Human Motion Reconstruction from Video Sequences with MPEG-4 Compliant Animation Parameters

by

Dan Carsky

Submitted in fulfillment of the academic requirements
for the degree of Master of Science in Engineering
in the School of Electrical, Electronic and Computer Engineering
at the University of KwaZulu-Natal, Durban, South Africa

October 2005

Abstract

The ability to track articulated human motion in video sequences is essential for applications ranging from biometrics, virtual reality, human-computer interfaces and surveillance. The work presented in this thesis focuses on tracking and analysing human motion in terms of MPEG-4 Body Animation Parameters, in the context of a model-based coding scheme. Model-based coding has emerged as a potential technique for very low bit-rate video compression. This study emphasises motion reconstruction rather than photorealistic human body modelling, consequently a 3-D skeleton with 31 degrees-of-freedom was used to model the human body. Compression is achieved by analysing the input images in terms of the known 3-D model and extracting parameters that describe the relative pose of each segment. These parameters are transmitted to the decoder which synthesises the output by transforming the default model into the correct posture.

The problem comprises two main aspects: 3-D human motion capture and pose description. The goal of the 3-D human motion capture component is to generate 3-D locations of key joints on the human body without the use of special markers or sensors placed on the subject. The input sequence is acquired by three synchronised and calibrated CCD cameras. Digital image matching techniques including cross-correlation and least squares matching are used to find spatial correspondences between the multiple views as well as temporal correspondences in subsequent frames with sub-pixel accuracy. The tracking algorithm automates the matching process examining each matching result and adaptively modifying matching parameters. Key points must be manually selected in the first frame, following which the tracking commences without the intervention of the user, employing the recovered 3-D motion of the skeleton model for prediction of future states. Epipo-

lar geometry is exploited to verify spatial correspondences in each frame before the 3-D locations of all joints are computed through triangulation to construct the 3-D skeleton. The pose of the skeleton is described by the MPEG-4 Body Animation Parameters. The subject's motion is reconstructed by applying the animation parameters to a simplified version of the default MPEG-4 skeleton.

The tracking algorithm may be adapted to 2-D tracking in monocular sequences. An example of 2-D tracking of facial expressions demonstrates the flexibility of the algorithm. Further results involving tracking separate body parts demonstrate the advantage of multiple views and the benefit of camera calibration, which simplifies the generation of 3-D trajectories and the estimation of epipolar geometry. The overall system is tested on a walking sequence where full body motion capture is performed and all 31 degrees-of-freedom of the tracked model are extracted. Results show adequate motion reconstruction (i.e. convincing to most human observers), with slight deviations due to lack of knowledge of the volumetric property of the human body.

Preface

The research work presented in this thesis was performed by Dan Carsky, under the supervision of Mr. Bashan Naidoo and Mr. Stephen McDonald, at the University of KwaZulu-Natal's School of Electrical, Electronic and Computer Engineering. This work has been generously sponsored by Thales Advanced Engineering and Armscor. The financial assistance of the Department of Labour (DoL) towards this research is hereby acknowledged.

Parts of this thesis were presented by the author at PRASA 2003, SATNAC 2004 and MICSSA 2003/2005.

The entire dissertation, unless otherwise indicated as a reference, is the student's own original work and has not been submitted, in whole or in part, to any other university for degree purposes.

Signed: 

Name: Dan Carsky

Date: 15/04/06

Acknowledgements

I would like to thank my supervisors Mr. Bashan Naidoo and Mr Stephen McDonald for allowing me to undertake this project and giving me the freedom to follow my research interests. I am extremely grateful particularly for their time and effort spent on proof reading this thesis and my conference papers. My sincere gratitude also extends towards Mr Peter Handley (Thales Advanced Engineering) and Franzette Vorster (ARMSCOR) for their generous financial support, their patience and interest in this work. Apart from providing the necessary hardware, their generosity allowed me to attend the numerous conferences.

I wish to thank my parents Eva and Milan for their constant love and support without which I would not be where I am today. Furthermore I would like to thank my girlfriend Caroline, for her encouragement and help with graphics related issues particularly in my presentations, and my office-mate and good friend Jon, for his technical help as well as the good times had away from work. I also thank Brice for his initial help with the maths, Colin for his help with data capture, Stefan for his LaTeX "customer support", all my other fellow post-graduates and members of staff in the department and friends who made this time greatly enjoyable.

Finally I would like to thