There are four versions of the code included with this research. As an agile methodology approach was taken each version reflects a sprint. The follow on sprint represents a refractored version of the previous sprint with addition functionality. The sprints were as follows:

1. MCAvatarMATLAB_v_ProjMxProblem - initial exploration code.
2. MCAvatarMATLAB_v2_DebugProjMx - projection matrix testing and debugging.
3. MCAvatarMATLAB_v3_DebugVoxels - volume reconstruction testing and debugging.
4. MCAvatarMATLAB_v4_Enhancements - final code with verified projection matrix and volume reconstruction functions.

The code and results presented in sprint 1 and 2 suffer from an incorrect application of the extrinsic camera matrix. Therefore the getPixel() function which provides the 3-D to 2-D projection was incorrectly projecting into the images, see figure 6.5. This code is referenced in the sections below to illustrate the analysis process and approach rather than the correctness of the code. Many of the outputs were initially conceived during the initial sprints as the results obtained with this code did not seem correct given the theoretical basis of the research. It was only through the use of these tools and ground truth data that the issues were identified, corrected and the current implementation verified. The verified getPixel() function can be found at MCAvatarMATLAB_v4_Enhancements/MCA_projection/GetPixels.m where the final implementation takes the the XYZ co-ordinates of the voxel centre and vertices. It can return the pixel location for the vertices, the vertices and centre or all pixel co-ordinates within a bounding box of the convex hull projection of the voxel - its image footprint.

6.1 Implementation

The co-ordinate system and the camera view results can be seen in figure 6.1.
Code reference: MCAvatarMATLAB_v1_ProjMxProblem/MCAvatar_v02_ComputeBestXYZ.m
Results reference: Results/01_InitialVolumeAnalysis

To obtain these results an initial world volume size was chosen and the voxel centres were project onto the images in each camera with the MATLAB plot function to establish the amount of the images in each camera represented by the world volume. See figure 6.2.
Next MATLAB plots were rendered on to the images where the 3-D to 2-D projection mapped to the silhouette of the subject, figure 6.3. The purpose of this test was to ensure that the CheckImage() function was implemended correctly. The CheckImage() function tests if the pixels returned from the getPixel() function were in the image, and, if in the image, are they within the subjects silhouette.
The final version of the CheckImage() function has a number of variations that return:

— only the pixels in the silhouette

— the pixels in the silhouette and any that were in the image space

— the pixels in the silhouette, those in the image space and the pixel colours on the image

Code reference:MCAvatarMATLAB_v4_Enhancements/MCA_projection/
CheckImageGetValidAndInvalidPlusColour.m

The results returned from the checkImage() function highlighted an issue where the XY co-ordinates returned by the GetPixel() function needed to be changed to YX co-ordinates in order to lineup the silhouette image and the silhouette as rendered by voxel projections. By projecting a rectangle into a MATLAB plot and writing the same rectangle onto an image it was established that MATLAB plots XY but in checking pixel co-ordinates in an images or writing to an image uses YX. The checkImage() function takes accounts of this.
Given that the actual images are the normative reference and MATLAB is just a visualisation tool. All further coverage analysis was performed by writing to copies of the actual images rather than using MATLAB plots.
The final stages of this sprint created a visualisation of the the voxels that were contributing to the silhouette image in each camera. If the volume chosen is the entire room then a plot is obtained of the cone

of voxels contributing to each camera. This is useful in determining the overlap in the cameras and thus provides a means of checking the quality of the visual hull. See figure 4.8.

6.2 Evaluation

The purpose of this stage of the analysis was to:

— Establish the co-ordinate system - positive Z is up, X and Y are the floor plane. The x on the floor mat is the 0,0,0 location of the world co-ordinates.

— Establish the volume of the room - in the X direction the room is approximately -3 metres to +3 metres, in the Y direction it is approximately -6 metres to +6 metres. See figure 6.6.

— To validate the 3-D to 2-D projection function.

— To develop tests and tools for analysing the data further.

— To compare HumanEva motion capture projection with the developed functions

All of the above goals were achieved and validated based on information provided on the HumanEva website in the FAQ section.
The code related to this analysis can be found in the following directories:

1. MCAvatarMATLAB_v_ProjMxProblem - initial exploration code.
2. MCAvatarMATLAB_v2_DebugProjMx - projection matrix testing and debugging.
3. MCAvatarMATLAB_v3_DebugVoxels - volume reconstruction testing and debugging.

With corresponding directories in the results directory.

6.3 Future Work

The analysis code produced at this stage was not carried through into the sprint 4 refractor. Therefore care should be taken in reviewing it as the projection matrix is incorrect. The following code should be rewritten within the sprint 4 framework:

— XYZ co-ordinate view

— Camera fields of view, for obtaining the optimum world volume

— Voxel to camera plot, with a voxel centre offset per camera so the plots don't overwrite each other.

As they are useful tools in visualising the environment and debugging the implementation code. These tools should also be incorporated in the C++ / OpenCV implementation.

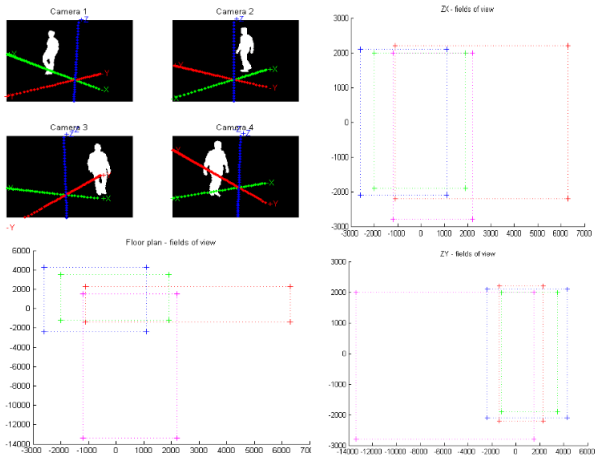

Figure 6.1: *Establishment of the world space co-ordinate system and camera fields of view. This diagrams data is incorrect as it was generated with a defective getPixel() function but it serves to illustrates the approach taken.*

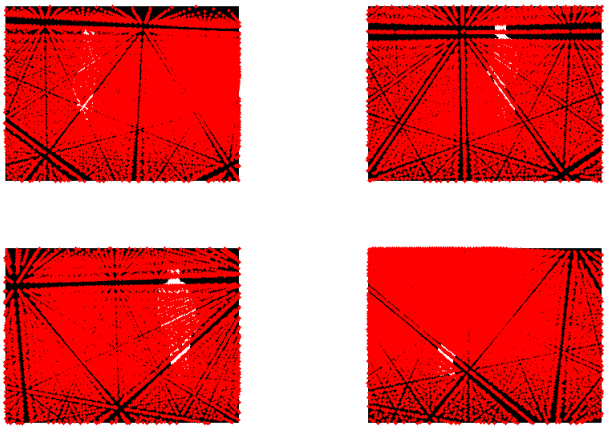

Figure 6.2: *Establishing the world volume. The red dots represent a voxel centre projected to the image. Various XYZ ranges were input and the images generated until the image was completely covered with voxel centre projections.*

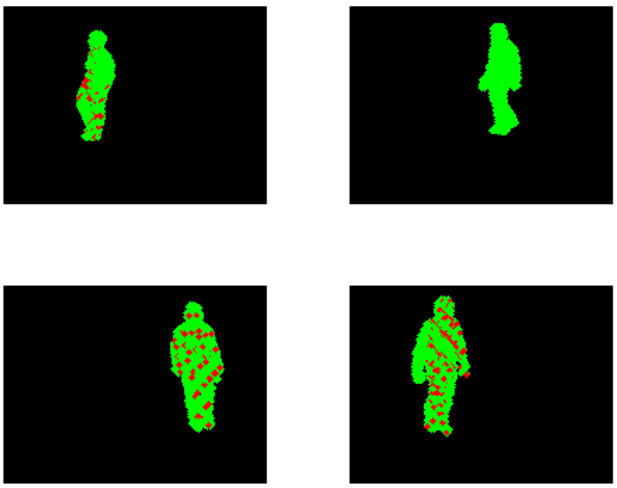

Figure 6.3: *Checking the CheckImage() function. Only voxel that project into the silhouette of the subject are recorded.*

Figure 6.4: *Coverage report presented on the actual image rather than plotted in MATLAB.*

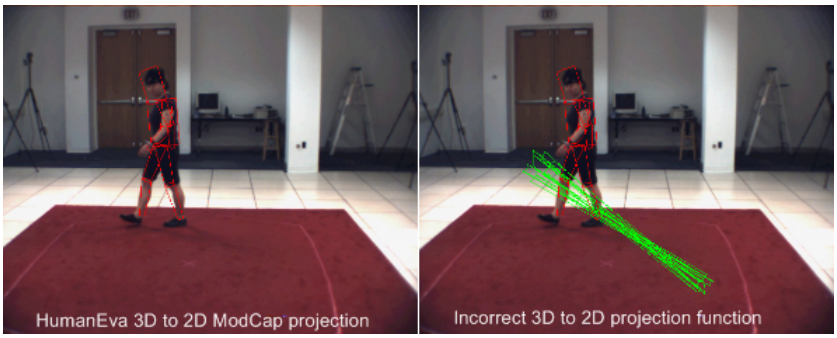

Figure 6.5: *Comparison of HumanEva projection of the motion capture data and the incorrect projection function, prior to the incorrect application of the extrinsic matrix being corrected.*

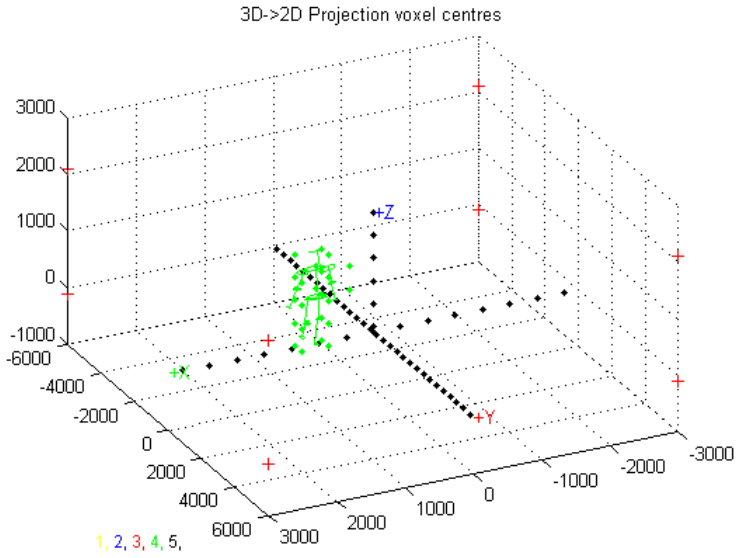

Figure 6.6: *Subject plotted in the volume based on the motion capture data.*

Chapter 7

Volume Rendering Code Overview

7.1 General Structure

The volume rendering code is comprised of five basic stages:

1. **Initialisation** - acquisition of the images, camera calibration files, motion capture data, volume to be rendered, voxel size and outputs to be generated.
2. **Loop volume** - iterate through the volume, projecting each voxel via the getPixels() function into 2-D image space.
3. **If voxel occupied** - check if the pixel co-ordinates returned by the projection are in the image and if so are they within the silhouette, this is the responsibility of the checkImage() function. Gather up data for the analysis stage.
4. **Analysis** - Plot the results of the volume traversal and occupancy tests.
5. **Output** - output the .vxl and .vra files which contain details on the voxels co-ordinates and colour.
6. **Controls** - Generate coverage images with motion capture projections.
7. **Display** - Display the analysis graphs, control images and call vxlviewer.exe to display the volume reconstruction of the subject.

7.2 Implementation

The current implementation has essentially 4 versions of the above program:

— HumanEva I, 3 colour camera, volume reconstruction

— HumanEva I, 4 grey scale camera, volume reconstruction

— HumanEva I, 7 cameras 3 colour camera, 4 grey scale, volume reconstruction

— HumanEva II, 4 colour camera, volume reconstruction

To render a volume for a particular subject, action and frame the parameter section of the code at the top of the file needs to be modified. It was intended that the next sprint would convert these functions to C++ / OpenCV and so they were not implemented as MATLAB functions.

It is for this reason that you will see copies of the code for various analysis runs within the MCAvatar-MATLAB_v4_Enhancements directory. Each copy has different parameters at the top of the file. This process ensures that the results can be traced back to the code that generated it so that anomalies found in other runs can be crosschecked with previous code rather than having to revert the source repository.

As the version number of the code is incremented so there may be changes to the code or it may be an analysis run with new parameters. The code is basically divided into a main execution trunk with segments of code in their own include files e.g. MCA_Plot/PlotHitsOnImages.m so that they can be easily commented out or if'ed out during analysis runs. Other segments of functionality are implemented as functions particularly in cases where they are involved in a loop and thus it is important that variables are descoped.

As the code is written, run and evaluated within a sprint changes to include files or function are versioned within their name to ensure previous code developed during that sprint still functions.

Moving from one sprint to another the code is refactored and the include file / function file versioning removed.

All code, data and results are also maintained within a subversion source repository.

The include and function files are contained within the following directories (packages)

— MCA_cameraDefinitions - camera related code

— MCA_imageDefinitions - definitions of the images to be used for an analysis run

— MCA_modcap - motion capture related functions

— MCA_projection - getPixels() and checkImages() functions

— MCA_report - reports generation code

— MCA_tests - debug includes, functions and standalone programs

— MCA_voxels - voxel visualisation and output file generation

Extractions of the HumanEva demo code are also included. These extractions are NOT used for volume rendering or by any portion of the prototype code developed here. They are used to compare results and provide ground truth data between the HumanEva motion capture data and the code developed here.

7.3 Evaluation

The code is fit for purpose as defined within agile methodologies. The MATLAB implementation is only intended as a proof of concept implementation prior to implementation of the code in C++ / OpenCV. The code has been structured and versioned in a manner that aids data analysis and function debugging NOT the deployment of standalone code, which is in keeping with the goals of the research, the current sprint and the time constraints. Output traceability was the primary concern in the design of the architecture and process under taken in the development of the code.

7.4 Future Work

The next step as regards the evolution of the code base is to convert it to C++ / OpenCV. Once this has been accomplished and verified with the test data / tools created during the MATLAB prototyping sprint the following additional functionality needs to be added:

— A lookup table for 3-D to 2-D projection which will reduce the computational overhead of the volume loop. Basically the voxel to pixel projections are stored in a lookup table for each camera rather than being calculated each time. This idea is presented in the research of [66] and a variation of it is presented by Cheung (German) [21].

— The voxel colouring algorithm is extremely simple. It takes the pixel colour projected to by the voxel's centre in the image of the closest camera. Where the closest camera is determined by what quadrant (side) of the person the voxel is on and the subjects position in the room, where the centre of the person is the horizontal axis, and the cameras own quadrants of the room. With four cameras the approach is simplistic but acceptable with less or more cameras there are unacceptable artifacts.

The data coverage of the work to data has been extremely small with only a few subjects, actions and frames investigated within each HumanEva dataset. Greater data coverage is necessary before the implementations can be considered to be robust. The most pragmatic approach would be to generalise the

functions to take subject, action, motion capture and camera parameters. Then create a batch processing pipeline to evaluate the full dataset and uncover the potential edge conditions and states.
The implementation of a batch processing pipeline would require the following work:

— Implement full pipeline, segmentation to volume rendering, in C++ / OpenCV.

— Improved silhouette creation by implementing multiple cues.

— Centroid / Blob detection to centre volume reconstruction over subject silhouette.

— Restrict shadow processing, morphology or connected component analysis to set regions based on blob detection.

— re-implement the analysis, output and control functionality to suit a batch processing regime.

With the batch processing pipeline in place newer algorithms / optimisation can be implemented and evaluated providing quantitative data on the best techniques to used for this problem domain. This platform would also provide a means of assessing possible multi-processor / GPU implementations.
The final goal is to achieve real-time generation of the subject volume within a game engine. This means implementing the batch processing pipe line as a real-time process with camera feeds. The critical next step is mesh generation. If the subjects volume reconstruction can be rendered as a textured mesh then this can be used to create collision detection based games. In this case the actual skeleton data is not required as rigging data for a model. Instead the bounding volume of the mesh is used to interact with the virtual environment and the objects in it. This is essentially the approach of non-model based implementations such as de Aguiar et la [31] and Vlasic et la [116] where skeleton and pose estimation are secondary cues to mesh creation, texture and photo-consistency.
The approach of reconstructing the volume of a subject as a mesh avatar rather than animating a model based on skeleton calculations would allow a game avatar to be implemented without the need for the actual skeleton to be inferred or body parts to be tracked, as suggested in the abstract of this research. But the approach is only valid for collision detection based games where the player is conceptualising the meaning of the structure of the mesh. For games that require the comparison of pose information between the player and a database of poses e.g. swing dancing [119] / karate, or animation of a model avatar this approach will not work.
A possible game scenario would be one where the player stands in the centre of the scene. Objects are thrown at the player from all directions and they must dodge in 3-D. If the object hits the player then a voxel or group of voxels are removed from the players mesh. The player looses when no part of them

remains, the last piece could be any part of their body e.g. a cube with an eye. Basically a 3-D asteroids game.

7.5 A Graphical Overview of the Volume Rendering Code

See figure 7.1 for a graphical overview of the MATLAB code used to render a 3-D volume of a subject from several 2-D images.

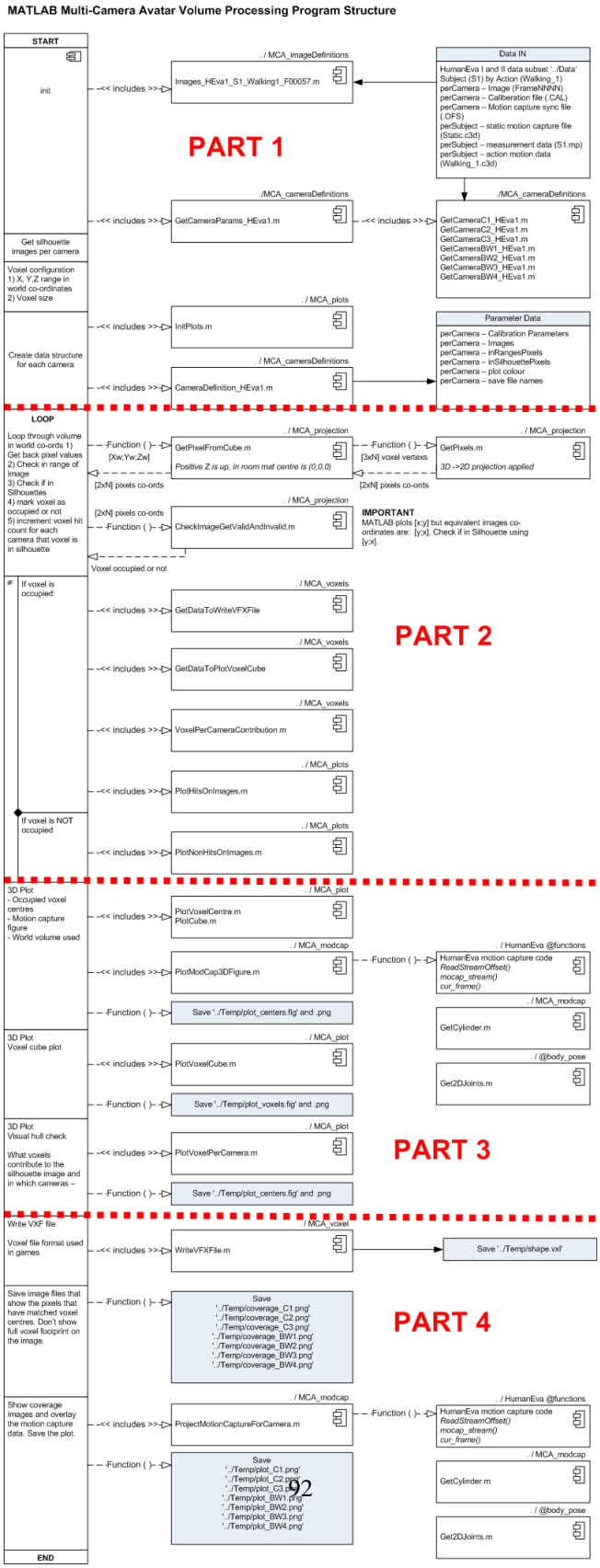

Figure 7.1: *Code overview*

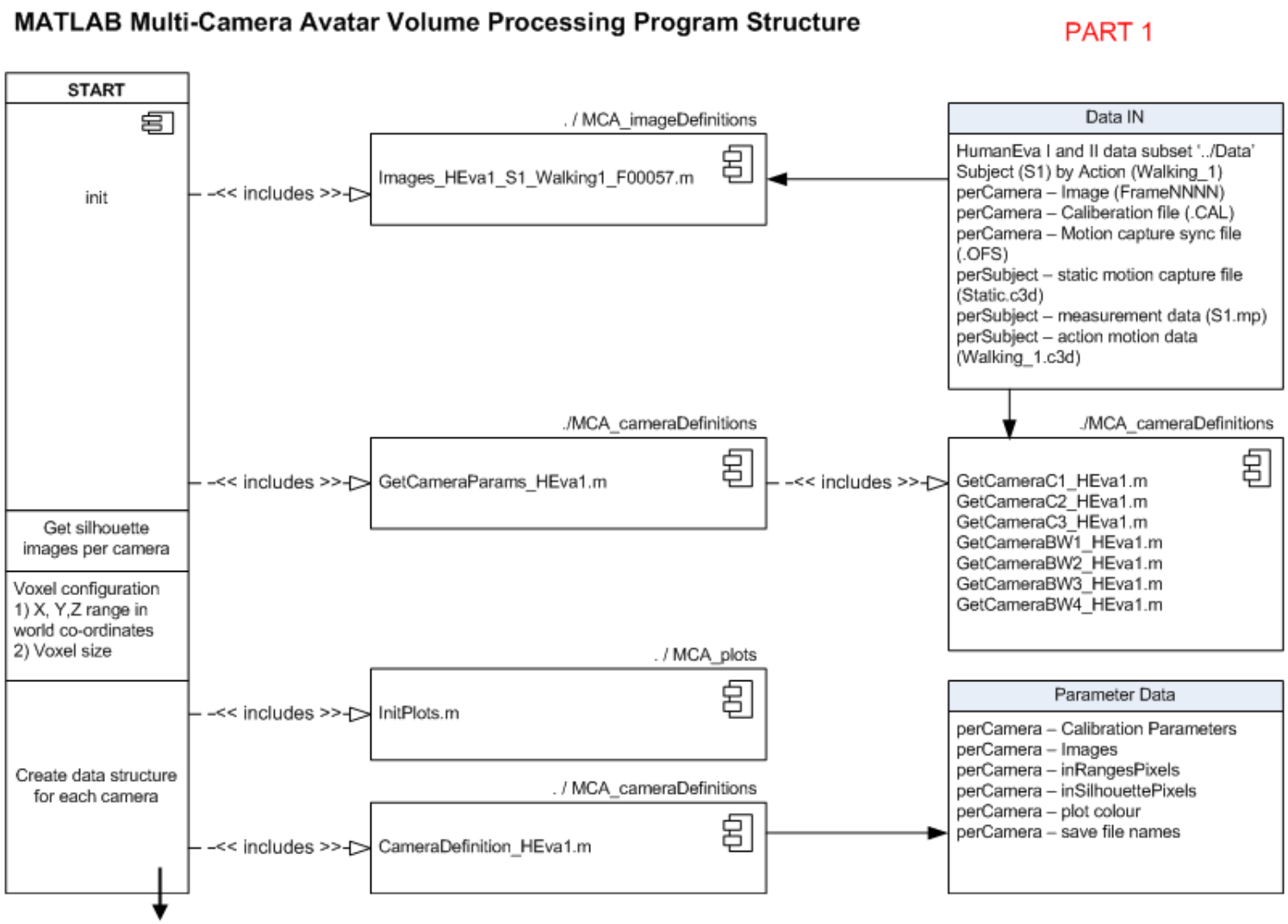

Figure 7.2: *Code overview: Initialisation - Part 1*

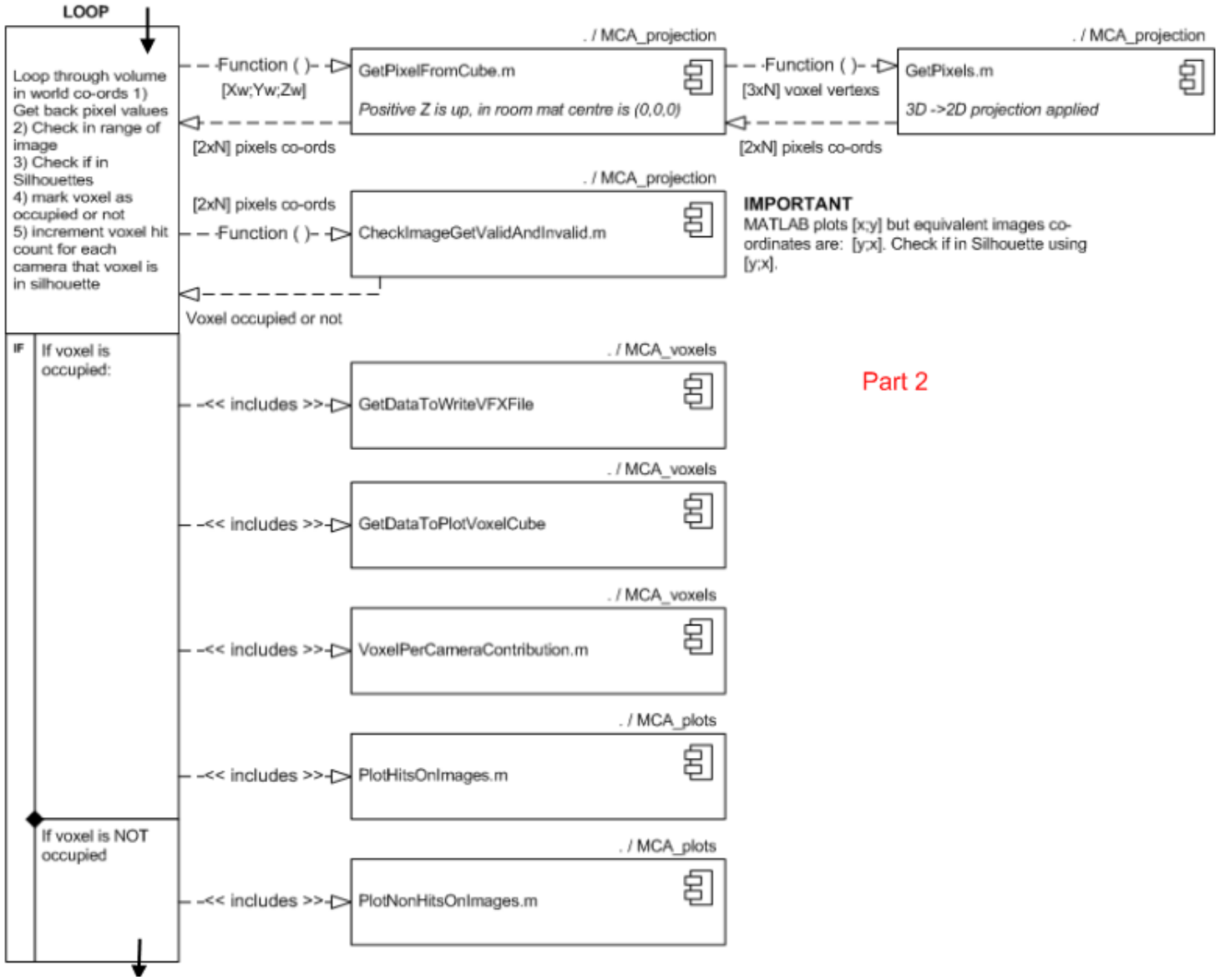

Figure 7.3: *Code overview: Loop - Part 2*

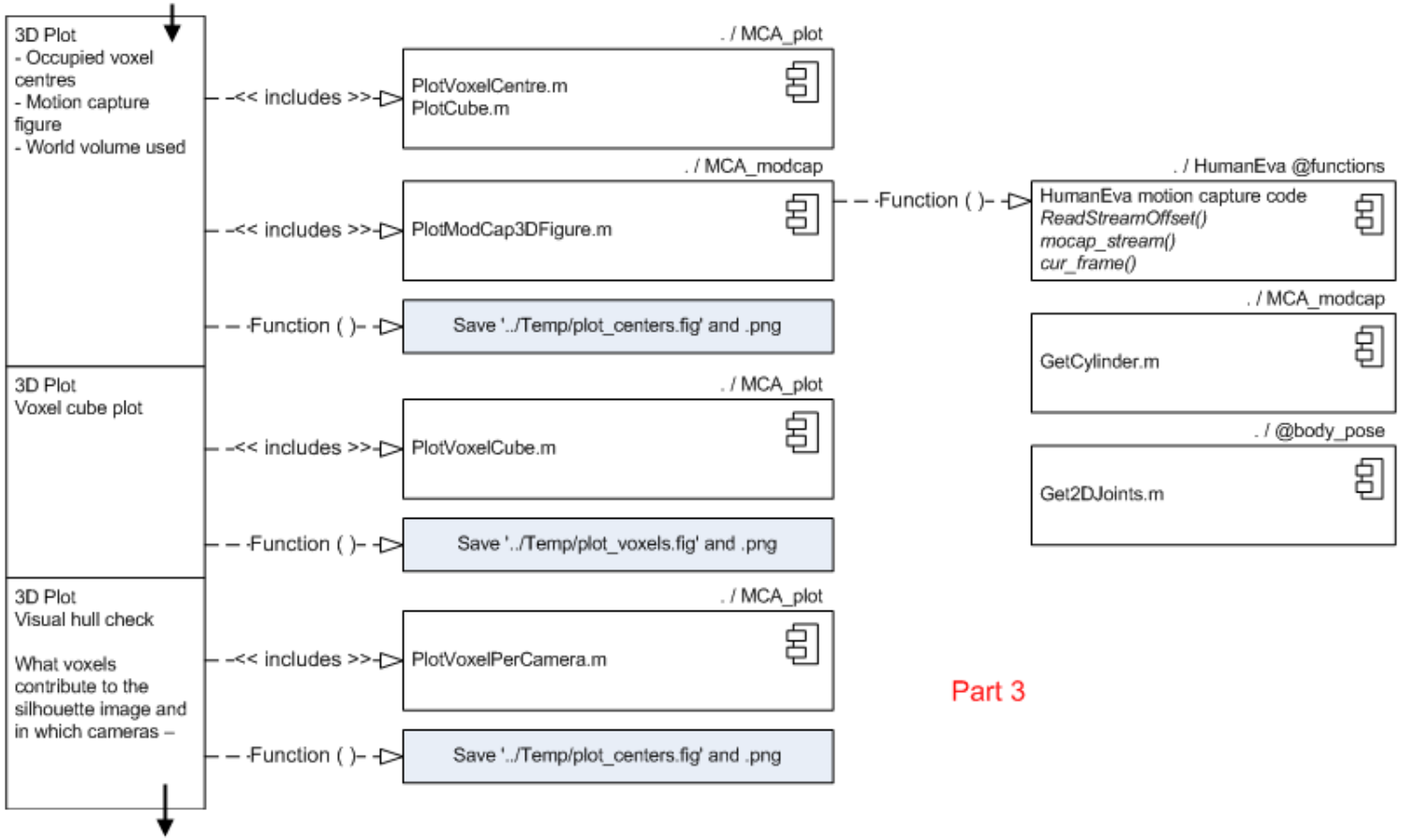

Figure 7.4: *Code overview: Analysis - Part 3*

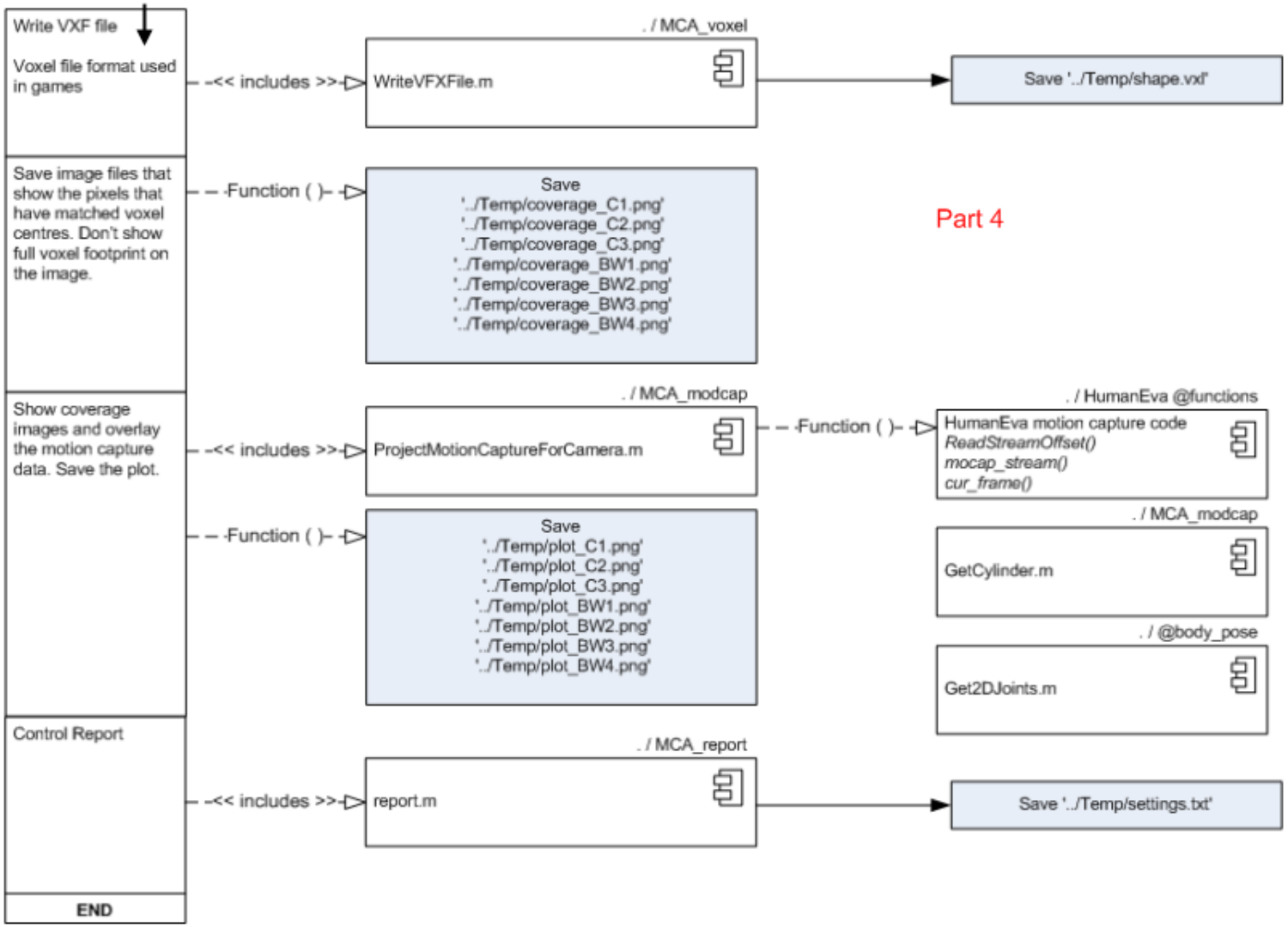

Figure 7.5: *Code overview: Control - Part 4*

Chapter 8

Volume Analysis : HumanEva I

The HumanEva-I dataset [105] contains 7 calibrated video sequences (4 grey scale and 3 colour) that have been software synchronised with 3D body poses obtained from a motion capture system. There are 4 subjects performing 6 common actions e.g. walking, jogging, gesturing.

Figure 8.1: *The HumanEva I Subjects - S1, S2, S3, S4*

8.1 Implementation

8.1.1 First steps - motion capture synchronisation with images

Volume analysis of the HumanEva I dataset commenced with the creation of manual silhouettes for Subject 1, Walking 1, Frame 57 as it was the first frame found with valid motion capture data that matched images in all cameras. Certain frames do not have valid motion capture data as marker tracking may have been lost during the recording session. Functions surrounding the motion capture data retrieval return the valid or invalid boolean based on the frame selected.
The process of syncing the motion capture to the images is currently manual. The portions of the HumanEva demo code that relate to motion capture have been extracted and MATLAB code written around them to provide a tool for syncing the motion capture to the images. The tools can be found in Code/MCAvatar/MCAvatarMATLAB_v4_Enhancements directory. The following motion capture views exist in the code base:

— v19_HEva1_ModCaptureViewer_S1.m - Subject 1 walking

— v19_HEva1_ModCaptureViewer_S3.m - Subject 3 boxing

— v19_HEva1_ModCaptureViewer_S3ThrowBall_Error.m - same code as above but with a runtime error produced from within the HumanEva code.

The variable FRAME is assigned the value (an integer) of the motion capture data frame to extract. There is an offset file per camera, .OFS, that can be found in the Sync_Data directory for each subject. The code reads as follows:
% read the OFS file
"["im_st, mc_st, mc_sc"]" = ReadStreamOffset(*OFSFileName*);

% compute the common offset and scaling for all the streams
common_time = mc_st;
common_scaling = mc_sc;
start_image_offset = im_st;

% Walking 6 to 1203
FRAME = 41;

The FRAME variable and the image frame (fr00000) numbers as extracted with the C++ / OpenCV program discussed in section 5 do not correlate. For instance - subject 1, walking 1, image frame fr00057, matches motion capture FRAME=6. The volume reconstruction agrees with this as the subjects motion capture data plotted in 3-D space agrees with its position in the image of all cameras. See figure 8.2.

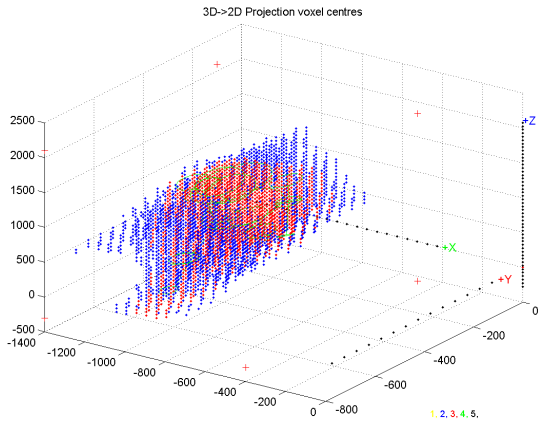

Figure 8.2: *A 3-D plot of the motion capture data with occupied voxel centres overlayed . This supports the correlation of the motion capture and image data. Though there is the possibility that the subject is basically in the same spot every N number of frames since they are walking around in a circle.*

The plot of the motion capture data is very useful in calculating the volume occupied by the subject. The X,Y co-ordinates can be used in defining the world space to be used by the volume reconstruction process. To do this, the 3-D motion capture plot, see figure 8.3, is rotated until the view is from above. Then the area occupied by the subject can then be inferred. see figure 8.4.

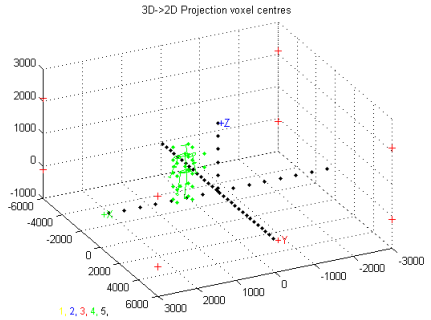

Figure 8.3: *A 3-D plot of the motion capture data. This plot allows the XY volume surrounding the subject to be calculated for volume reconstruction. The Z range used is -100mm to 2100mm.*

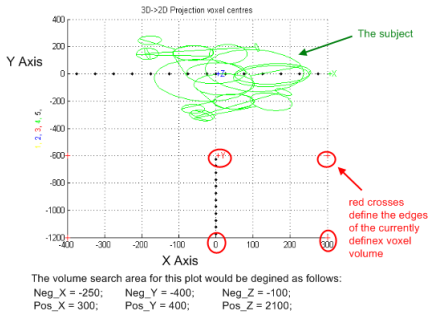

Figure 8.4: *Top view of the motion capture data plot. Showing the volume reconstruction parameters that would be used for the volume occupied by the subject.*

In setting up and using the motion capture data the following files in the Mocap_Data directory are required:

— *Sx*.mp file e.g. S1.mp is the file for the subject S1.

— static.c3d for that particular subject

— *action*.c3d e.g. ThrowCatch_1.c3d the actual action motion capture data for that subject.

In the MCA_modcap directory of the code is the code for creating the cylinders around the body parts, projecting them to 2-D (ProjectMotionCaptureForCamera.m) or projecting them to 2-D to create fake silhouettes (ProjectMotionCaptureForImage.m).

8.1.2 Volume Reconstruction with 7 Cameras

The analysis of subject 1, Frame 57 in the action sequence walking 1 highlighted a number of issues:

— The BW1 & BW4 cameras created split cones when the voxels contributing to their images were plotted.See figure 8.9. Mundermannhad [71] also a reported a problem with these cameras but did not give a reason.

— A test to check the position of the subject in all seven images found that the three colour images agreed with the slice of volume chosen whilst the grey scale cameras did not.See figure 8.7.

— A coverage test of all seven images with a large world volume (X=-3000:3000, Y=-6000:6000, Z=-300:2000, the units are millimetres), rather than a column through the subject, showed that the pixels returned in the grey scale cameras were off centre. Particularly in BW1 and BW2, which

showed the edge of the volume when it should have been cantered on the image as a whole.See figure 8.6.

— Plotting the motion capture data for all seven cameras showed that the 3 colour cameras agreed on the location of the figure. The 4 grey scale cameras also agreed on a location but it was a different location from the colour cameras and colour camera motion capture data.See figure 8.5.

— The issues were investigated further by rendering the joint positions rather than the cylinders. The world co-ordinates for the marker on the top of the subjects head was extracted from the colour camera one's motion capture data - head proximal $\tilde{(}$-908,-444,1271). This world point was plotted into all images with both the HumanEva 2-D to 3-D projection matrix and the one developed during this research. On the colour images the point was at the top of the head but not so for the grey scale images. Checking the motion capture data for the grey scale cameras for the head proximal point returned world co-ordinates of $\tilde{(}$27,-943,1260). Plotting this point into all seven images resulted in the correct location of the point on top of the head in the grey scale images but an incorrect location in the colour images. Results:Code/Results/ 03_DebugVolumeAnalysis/ S1_HEva1_HeadPointsDebug_F57_v12. So the motion capture data maps correctly to the images as the world co-ordinates being projected for the colour cameras in different than for the grey scale images. See figure 8.8.

— There was a critical up date to the HumanEva I OSF (sync files) that was released with HumanEva II. Though this had been applied, it was reapplied. The results were the same.

These issues may explain the decision to release the HumanEva II dataset with hardware rather than software synchronisation as the motion capture and image streams synchronisation method.
The code related to these investigations can be found in the Code/MCAvatat/MCAvatarMATLAB_v3_DebugVoxels directory and the results in the Code/Results/03_DebugVolumeAnalysis directory.

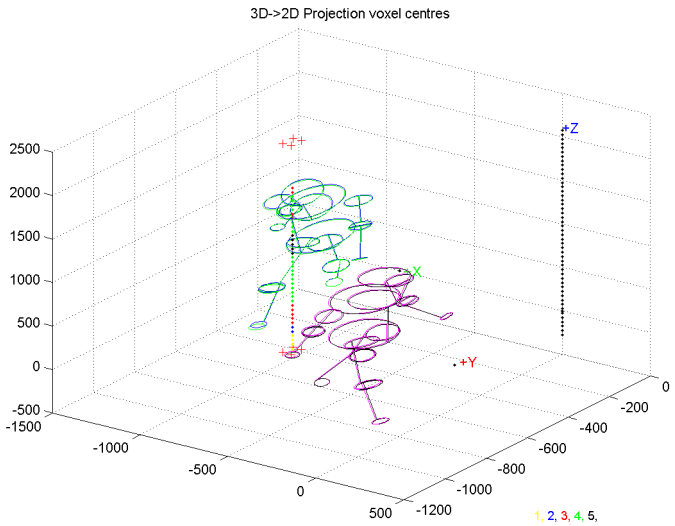

Figure 8.5: *HumanEva I, 7 cameras, S1, Walking 1, Frame 57. The motion data for the frame were plotted showing that the colour cameras had different world co-ordinates for the subject as compared to the grey scale cameras.*

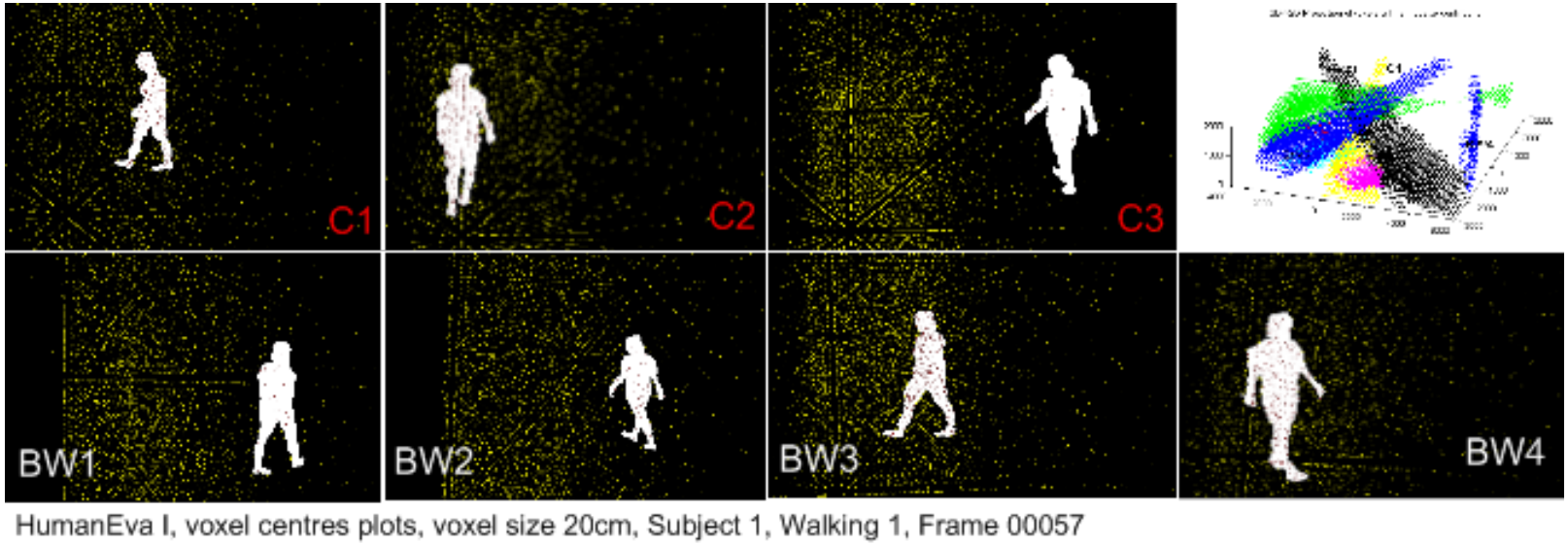

Figure 8.6: *HumanEva I, 7 cameras, S1, Walking 1, Frame 57. Coverage test with a large world volume. X=-3000:3000, Y=-6000:6000, Z=-300:2000, the units are millimeters. Only voxel centres are plotted.*

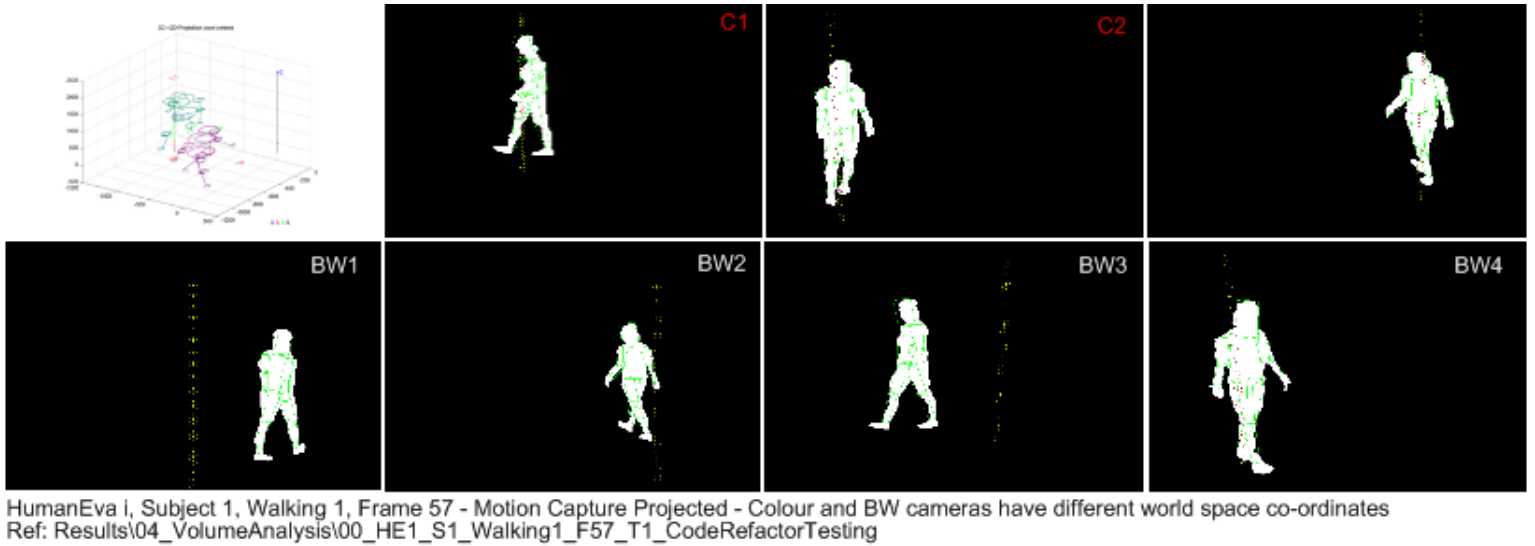

Figure 8.7: *HumanEva I, 7 cameras, S1, Walking 1, Frame 57. World volume check by projecting a small man height volume into the images. This allows the volume selected to be tested prior to running the longer volume reconstruction runs. In this case it was used to test if the colour and grey scale cameras were calibrated to the same world space co-ordinates.*

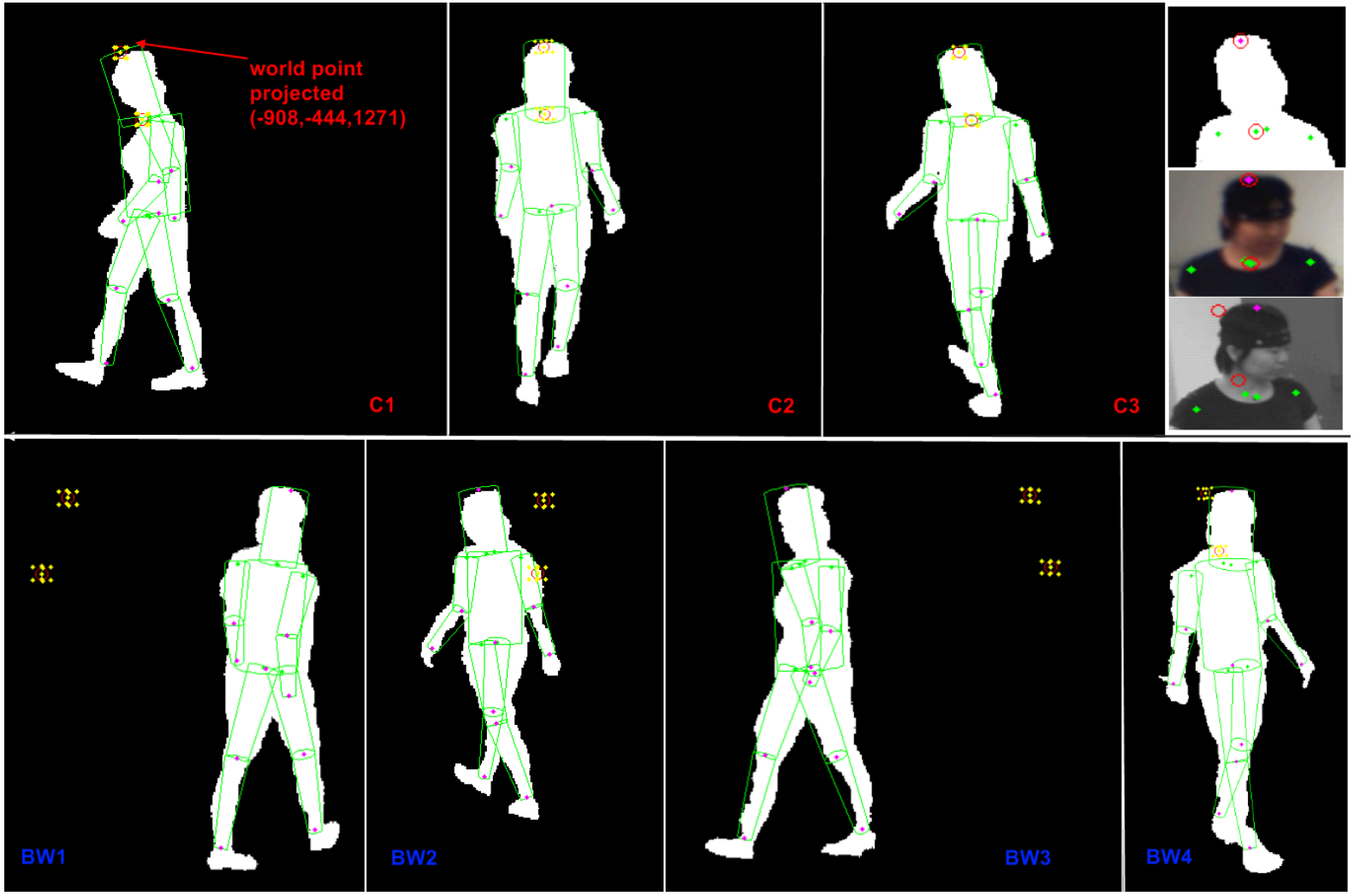

Figure 8.8: *HumanEva I, 7 cameras, S1, Walking 1, Frame 57. The world point form the marker at the top of the head of the subject was taken from the C1 motion capture stream and projected into all images. Found a match in the colour cameras but not in the grey scale cameras. If all cameras were calibrated to the same world space then it should have appeared on the top of the head of the subject in all images.*

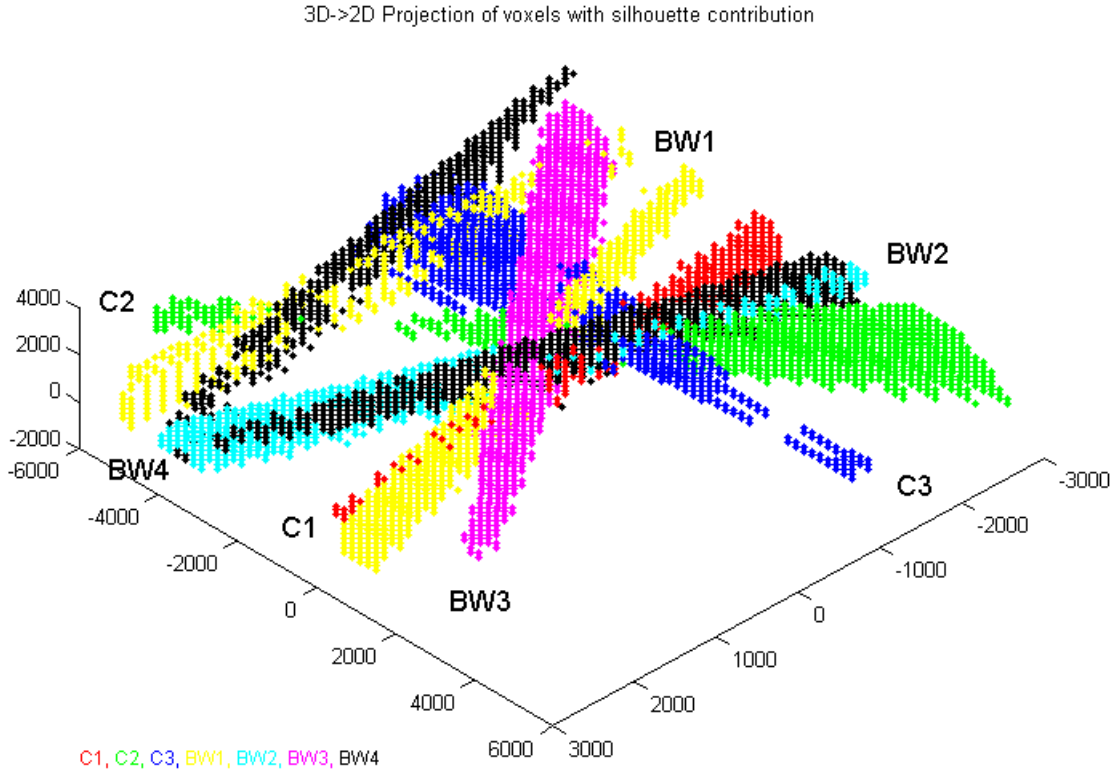

Figure 8.9: *HumanEva I, 7 cameras, S1, Walking 1, Frame 57. BW1 & BW4 cameras created split cones when the voxels contributing to their images were plotted. There may be an issue with their calibration or distortion parameters.*

8.1.3 3-D skeleton: 3 colour cameras OR 4 grey scale

If either the 3 colour cameras or the 4 grey scale cameras are used the subject volume reconstruction is quite good. There are a number of parameters involved in these reconstructions. The first is the size of the world volume scanned during the reconstruction. Using the motion capture data as described above the volume is placed about the subject, in a real-time application tracking or blob searching would be needed to position this volume as searching the complete room volume each frame is not an optimal solution. The second parameter is the size of the voxel, 50mm and 25mm reconstructions have been performed. The code implementation at this time is basic and not optimised, with a resolution of 50mm the reconstruction can take 30 minutes to an hour depending on the number of voxels, typically 8448 to 13440, if the volume surrounds the subject. The exact numbers are contained in the settings.txt files for each run. The third parameter, is a threshold, it determines how many images a voxel must project into the subject silhouette of before it is considered to be occupied. It is referred to in this text as the voxel hit rate. With 3 cameras and a voxel hit rate threshold of 3, a voxel is only recorded as occupied if it is seen by all three cameras. With a voxel hit rate equal to the number of cameras a relatively clean volume reconstruction can be obtained, the more cameras, the smaller the size of the voxel the greater the detail of the reconstruction. See figure 8.11.

One of the objects of this thesis was to infer the 3-D skeleton from the 2-D images. The premise being that the voxel hit rate values could be used to map the internal structure of the subject similar to a MRI scan. Therefore volume reconstructions runs were performed at various voxel hit values. The idea being that the main axis of the body would be seen in all cameras therefore it would have a high voxel hit rate. By visualising the voxel hit rate as colours a contour map could be drawn with a high density core representing the 3-D skeleton. With 7 cameras the analysis of this premise would have been possible with 3 and 4 cameras its not, as these cameras are needed just to form the visual hull. The reconstruction examples below illustrate the issues. Also setting a voxel hit rate less than the number of cameras results in additional detail at view extremities but results in outgrowths especially at larger voxel sizes, see figure 8.10.

8.1.4 Poor Background Segmentation issues

The examples above were created from manually generated silhouette images. The reconstructions generated from the GMM based background subtraction algorithm (Section 5) were not so successful. See figure 8.12. The volume reconstruction is actually hollow.

8.1.5 Analysis Runs

— Visual hull, camera alignment check - 7 cameras.

— Visual hull, camera alignment check - 3 colour cameras.

— Visual hull, camera alignment check - 4 grey scale cameras.

— Subject 1, Walking 1, Frame 57, 7 cameras, voxel size = 50mm, voxel hit threshold = 3.

— Subject 1, Walking 1, Frame 57, 4 grey scale cameras, voxel size = 50mm, voxel hit threshold = 3.

— Subject 1, Walking 1, Frame 57, 3 colour cameras, voxel size = 50mm, voxel hit threshold = 3.

— Subject 1, Walking 1, Frame 57, 3 colour cameras, voxel size = 50mm, voxel hit threshold = 2.

— Subject 1, Walking 1, Frame 57, 3 colour cameras, voxel size = 25mm, voxel hit threshold = 1.

— Subject 1, Walking 1, Frame 57, 3 colour cameras, voxel size = 25mm, voxel hit threshold = 3.

— Subject 1, Walking 1, Frame 91, 3 colour cameras, voxel size = 50mm, voxel hit threshold = 3.

— Subject 1, Walking 1, Frame 91, GMM Segmentation, 3 colour cameras, voxel size = 50mm, voxel hit threshold = 3.

— Subject 1, Walking 1, Frame 91, Fake Silhouette, 3 colour cameras, voxel size = 50mm, voxel hit threshold = 3.

— Work has commenced on Subject 3, throwing 1 and boxing 1.

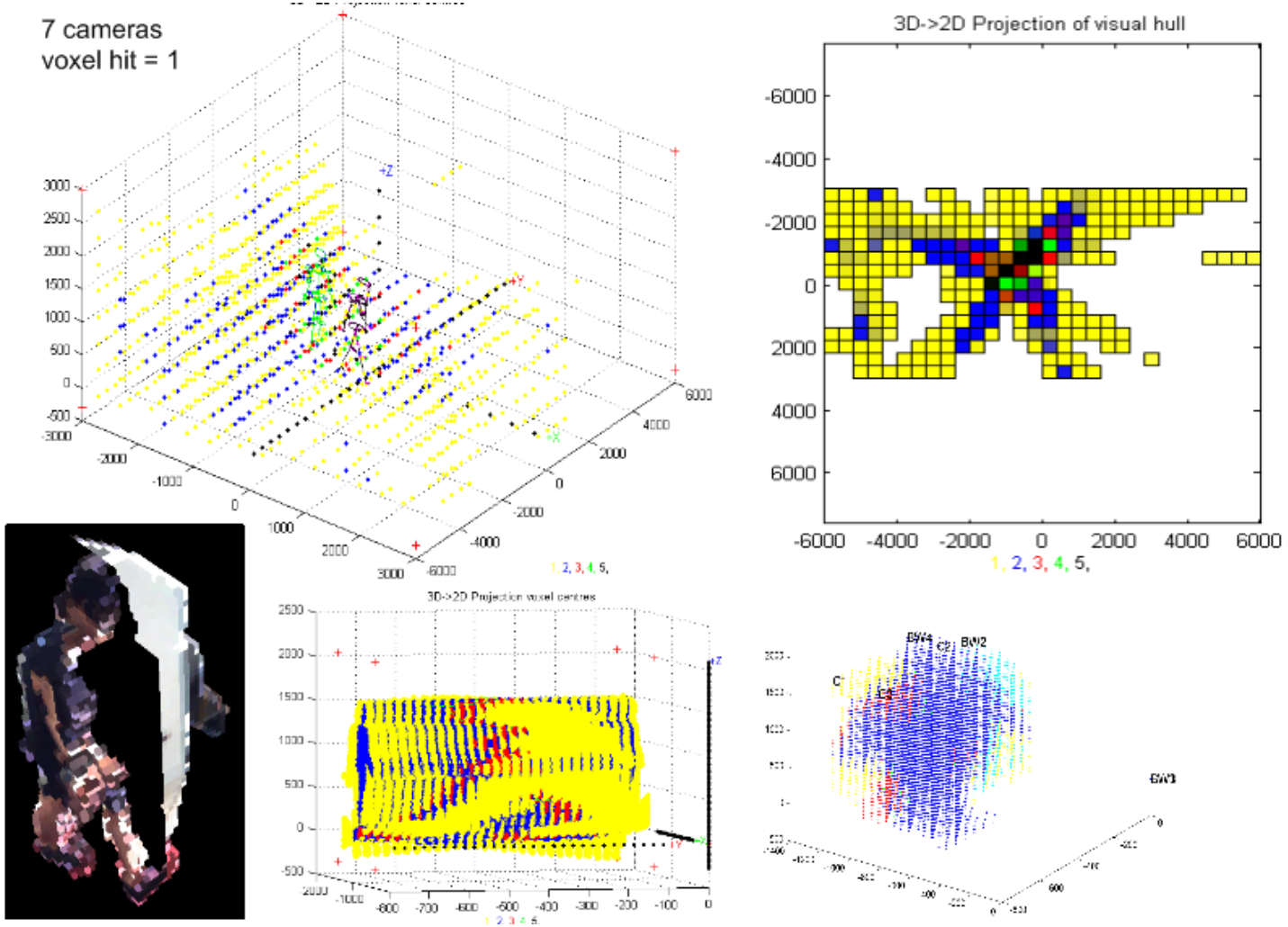

Figure 8.10: *HumanEva I, 7 cameras, S1, Walking 1, Frame 57. With a voxel hit threshold set to 1, the figure as plotted by motion capture data is lost in a cloud of voxels with low hit rates.*

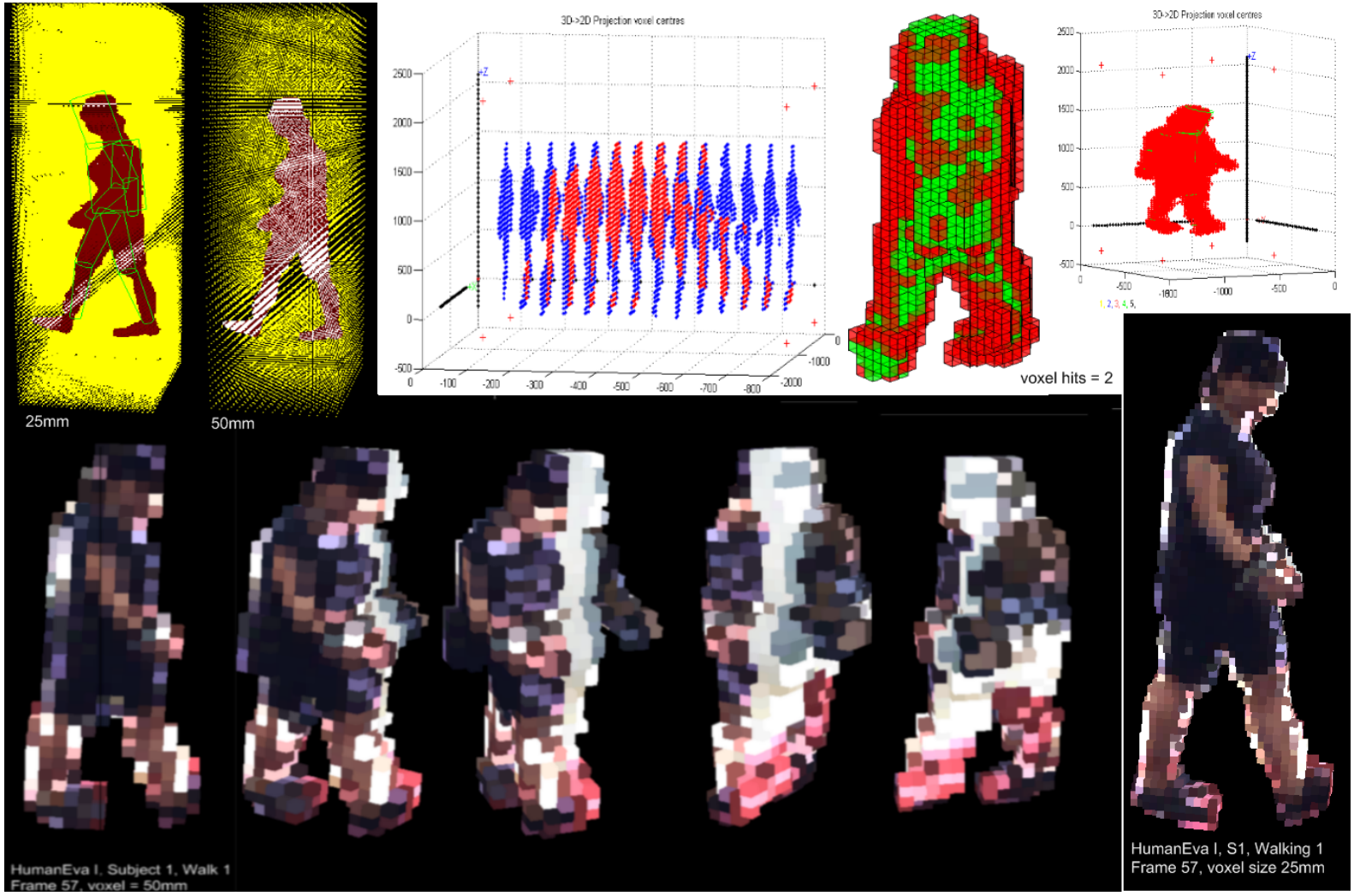

Figure 8.11: *Volume Reconstructions of Subject 1, Walking 1, at various parameter settings. The full results and plots can be found at Code/MCAvatar/Results/04_VolumeAnalysis/ the directory names are based on the parameters used during the reconstruction run e.g. 08_HE1_S1_Walking1_F57_T1_25mm_VoxelColour*

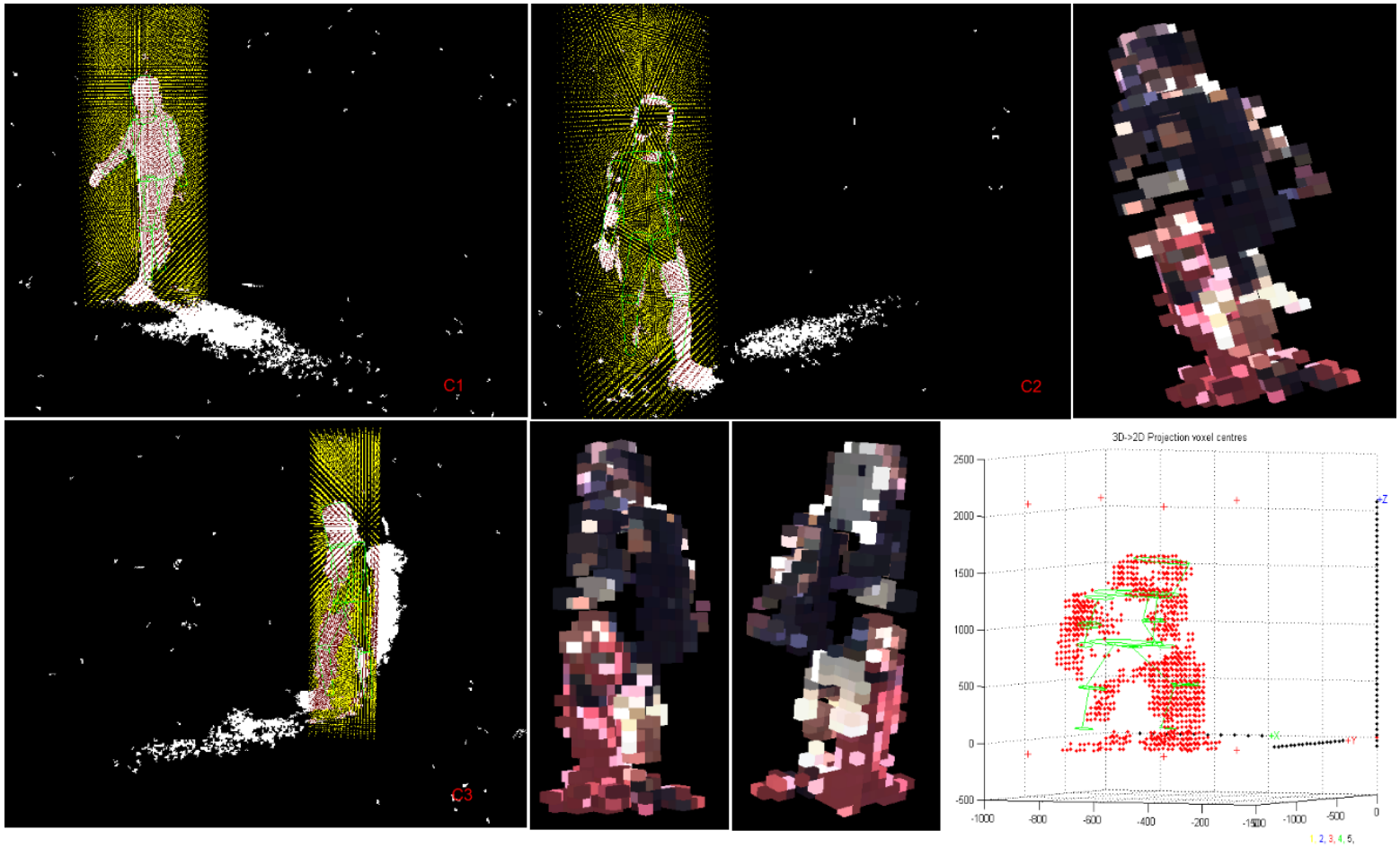

Figure 8.12: *Results of background subtraction based on the silhouettes generated by the GMM algorithim. See run directory : 09_HE1_S1_Walking1_F91_T3_50mm_GMM*

8.2 Evaluation

The conclusion from the analysis of the 7 camera configuration is that either:

— The colour and grey scale cameras have not been calibrated to the same world space

— Or that the colour and grey scale camera videos streams may need to be offset from each other

To reach conclusive proof that the colour and grey scale cameras are not calibrated to the same world space the full set of video streams for subjects S1 to S3 (there is no motion capture data for S4) and their various actions need to be sampled where the same tests as performed above are applied. The same issues were found for S1, walking 1, Frame 91. Analysis was also started on Subject 3, Throwing a ball 1, but the motion capture stream was not accessible due to a runtime error within the HumanEva demo code. Analysis on S3, Walking 1 was not completed.

In order to investigate the usefulness of deriving a 3-D skeleton in the manner proposed in this research more cameras are needed beyond the 3 or 4 needed to generate a reasonable visual hull. Alternatively the colour and grey scale data could be combined or possibly the separate time steps could be combined in order to build a model up over time.

8.3 Future Work

The motion data capture implementation is highly manual and has not been robustly tested. The current implementation has been hacked out of the HumanEva demo code. A real-time implementation of the motion capture data visualisation is required, similar to the HumanEva demo. Such an implementation would allow the synchronised extraction of images, silhouettes, fake silhouettes and motion capture data in a single run thus removing any doubts regarding the synchronisation of the images and motion capture data.

As mentioned above the issue regarding the world space co-ordinate calibration mismatch between the colour and grey scale cameras in HumanEva 1 needs further validation by sampling further subjects and their actions to determine if the issues are the same.

The HumanEva technical report mentions segments of the video stream that have issues. It would be worth investigating these sections of the dataset to see if the issues can be identified using the analysis tools and techniques developed here as they maybe issues that will also occur in real-time implementations.

The voxel colouring algorithm needs to be updated, whilst it works well with four cameras in the four corners of the room, artifacts are created if only three cameras are available.

Mundermannhad [71] also a reported a problem with the BW1 and BW4 cameras but did not give a reason. He simply excluded them from his visual hull analysis. His paper lists the frames he used. It would be worth while comparing subject volume reconstruction, where all 7 cameras are used, all colour and BW2+BW3, all colour and all grey scale (BW1 - BW4) to see the differences. Also to see how Mundermannhad was able to construct the visual hull if the camera types are not calibrated in the same world space. Or perhaps his success is an indication of a flaw in this implementation that needs to be corrected.

As mentioned in the future work section of 'Volume Rendering Code' (section 7.4) the techniques developed here need to be implemented first as a batch processing pipe line so that all aspects of the dataset can be analysed. With a batch processing pipeline in place the algorithms can be optimised prior to a real-time implementation being realised.

Chapter 9

Volume Analysis : HumanEva II

The HumanEva-II dataset [106] is smaller than the first but it has been generated using a hardware synchronised system with 4 colour video cameras and 12 Vicon motion capture cameras. The motion capture was recorded at 120 FPS and the video at 60 FPS. The video has been distributed as a sequenced of PNGs rather than as AVI files. It contains 2 test sequences with subjects S2 and S4 from the HumanEva-I set.

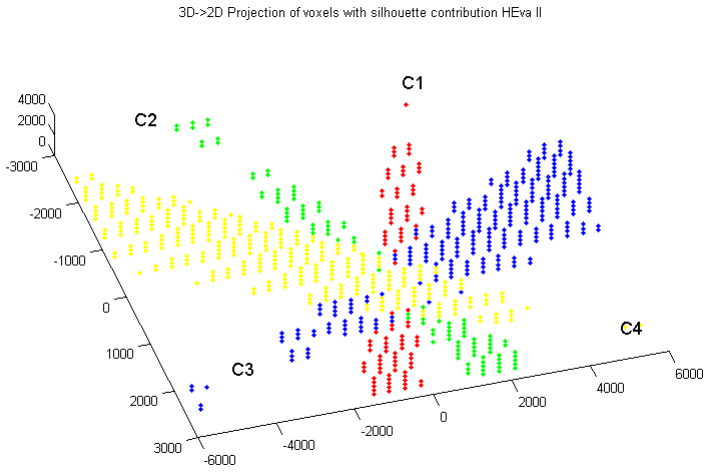

Figure 9.1: *HumanEva II, 4 colour cameras alignment checked by plotting the voxels contributing to the silhouette image in each camera*

9.1 Implementation

Similar processes and procedures were applied to the HumanEva II dataset as described in the previous chapters with the exception that this dataset does not contain AVI files. Therefore no background subtraction or video frame chopping was performed on the dataset. The silhouettes used for the initial analysis of this set were manually generated.
Analysis was performed primarily on Subject 2, combo 1 , frame 1. The motion capture FRAME is also equal to 1.
The following volume reconstructions analysis runs were performed:

— Visual hull, camera alignment check. See figure 9.1.

— Subject 2, Combo 1, Frame 1, voxel size = 50mm, voxel hit threshold = 3.(See figure 9.2)

— Subject 2, Combo 1, Frame 1, voxel size = 25mm, voxel hit threshold = 3.

— Subject 2, Combo 1, Frame 1, voxel size = 50mm, voxel hit threshold = 1.

— Subject 2, Combo 1, Frame 1, Fake Silhouette , voxel size = 50mm, voxel hit threshold = 3. See figure 9.3

The results are depicted below and can be found in the Code/Results/04_VolumeAnalysis directory. The subdirectories are named based on the parameters used e.g. 07_HE2_S2_Combo1_F1_T1_50mm.

9.2 Evaluation

The HumanEva II dataset with the hardware synchronised images and video is far more intuitive than the HumanEva I dataset synchronisation. No camera issues were encountered as in the HumanEva I dataset. With four cameras the basic voxel colouring algorithm works well unlike the results from HumanEva I's three cameras.

9.3 Future Work

As with the HumanEva I dataset a batch processing framework needs to be implemented to allow the algorithms and methods developed here to be applied to a larger set of data.
The C++ / OpenCV background segmentation programs need to be enhanced to take .PNG files and apply the background segmentation processes to these.

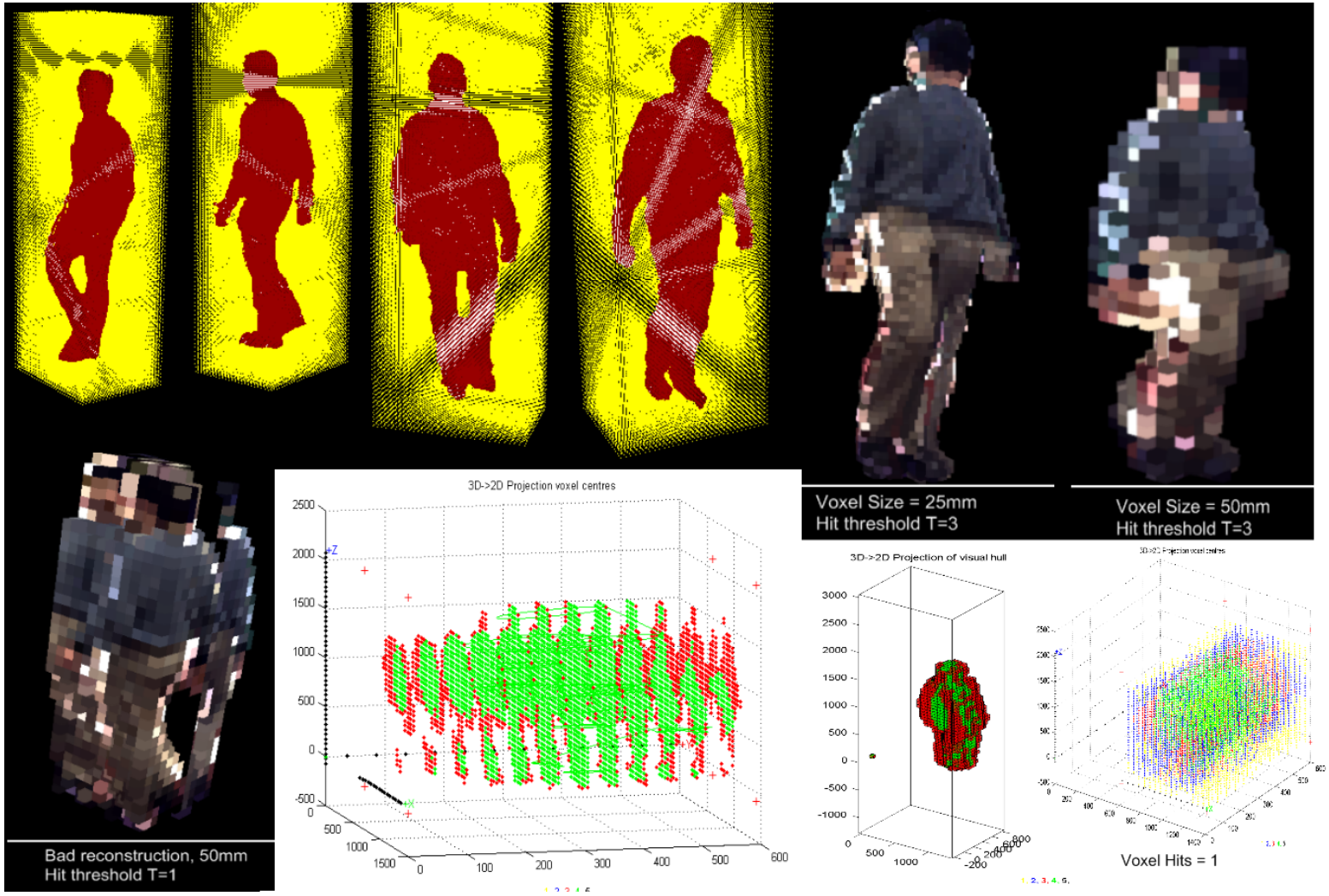

Figure 9.2: *HumanEva II, 4 cameras, S2, Combo 1, Frame 1.*

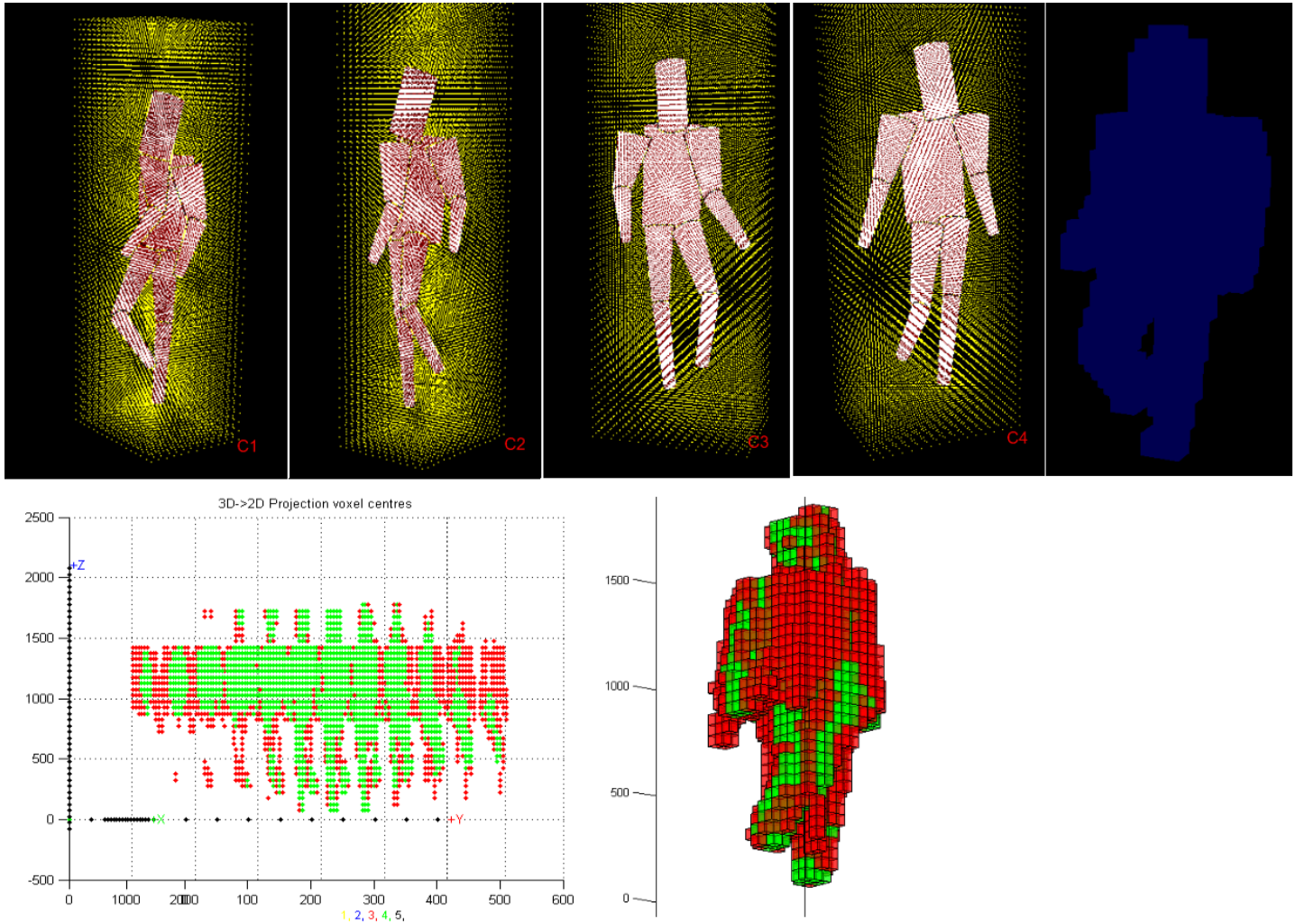

Figure 9.3: *HumanEva II, 4 cameras, S2, Combo 1, Frame 1. Volume Construction based on fake silhouettes generated from the motion capture data.*

Chapter 10

Conclusions

10.1 Abstract Review

The analysis research performed here is only the tip of the iceberg in term of the 10 years of research that has currently been devoted to the area of markerless human motion capture. This research does however take a particular slant on the research performed to date, that of using the current techniques to animate an avatar in a game, based on multiple camera inputs obtained within an unconstrained environment. The abstract directed the research in this domain as follows:

— Analyse the HumanEva datasets as a proof of concept for the implementation of a 3-D pose estimation system for real-time avatar animation within a single player game played in a family room installed with multiple stationary cameras.

— Infer a 3-D skeleton from multiple 2-D video streams where a single players body parts will be tracked in real-time.

The use of the HumanEva dataset allows this research to be conducted without the need to initially set up a multi-camera environment, as these dataset provide multiple camera video streams of four subjects performing a variety of actions with accompanying motion capture data.

10.2 Motivation Review

The motivation behind the research was to establish if computer vision was ready to be used as a human computer interface for a single player game within a completely unconstrained environment - a family

room. Upon commencing the research this researchers view of the abstract above was that the skeleton of the subject needed to be extracted, and the body parts / joints identified, in order to animate a 'model' avatar in a game. Where the model represented the users avatar in the game and needed to be rigged / animated with the skeleton data obtained through real-time tracking and pose estimation.

The state of the art review was conducted with one intent in mind - building a game where the player had a 3-D presence and could act within that world in three dimensions. The review revealed that there were a number of possible approaches to the problem. The inference of a 3-D skeleton and the subjects pose are of particular important to markerless human motion capture where the aim of the research is to accurately estimate the skeleton from 2-D video images so that the data could be used within medical applications, or within the entertainment industry for the life like animation of character models in films or the animation of player and non-player models in games. In the case of games the players presence and movement in the game are given only basic instructions - run, fire, walk, pickup - in the following direction. The model and its database of motion sequences are then interpolate to animated a sequence that bring the action of that character to life within the game world.

It becomes obvious upon reading the research work of de Aguiar et la [31] and Vlasic et la [116] that the reconstruction of the players volume and the animation of this volume as a mesh within the game would create a 3-D player presence in the game. The mesh is animated by the 3-D volume of their movements not by model based internal skeleton rigging. So long as their movements are relayed to the game world in a perceptually timely manner, 25 milliseconds to 50 milliseconds [48] - depending on the nature of the game, then the player can interact with a collision detection based game in real-time. If the player wants to take on the persona of different characters the mesh can be deformed or other meshes can be grafted to it.

Taking the concept of a subjects volume being reconstructed in 3-D by multiple cameras leads to a metaphorical analogy with MRI scanning. The reconstruction of a subjects volume is conducted based on checking if a point in world space defined by a voxel is within the silhouette of the subject in all the camera images. If there are a sufficient number of cameras the core of the body would be in more camera images than the outer extremities. Thus plotting the voxel image projection counts should provide a means of inferring the 3-D skeleton of the subject as a contour map of the subjects volume could be visualised.

10.3 Goal Review

The goal of the research was to ...

— Review the current techniques to assess their applicability to the domain under review - a family room.

— Implement a basic multi-camera subject reconstruction pipeline based on the HumanEva I and II datasets.

— Conduct the implementation and implementation modelling with the view to the completed system being developed in C++ / OpenCV within the OGRE game engine.

10.4 Results

Whilst the volume reconstruction pipeline developed during this research is not real-time, nor is it optimised in terms of the algorithms used and their implementation. It does however meet the goals of the research ...

— The implementation of a basic pipeline that provides a model for a C++ / OpenCV implementation.

— The reconstruction of the volumes of subjects within the HumanEva I and II dataset based on data obtained from multiple 2-D video streams, see figures 10.2 and 10.2.

— The generation of data for the further analysis of the MRI based 3-D skeleton estimation approach.

— Analysis of the HumanEva I and II dataset from the perspective of using these datasets for further research in this area and the identification of issues within the datasets.

— Implementation and evaluation of a number of background subtraction methods in C++ / OpenCV.

— Implementation of a set of tools and approached for analysing multi-camera video streams and generating 3-D reconstructions of the subjects in MATLAB. These methods do not rely on any HumanEva or MATLAB camera calibration code, though these routines were used as ground truth for evaluating their implementations.

— The OGRE game engine implementation was not completed but a file format was generated that allows the volume reconstruction to be viewed and rotated in a voxel viewer used by voxel game element designers.

As discussed in the evaluation and future work sections on Background Subtraction (sections 5.2, 5.3), Volume Rendering Code Overview (sections 7.3, 7.4) and HumanEva I Volume Analysis (sections 8.2, 8.3) there is significant potential for multi-processor or GPU (Graphics Processing Unit) optimisation

of the processing pipeline. This is extremely practical in this environment as the main games consoles, XBOX360 [4] and PS3 [55] contain GPU's and multiple processors.

This research area falls under the heading of Markerless Human Motion Capture though Intelligent Spaces research is also an appropriate categorisation. Adding the game avatar dimension places the research into the realms of perception relevant real-time processing rather than accuracy relevant real-time processing which is the predominant interest of Markerless Human Motion Capture research. Where perception in this context implies acceptable player perception of the avatars movement and interaction within a virtual 3D space. Viewing the problem from this perspective and taking account of the research taking place in terms of 3-D video leads this researcher to believe that a mesh based approach to the animation of the game avatar is practical where multiple cues are implement in the extraction of the players silhouette from the video streams and the solution is deployed on a multi-processor platform.

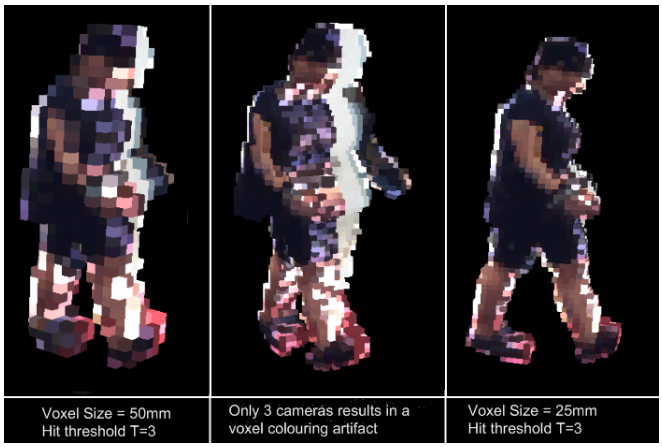

Figure 10.1: *HumanEva I, Subject 1, voxel representation of the subject where basic voxel colouring has been applied to each voxel based on the colour of the pixel of the nearest camera at the voxels centre*

10.5 Analysis of the HumanEva dataset

A significant amount of time was spend analysing the HumanEva dataset diagnosing the following issues:

— The colour and grey scale cameras in the HumanEva I data set do not appear to be calibrated to the same world space. A definitive statement in this regard required the implementation of the batch processing version of the code implemented here. This would provide better data cover and analysis the issues. The investigations, evaluations and conclusions formed to date can be found in

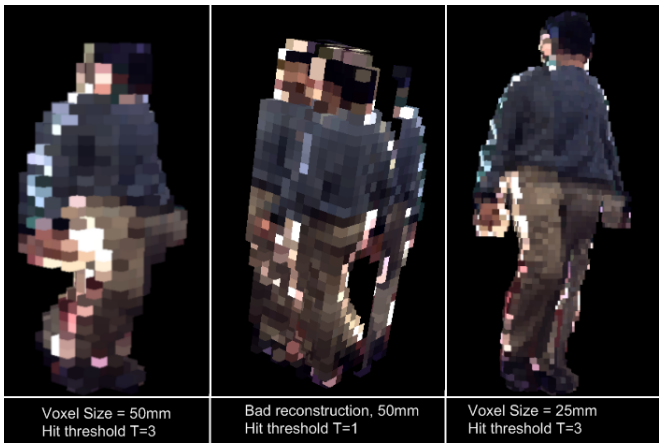

Figure 10.2: *HumanEva II, Subject 2, voxel reconstruction*

the HumanEva I Volume Analysis chapter, sections 8.1.2, 8.2 and 8.3.

— The HumanEva1 motion capture to image syncing is problematic producing invalid results for certain frames. To feel completely confident of the motion capture to image synchronisation an implementation similar to the HumanEva demo software needs to be implemented where per frame - image, silhouette, fake silhouette and motion capture data are generated. The 3-D volume reconstructions based on the motion capture data (fake silhouettes) and image silhouettes can then be compared. This would also allow the joint positions to be overlayed on the 3-D reconstructions. It should be noted that the cylinders especially in the head region are larger than the actual area of the subject.

— There are problems with two of the HumanEva I grey scale cameras - BW1 and BW4. This issue was also reported by Mundermannhad [71] but no explanation was provided. On plotting the voxel contributing to each cameras silhouette image, figure 8.9, it can be seen that the cone representing the volume of space contributing to their images is split. This is potentially the result of inaccurate calibration or distortion co-efficients.

— Other issues within the dataset such as the environment, clothing and bad background data made this environment closer to a real world environment and thus good for the research analysis conducted here.

10.6 3-D Skeleton

The skeleton reconstruction method investigated requires additional cameras or the combination of the data across time. HumanEva effectively has a 3 camera (colour) and 4 camera (grey scale) environment configuration not a 7 camera configuration which means that the volume reconstructions are crude. Potentially a 5 camera configuration could be implemented if cameras BW1 and BW4 are excluded. The HumanEva II dataset has 4 cameras. A review of the voxel occupancy contours with a threshold set at 3, one below the number of cameras does indicate that there is some potential in this data but further analysis is required. See figure 10.3.

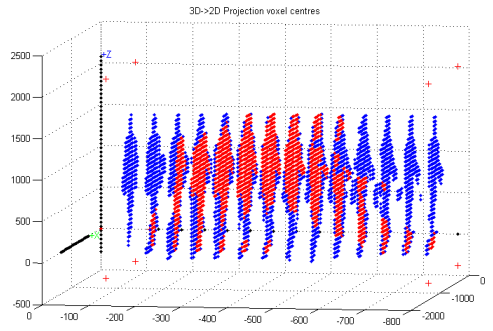

Figure 10.3: *Contour slices through the 3-D volume of a subject*

10.7 Research Challenges

Some of the research challenges in this area have been over come but many remain to be met in future work. The challenges are as follows:

— Investigation, analysis and interpretation of the HumanEva I & II multi-camera video and motion capture data.

— Efficient and accurate subject segmentation and the creation of silhouettes from a noisy environment where the subjects is partly indistinguishable from the background.

— Efficient volume reconstruction of the subject given the available camera configurations.

— Progression from offline techniques to real-time accurate subject reconstruction with regard to player perception.

— The real challenge is that many of the markerless human motion capture techniques have been investigated and deployed in constrained environments. In order to create a robust real-time, real world application, the assumptions listed in section 2.1 need implementation specific solutions and a degree of automatic adaptability that is not currently being adequately addressed in current research.

10.8 Future Work

Detail evaluation and future work sections are included at the end of the following chapters:

— Evaluation of Background Subtraction Methods, chapter 5.

— Volume Rendering Code Overview, chapter 7.

— Volume Analysis : HumanEva I, chapter 8.

— Volume Analysis : HumanEva II, chapter 9.

In brief, the continued analysis and implementation of the research work presented here requires the following approach:

1. Implementation of the MATLAB volume reconstruction in C++ / OpenCV.
2. Implementation of a processing pipe line similar to the HumanEva demo code to allow per frame - image, silhouette, fake silhouette and motion capture data are generated thus ensuring synchronisation of the various data components in the HumanEva dataset.
3. Implementation of a batch processing pipe line from silhouette generation through to volume reconstruction and test automation.
4. Implement technique / algorithm improvements within the batch pipeline in the following areas:
   — Silhouette creation based on multiple cues e.g. motion and colour
   — Silhouette cleaning techniques e.g. morphology, connected component analysis, photo-consistency.
   — Centroid / blob recognition to centre volume reconstruction over subject silhouette.
   — Harr based facial recognition techniques to determine the subjects orientation.
   — Restrict shadow processing to set regions.
   — Implement improved voxel colouring techniques e.g. SPOT [21].

— Improved volume traversal methods by creating a 3-D to 2-D lookup table rather than performing the projection calculation each time [65]

— Implement a mesh generation technique e.g. marching cubes [63].

— Implement quantitative measurement tools for assessing the quality and efficiency of the improvements.

5. Based on the batch processing pipeline, implement real-time tracking algorithms, evaluate their quality and performance.

6. Implement multi-processor / GPU optimisation into the batch processing pipeline. Review the algorithms and techniques previously implemented in terms of their multi-processor implementation efficiency verses their output quality and performance.

7. Review, evaluate and implement real-time mesh generation and deformation techniques. Evaluate their performance within the context of real-time game play.

8. Implement a real-time multi-processor variant of the batch processing pipeline and analyse the HumanEva I and II datasets particularly in the areas designated as problematic.

9. Source other video and motion capture datasets for analysis.

10. Build a multi-camera environment within an environment simulating a family room. Implement a game collision detection based game using the developed real-time multi-processor software and analyse the results. The game could be as simple as dodging virtual objects thrown at the players 3-D avatar, basically 3-D asteroids.

10.9 Contribution

The contribution of this research are as follows:

— Development and analysis of a new technique for extract the 3-D skeleton of a subject from the multiple camera views. Whilst other researchers have used silhouettes and voxels to interpret the 3-D volume of a subject their interest has been in the creation of a visual hull for the subject. The volume occupied and the internal voxels are purged in favour of the outer surface voxels. The method used here creates a density contoured volume of the subject similar to an MRI scan. This technique allows the inner skeleton of the subject to be reliably obtained whilst also leaving more detail than shape from silhouette techniques.

— Analysis of the current research in the area in terms of using markerless vision techniques to animate an avatar in a game.

— Analysis of the HumanEva I and II dataset with respect to volume based subject reconstruction techniques and real-time subject segmentation techniques. Development of tools and tests in OpenCV/C++ and Matlab in order to perform the analysis.

— Identification of issues within the HumanEva I data set in relation to the calibration of the colour and grey scale cameras.

  - They are not calibrated to the same world space and so are not immediately usable for visual hull construction as a 7 camera configuration.
  - The grey scale cameras BW1 and BW4 cameras have calibration issues as also identified by Mundermannhad [71].

— An implementation approach to the area defining the key modules required to build a system which would allow computer vision techniques to be used to animate an avatar in a game.

— Identification of the potential issues and challenges involved in building a system which would allow computer vision techniques to be used to animate an avatar in a game.

— Outline of an approach to completing the research above and implementing a mesh based game avatar within a collision detection based game.

Appendix A - Glossary of Terms

Articulated Objects Articulated objects are composed of a set of rigidly moving parts which are connected to each other at joints which allow movement e.g. articulation points [22].The human body is a good example of an articulated object.

Bayesian network A probabilistic graph that represents the probabilistic interdependence between a set of variables. The graph allows prior knowledge to be used with a sample set of all possible observations in order to infer information or make valid predictions. Classical inferential models do not permit the introduction of prior knowledge into the calculations. Bayes' Theorem, developed by the Rev. Thomas Bayes, and first published in 1763 is the foundation for expressing and evaluating a Bayesian network. The theorem provides a value for P(H |E,c), which is our belief in hypothesis H given the additional evidence E and the background context c. P(H |E,c) is known as the "posterior probability" or the probability of H after considering the effect of E on c. Another perspective on the value P(H |E,c) is the "likelihood" value, gives the probability of the evidence assuming the hypothesis H and the background information c is true [75].

Bio-mechanics The problem of determining the mechanical energy expended for motion.

Camera model The camera model relates the pixel co-ordinates in an image to their equivalent co-ordinates in the real world.

Camera Calibration Camera calibration is the process by which the parameters of the camera model are calculated such that a 3-D to 2-D projection matrix for the camera can be calculated.

Degrees of freedom (DOF) The movement of a body in a co-ordinate system with respect to time can be described as a combination of a translation and a rotation. The translations at any point in time can be described in terms of three perpendicular (orthogonal) translations and a rotation can be described as three perpendicular rotations. So within a 3-D space there are potentially six degrees of freedom, in 2-D

space three degrees of freedom. If a body cannot perform one of the possible degrees of freedom then it is said to have an obstruction. So degrees of freedom can be defined as the number of relative independent movements a body has within a specific co-ordinate system at a particular moment in time.

Intensity profile analysis This is a motion detection method [83]. Intensity profile analysis classifies all pixels in a single frame with the following attributes: Moving , Stationary or Background.

Kinematic chains Kinetic chains describe objects and how they are connected. They may be described in terms of movement as in the degrees of freedom of the connected system. Two types of kinematic chains are considered, open and closed kinematic chain. In a closed chain every object is connected to any other object by at least two distinct paths. An open kinematic chain has one and only one distinct path that connects objects in the chain.

The degrees of freedom of a closed kinematic chain are given by the following formula:

DOF = ((i - 1) * D) - O

DOF = the number of degrees of freedom of the chain relative to one object in the chain. i = the number of independent objects in the chain, less the object taken as the movement reference point D = the number of possible degrees of freedom in the dimensional space, 3-D = 6, 2-D = 3. O = the number of obstructions to movement, DOF limitations based on the connections. This is calculated based on the number of DOF an object has minus the potential DOF for the dimensional space under consideration, summed for each connection point.

Kinematic pairs Kinematic pairs refers to the connection or joint between two objects in a kinematic chain that allows relative motion between the objects.

Matting Matting is another term for foreground / background segmentation and refers to the problem accurately estimating the foreground in images and video. the term is more widely used in many image editing and film production environments.

Motion Verbs Motion verb recognition is the process of associating natural language verbs with the motion performed by a moving object in a sequence of images.[17]

Octree An octree is a tree data structure where each node has up to 8 children. In this context it is being used to partition the three dimensional volume being observed by multiple cameras. The volume is partitioned by recursively subdividing it into eight octant's. [37]

Priori Shape Modelsv In statistics, a priori means knowledge present before a particular observation is

made, and a posteriori is knowledge once the outcome of the observation has been taken into account. Some approaches to human pose estimation rely on a model of a human while others dont. Papers that refer to a priori subject model being used are model based pose estimation approaches.

Shape-from-Silhouette(SFS) Shape-from-Silhouette is also known as Visual Hull (VH) construction is a 3-D reconstruction method that provides an estimate of the shape of an object based on multiple silhouette images [23]. The output from SFS is a Visual Hull (VH). The concept was first introduced by Baumgart in 1974 [9].

Twists and Exponential Maps "Twists representation is based on the observation that every rigid motion can be represented as a rotation around a 3-D axis and a translation along that axis. Exponential maps convert a twist representation into a standard rotation matrix and translation vector" [65] Twists are described in [72] which is a book providing a mathematical introduction to robotic manipulation. Twists are based on screw theory which can be traced back to Chasles and Poinsot in the early 1800s. Screw motion proves that a rigid body can be moved from any one position to another position by a rotation about a straight line followed by a translation parallel to that line. An infinitesimal version of a screw motion is called a twist. It describes the instantaneous velocity of a rigid body in terms of its linear and angular components. Screws and twists are used in the formulation of robot kinematics.

Voxel A three-dimensional pixel. The smallest unit of three-dimensional space in a computer image, equivalent to a three-dimensional pixel. Voxel Reconstruction A technique for reconstructing a 3 dimensional representation of an object based on images of the object obtained from multiple cameras. The volume captured by the the multiple cameras is divided into unit volumes. These unit volumes are called voxels, as in volume pixels. In order to reconstruct a 3-D volume, a model of the imaging process as it relates to each of the cameras is required. This allows the positions in 3-D space to be mapped to image positions in 2-D space (the image plane). The object must also be segmented from the background to create a silhouette of the object. Each voxel, for each camera view is then projected into the 2-D image plane for that camera and its location is compared with the location of the silhouette to determine if that voxel is an volume element of the object. The voxel must be within the silhouette for all camera projections for it to be considered a voxel of the object. [65]

Visual Hull The term was first coined by Laurentini in [57] to describe the 3-D shape obtained by intersecting the silhouettes obtained for an object from the visual cones of multiple cameras. The visual hull is defined as the maximal object that can reconstruct the original objects silhouette from any viewpoint [57], in other words it provides an upper bound on the shape of the object. The visual hull is a geometric

representation of an objects which relates the 3-D shape of a concave object to its silhouettes or shadows. [59].

Appendix B - 3-D to 2-D Projection Code

```
function pixelPoints = GetPixels( worldPointsXYZ,omc,rc,tc,fc,cc,kc,alpha_c,...
    intrinsicMatrix, extrinsicMatrix)

% GETPIXEL
% calculate the 2D pixel position of a 3D world point based on the a
% calibrated camera matrix
%
% Returns a 2xN matrix where N is the number of points sent in to be
% transformed
%
% Two implementations are presented here, only calc type 1 is used.
%
% Ref:
% http:///www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.htm
%

%number of points to be converted
[tmp,i] = size(worldPointsXYZ);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CALC type 1 - with distortation accounted for

R = [ rc(1,1), rc(1,2), rc(1,3),0;...
      rc(2,1), rc(2,2), rc(2,3),0;...
      rc(3,1), rc(3,2), rc(3,3),0;...
      0,       0,       0 ,     1 ];
```

```
T = [ tc(1,1); tc(2,1); tc(3,1); 1 ];

for k = 1:i

    % (1) Change world co-ordinates to the cameras co-ordinate frame of
    %     reference by applying the rotation matrix and translation vector
    %     Xc = R * Xw + T equivalent to row multiply and add below

    worldXYZ = [worldPointsXYZ(1,k);worldPointsXYZ(2,k);worldPointsXYZ(3,k);1];

    cameraXYZ = R * worldXYZ + T;

    % (2) Convert to pin-hole camera 2D co-ordinates on image plane

    u = cameraXYZ(1,1) / cameraXYZ(3,1);
    v = cameraXYZ(2,1) / cameraXYZ(3,1);

    % (3) Adjust 2D pin-hole co-ordinates for distortion so ud = u distorted

    r = u * u + v * v; % r = r^2

    ud = ( 1 + kc(1)*r + kc(2)*r*r + kc(5)*r*r*r ) * u + ...
          2*kc(3)* u * v + kc(4)*( r + 2*u*u );

    vd = ( 1 + kc(1)*r + kc(2)*r*r + kc(5)*r*r*r ) * v + ...
          kc(3) * ( r + 2 * v * v ) + 2 * kc(4) * u * v ;

    % (4) Adjust the co-ordinated based on the intrinsic properties of the camera
    %  focal length, center points and skew

    pixelPoints(1,k) = fc(1) * ( ud + alpha_c * vd ) +  cc(1);
    pixelPoints(2,k) = fc(2) *  vd +  cc(2);

    v=0;
    u=0;
    ud = 0;
    vd = 0;
    r = 0;
    clear worldXYZ;
    clear cameraXYZ;
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CALC type 2 - use combined camera matrix
% Distortion not taken into account
% Implemented to test differences in pixel point generation

%THIS CODE IS COMMENTED OUT
%{
projMatrix = intrinsicMatrix * extrinsicMatrix;

for k = 1:i

    worldXYZ = [worldPointsXYZ(1,k);worldPointsXYZ(2,k);worldPointsXYZ(3,k);1];

    XYW = projMatrix * worldXYZ;

    x = XYW(1,1) / XYW(3,1);
    y = XYW(2,1) / XYW(3,1);

    pixelPoints(1,k) = x;
    pixelPoints(2,k) = y;

    x=0;
    y=0;
    clear worldXYZ;

end
%}
% END OF COMMENTED OUT CODE

return;
```

Bibliography

[1] AGGARWAL, J., AND CAI, Q. Human motion analysis: a review. *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE* (Jun 1997), 90–102.

[2] AHMED, N., DE AGUIAR, E., THEOBALT, C., MAGNOR, M., AND SEIDEL, H.-P. Automatic generation of personalized human avatars from multi-view video. In *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology* (Monterey, USA, December 2005), Association for Computing Machinery (ACM), ACM, pp. 257–260.

[3] AMEDI, A., STERN, W., CAMPRODON, J., BERMPOHL, F., MERABET, L., ROTMAN, S., HEMOND, C., MEIJER, P., AND PASCUAL-LEONE, A. Shape conveyed by visual-to-auditory sensory substitution activates the lateral occipital complex. *Nature Neuroscience 10* (2007), 687–689.

[4] ANDREWS, J., AND BAKER, N. Xbox 360 system architecture. *Micro, IEEE 26*, 2 (March-April 2006), 25–37.

[5] ARIKAN, O., FORSYTH, D. A., AND O'BRIEN, J. Motion synthesis from annotations. In *ACM Transactions on Graphics (ACM SIGGRAPH 2003)* (2003), vol. Vol: 33, No: 3, pp. pp 402–408.

[6] BAILENSON, J. N., AND YEE, N. The use of immersive virtual reality in the learning sciences: Digital transformations of teachers, students, and social context. *THE JOURNAL OF THE LEARNING SCIENCES 17* (2008), 102141.

[7] BALAN, A., BLACK, M., HAUSSECKER, H., AND SIGAL, L. Shining a light on human pose: On shadows, shading and the estimation of pose and shape. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (2007), 1–8.

[8] BALAN, A., SIGAL, L., AND BLACK, M. A quantitative evaluation of video-based 3d person tracking. *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on* (2005), 349–356.

[9] BAUMGART, B. *Geometric Modeling for Computer Vision*. PhD thesis, Stanford University, 1974.

[10] BEBIS, G. Image formation basics. `http://www.cse.unr.edu/~bebis/CS791E/`, 2008.

[11] BOBICK, A. F., INTILLE, S. S., DAVIS, J. W., PINHANEZ, C. S., CAMPBELL, L. W., IVANOV, Y. A., TTE, A. S., WILSON, A., AND ENVIRONMENT, I. S. The kidsroom: A perceptually-based interactive and immersive story environment. In *PRESENCE* (1999), pp. 367–391.

[12] BOBICK, A. F., AND JOHNSON, A. Y. Gait recognition using static, activity-specific parameters. In *CVPR (1)* (2001), pp. 423–430.

[13] BOUGUET, J.-Y. `http://www.vision.caltech.edu/bouguetj/calib_doc/`, camera calibration toolbox for matlab.

[14] BREGLER, C. Learning and recognizing human dynamics in video sequences. In *IEEE Conf. on Computer Vision and Pattern Recognition* (1997).

[15] BROADHURST, A. A probabilistic framework for space carving. In *ICCV, page(s): 388 - 393* (2001).

[16] BROWN, D. Decentering distortion of lenses. *Photometric Engineering Vol. 32 , No. 3* (1966), pages 444–462.

[17] CEDRAS, C., AND SHAH, M. Motion-based recognition: A survey. *Image and Vision Computing 13*, 2 (March 1995), 129–155.

[18] CHEN, H. H. Determining motion and depth from binocular orthographic views. *CVGIP: Image Understanding 54* (July 1991), 47–55.

[19] CHENG, S. Y., AND TRIVEDI, M. M. Articulated body pose estimation from voxel reconstructions using kinematically constrained gaussian mixture models: Algorithm and evaluation. In *CVPR 2007, Workshop: EHuM2: 2-nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation* (2007).

[20] CHEUNG, G., BAKER, S., HODGINS, J., AND KANADE, T. Markerless human motion transfer. *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on* (Sept. 2004), 373–378.

[21] CHEUNG, G., KANADE, T., BOUGUET, J.-Y., AND HOLLER, M. A real time system for robust 3d voxel reconstruction of human motions. *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on 2* (2000), 714–720 vol.2.

[22] CHEUNG, G. K. M., BAKER, S., AND KANADE, T. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *CVPR (2)* (2003).

[23] CHEUNG, K. M., BAKER, S., AND KANADE, T. Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (June 2003).

[24] CHEUNG, K. M., BAKER, S., AND KANADE, T. Shape-from-silhouette across time part i: Theory and algorithms. *International Journal of Computer Vision 62*, 3 (May 2005), 221 – 247.

[25] CHEUNG, K. M., BAKER, S., AND KANADE, T. Shape-from-silhouette across time: Part ii: Applications to human modeling and markerless motion tracking. *International Journal of Computer Vision 63*, 3 (August 2005), 225 – 245.

[26] COLLINS, R. T., LIPTON, A. J., KANADE, T., FUJIYOSHI, H., DUGGINS, D., TSIN, Y., TOLLIVER, D., ENOMOTO, N., HASEGAWA, O., BURT, P., AND WIXSON, L. A system for video surveillance and monitoring. Tech. rep., The Robotics Institute, Carnegie Mellon University, Pittsburgh PA and The Sarnoff Corporation, Princeton, NJ, 2000.

[27] COLOMBO, C., DEL BIMBO, A., AND VALLI, A. Real-time tracking and reproduction of 3d human body motion. *Image Analysis and Processing, 2001. Proceedings. 11th International Conference on* (Sep 2001), 108–112.

[28] COVELL, RAHIMI, HARVILLE, AND DARRELL. Articulated-pose estimation using brightness- and depth-constancy constraints. In *Computer Vision and Pattern Recognition* (2000).

[29] CUTLER, R., AND DAVIS, L. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 22* (2000), page(s): 781–796.

[30] DE AGUIAR, E. Character animation from a motion capture database. Master's thesis, Universität des Saarlandes, November 2003.

[31] DE AGUIAR, E., STOLL, C., THEOBALT, C., AHMED, N., SEIDEL, H.-P., AND THRUN, S. Performance capture from sparse multi-view video. In *SIGGRAPH* (2008).

[32] DE AGUIAR, E., THEOBALT, C., MAGNOR, M., AND SEIDEL, H.-P. Reconstructing human shape and motion from multi-view video. In *2nd European Conference on Visual Media Production (CVMP)* (London, UK, December 2005), The IEE, pp. 42–49.

[33] DE AGUIAR, E., THEOBALT, C., MAGNOR, M., THEISEL, H., AND SEIDEL, H.-P. M3 : Marker-free model reconstruction and motion tracking from 3d voxel data. In *12th Pacific Conference on Computer Graphics and Applications, PG 2004* (Seoul, Korea, October 2004), D. Cohen-Or, H.-S. Ko, D. Terzopoulos, and J. Warren, Eds., IEEE, IEEE, pp. 101–110.

[34] DE AGUIAR, E., THEOBALT, C., THRUN, S., AND SEIDEL, H.-P. Automatic conversion of mesh animations into skeleton-based animations. vol. 27, pp. xx–xx.

[35] DELAMARRE, Q., AND FAUGERAS, O. 3d articulated models and multi-view tracking with silhouettes. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on 2* (1999), 716–721 vol.2.

[36] DEUTSCHER, J., BLAKE, A., AND REID, I. Articulated body motion capture by annealed particle filtering. In *Computer Vision and Pattern Recognition, Volume 2, Page(s):126 - 133* (2000).

[37] DYER, C. R. *Foundations of Image Understanding - Volumetric Scene Reconstruction from Multiple Views*. No. Ch. 16. Kluwer, Boston, 2001.

[38] FREEMAN, R., AND STEED, A. Interactive modelling and tracking for mixed and augmented reality. In *VRST '06: Proceedings of the ACM symposium on Virtual reality software and technology* (New York, NY, USA, 2006), ACM, pp. 61–64.

[39] FRIEDLAND, G., AND ROJAS, R. Anthropocentric video segmentation for lecture webcasts. *J. Image Video Process. 8*, 2 (2008), 1–10.

[40] GAVRILA, D. The visual analysis of human movement: A survey, 1999.

[41] GAVRILA, D., AND DAVIS, L. 3-d model-based tracking of humans in action: a multi-view approach. *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on* (Jun 1996), 73–80.

[42] GRIESSER, A., DE ROECK, S., NEUBECK, A., AND VAN GOOL, L. Gpu-based foreground-background segmentation using an extended colinearity criterion. In *Vision, Modeling, and Visualization (VMV)* (2005).

[43] HARTLEY, R., AND ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. Cambridge, 2000.

[44] HILTON, A. Towards model-based capture of a persons shape, appearance and motion. In *MPEOPLE '99: Proceedings of the IEEE International Workshop on Modelling People, page(s) 37* (1999).

[45] HORPRASERT, T., HARWOOD, D., AND DAVIS, L. S. A statistical approach for real-time robust background subtraction and shadow detection. In *7th IEEE International Conference on Computer Vision, Frame Rate Workshop (ICCV '99), pp. 1-19* (1999).

[46] HU, J.-S., AND SU, T.-M. Robust background subtraction with shadow and highlight removal for indoor surveillance. *EURASIP J. Appl. Signal Process. 2007*, 1 (2007), 108–108.

[47] HUNTER, E. *Visual estimation of articulated motion using the expectation-constrained maximization algorithm*. PhD thesis, University of California, San Diego, 1999.

[48] JAY, C., GLENCROSS, M., AND HUBBOLD, R. Modeling the effects of delayed haptic and visual feedback in a collaborative virtual environment. *ACM Trans. Comput.-Hum. Interact. 14*, 2 (2007), 8.

[49] JUNG, S., AND WOHN, K. Tracking and motion estimation of the articulated object: a hierarchical kalman filter approach. *Real-Time Imaging Vol. 3, No. 6* (1997), page(s) 415 – 432.

[50] KAEWTRAKULPONG, P., AND BOWDEN, R. An improved adaptive background mixture model for realtime tracking with shadow detection. Kluwer Academic Publishers, 2001.

[51] KAKADIARIS, I., AND METAXAS, D. 3d human body model acquisition from multiple views. In *ICCV* (1995).

[52] KAKADIARIS, I. A., AND METAXAS, D. N. Vision-based animation of digital humans. In *Computer Animation 1998, pages 114* (1998).

[53] KANADE, COLLINS, LIPTON, BURT, AND WIXSON. Advances in cooperative multı-sensor video surveillance. In *DARPA Image Understanding Workshop (IUW), Monterey, CA, pp. 3-24.* (1998).

[54] KIM, S.-E., LEE, R.-H., PARK, C.-J., AND LEE, I.-H. Mimic: real-time marker-free motion capture system to create an agent in the virtual space. *Computers in Education, 2002. Proceedings. International Conference on* (Dec. 2002), 48–49 vol.1.

[55] KURZAK, J., BUTTARI, A., LUSZCZEK, P., AND DONGARRA, J. The playstation 3 for high-performance scientific computing. *Computing in Science & Engineering 10*, 3 (May-June 2008), 84–87.

[56] LAURENTINI, A. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell. 16*, 2 (1994), 150–162.

[57] LAURENTINI, A. How far 3d shapes can be understood from 2d silhouettes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 17*, 2 (Feb 1995), 188–195.

[58] LAURENTINI, A. How many 2d silhouettes does it take to reconstruct a 3d object? *Computer Vision and Image Understanding 67* (July 1997), 81–87.

[59] LAURENTINI, A. The visual hull of curved objects. In *ICCV99, Corfu* (1999).

[60] LI, B., PAN, H., AND SEZAN, I. A general framework for sports video summarization with its application to soccer. In *2003 IEEE International Conference on Accoustics, Speech, and Signal Processing, Apr 6-10 2003* (2003), vol. 3, Institute of Electrical and Electronics Engineers Inc, pp. 169–172. Source type:Electronic.

[61] LI, B., AND SEZAN, M. I. Event detection and summarization in sports video. In *CBAIVL '01: Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'01)* (Washington, DC, USA, 2001), IEEE Computer Society, p. 132.

[62] LI, L., HUANG, W., GU, I. Y. H., AND TIAN, Q. Foreground object detection from videos containing complex background. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia* (New York, NY, USA, 2003), ACM, pp. 2–10.

[63] LORENSEN, W. E., AND CLINE, H. E. Marching cubes: A high resolution 3d surface construction algorithm. vol. Vol. 21, pp. pages 163–169.

[64] METAXAS, D., AND TERZOPOULOS, D. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Trans. Pattern Anal. Mach. Intell. 15*, 6 (1993), 580–591.

[65] MIKIC, I. *Human body model acquisition and tracking using multi-camera voxel data*. PhD thesis, 2002. Chair-Mohan Trivedi.

[66] MIKIC, I., TRIVEDI, M., HUNTER, E., AND COSMAN, P. Articulated body posture estimation from multi-camera voxel data. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on 1* (2001), I–455–I–460 vol.1.

[67] MOESLUND, T. B. *Interacting with a Virtual World through Motion Capture*. Virtual Interaction: Interaction in Virtual Inhabited 3D Worlds, Chapter 11 published by Springer (ISBN: 1852333316)., 2000.

[68] MOESLUND, T. B., AND GRANUM, E. A survey of computer vision-based human motion capture. *Comput. Vis. Image Underst. 81*, 3 (2001), 231–268.

[69] MOESLUND, T. B., HILTON, A., AND KRÜGER, V. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst. 104*, 2 (2006), 90–126.

[70] MUENDERMANN, CORAZZA, AND ANDRIACCHI. The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications. *Journal of NeuroEngineering and Rehabilitation, 3(1)* (2006).

[71] MUNDERMANN, L., CORAZZA, S., AND ANDRIACCHI, T. P. Markerless human motion capture through visual hull and articulated icp. In *NIPS 2006, workshop:EHuM: Evaluation of Articulated Human Motion and Pose Estimation* (2006).

[72] MURRAY, R. M., SASTRY, S. S., AND ZEXIANG, L. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1994.

[73] MUYBRIDGE, E. *Animal Locomotion*. University of Pennsylvania, 1887.

[74] MUYBRIDGE, E. *The Human Figure in Motion*. Dover Publications, 1901.

[75] NIEDERMAYER, D. An introduction to bayesian networks and their contemporary applications. `http://www.niedermayer.ca/papers/bayesian/`, 1998.

[76] NORTON, K., AND OLDS, T. *Anthropometrica: A Textbook of Body Measurement for Sports and Health Courses*. UNSW Press, 1996.

[77] O'ROURKE, J., AND BADLER, N. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. Pattern Analysis and Machine Intelligence 2*, 6 (November 1980), 522–536.

[78] PARKER, J. R. Human motion as input and control in kinetic games. In *FuturePlay 2006, London, Ontario, Canada* (2006).

[79] PLANKERS, R., AND FUA, P. Tracking and modeling people in video sequences. *Computer Vision and Image Understanding Vol. 81* (2001), page(s): 285 – 302.

[80] POPPE, R. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding 108* (2007), 4–18.

[81] PORIKLI, F., AND TUZEL, O. Human body tracking by adaptive background models and mean-shift analysis. In *IEEE Intl. Conference on Computer Vision Systems (ICVS), Workshop on PETS* (2003).

[82] PRATI, A., MIKIC, I., CUCCHIARA, R., AND TRIVEDI, M. M. Comparative evaluation of moving shadow detection algorithms. In *CVPR workshop on Empirical Evaluation Methods in Computer Vision, Kauai* (2001).

[83] RAMOSER, H., SCHLOGL, T., BELEZNAI, C., WINTER, M., AND BISCHOF, H. Shape-based detection of humans for video surveillance applications. *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on 3* (Sept. 2003), III–1013–16 vol.2.

[84] RANDER, P., AND SIEGEL, M. A multi-camera method for 3d digitization of dynamic, real-world events. Tech. rep., The Robotics Institute, Carnegie Mellon University, 1997.

[85] REN, L., SHAKHNAROVICH, G., HODGINS, J., PFISTER, H., AND VIOLA, P. Learning silhouette features for control of human motion. *ACM Transactions on Graphics Vol. 24,No. 4* (2005), pp 13031331.

[86] ROSENHAHN, B., AND SOMMER, G. Pose estimation of free-form objects. In *Proceedings of the European Conference on Computer Vision ECCV '04, part 1, T. Pajdla and J. Matas (Eds.)* (2004), pp. Springer–Verlag, Berling Heidelberg, LNCS 3021, pp. 414–427.

[87] SALVADOR, E., CAVALLARO, A., AND EBRAHIMI, T. Shadow identification and classification using invariant color models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 3, pages 15451548* (2001).

[88] SEITZ, S. M., AND DYER, C. R. Photorealistic scene reconstruction by voxel coloring. *Journal of Computer Vision, volume 35, number 2,* (1999).

[89] SETIAWAN, N. A., SEOK-JU, H., JANG-WOON, K., AND CHIL-WOO, L. Gaussian mixture model in improved hls color space for human silhouette extraction. In *ICAT 2006, LNCS 4282, pp. 732741* (2006).

[90] SIDENBLADH, H. A framework for modeling the appearance of 3d articulated figures. In *In Automatic Face and Gesture Recognition, page(s) 368 -375* (2000).

[91] SIGAL, L., AND BLACK, M. J. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Tech. rep., Brown University, 2006.

[92] SLABAUGH, G., CULBERTSON, B., MALZBENDER, T., AND SCHAFER, R. A survey of methods for volumetric scene reconstruction from photographs, 2001.

[93] SMINCHISESCU, C. *Estimation Algorithms for Ambiguous Visual Models–Three-Dimensional Human Modeling and Motion Reconstruction in Monocular Video Sequences*. PhD thesis, Institute National Politechnique de Grenoble (INRIA), July 2002.

[94] SUNDARESAN, A. *Towards Markerless Motion Capture: Model Estimation, Initialization and Tracking*. PhD thesis, University of Maryland, College Park, 2007. won algorithm award 2006.

[95] SUTHERLAND, J. Agile development: Lessons learned from the first scrum. Tech. rep., Cutter Agile Project Management Advisory Service, 2004.

[96] SVOBODA, T., MARTINEC, D., AND PAJDLA, T. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments, pp 407-422, 14(4), August 2005. MIT Press*.

[97] SZELISKI, R. Rapid octree construction from image sequences. *CVGIP: Image Understanding 58* (July 1993), 23–32.

[98] THEOBALT, C., DE AGUIAR, E., MAGNOR, M. A., THEISEL, H., AND SEIDEL, H.-P. Marker-free kinematic skeleton estimation from sequences of volume data. In *VRST '04: Proceedings of the ACM symposium on Virtual reality software and technology* (New York, NY, USA, 2004), ACM, pp. 57–64.

[99] THEOBALT, C., MAGNOR, M., SCHULER, P., AND PETER SEIDEL, H. Combining 2d feature tracking and volume reconstruction for online video-based human motion capture. In *Pacific Graphics 2002* (2002).

[100] TRIVEDI, M., CHENG, S. Y., CHILDERS, E., AND KROTOSKY, S. Occupant posture analysis with stereo and thermal infrared video: algorithms and experimental evaluation. *Vehicular Technology, IEEE Transactions on 53*, 6 (Nov. 2004), 1698–1712.

[101] TRIVEDI, M., HUANG, K., AND MIKIC, I. Dynamic context capture and distributed video arrays for intelligent spaces. *Systems, Man and Cybernetics, Part A, IEEE Transactions on 35*, 1 (Jan. 2005), 145–163.

[102] TSAI, R. Y. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation Vol. RA-3, No. 4* (1987), pages 323–344.

[103] HTTP://AGILEMANIFESTO.ORG/. Manifesto for agile software development.

[104] HTTP://SUBVERSION.TIGRIS.ORG/. Subversion source control software.

[105] HTTP://VISION.CS.BROWN.EDU/HUMANEVA/. Humaneva - synchronized video and motion capture dataset for evaluation of articulated human motion. Web, May 28 2008.

[106] HTTP://WWW.CS.BROWN.EDU/~LS/EHUM2/SUBMIT.HTML. Humanevaii data collected using hardware synchronized system with 4 color video cameras and 12 vicon motion capture cameras.

[107] HTTP://WWW.CS.PRINCETON.EDU/COURSES/ARCHIVE/SPR02/CS496/VXLVIEW/VXLVIEW.EXE. Voxel visualiser.

[108] HTTP://WWW.EXTREMEPROGRAMMING.ORG/. Extreme programming.

[109] HTTP://WWW.EYETOY.COM/INDEX.ASP. Eyetoy - richard marks, manager, special projects, sony entertainment.

[110] HTTP://WWW.INTEL.COM/TECHNOLOGY/COMPUTING/OPENCV/. Opencv - open source computer vision library.

[111] HTTP://WWW.NATIONMASTER.COM/GRAPH/PEO_SIZ_OF_HOU-PEOPLE-SIZE-OF-HOUSES.

[112] HTTP://WWW.OGRE3D.ORG/. Object-oriented graphics rendering engine.

[113] HTTP://WWW.XVID.ORG/DOWNLOADS.HTML. Codec used with humaneva video files.

[114] VASILESCU, M. A. O. Human motion signatures: Analysis, synthesis, recognition. In *Proceedings of International Conference on Pattern Recognition (ICPR 2002) Quebec City, Canada* (2002), vol. 3, pp. 456–460.

[115] VIOLA, P., AND JONES, M. Rapid object detection using a boosted cascade of simple features. pp. 511–518.

[116] VLASIC, D., BARAN, I., MATUSIK, W., AND POPOVIC, J. Articulated mesh animation from multi-view silhouettes. In *SIGGRAPH 2008* (2008).

[117] WAGG, AND NIXON. Automated markerless extraction of walking people using deformable contour models. *Computer Animation and Virtual Worlds 15(3-4)* (2004), pp 399–406.

[118] WANG, L., HU, W., AND TAN, T. Recent developments in human motion analysis. *Pattern Recognition 36* (Mar. 2003), 585–601.

[119] WREN, C. R. *Understanding Expressive Action*. PhD thesis, Massachusetts Institute of Technology, 2000.

[120] YONEMOTO, S., AND TANIGUCHI, R. Human figure control software for real-virtual application. *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on* (July 2004), 858–862.

[121] ZHANG, C., AND CHEN, T. A self-reconfigurable camera array. In *Eurographics Symposium on Rendering* (2004), H. W. Jensen and A. Keller, Eds., ECE, Carnegie Mellon University, Pittsburgh, PA 15213, USA.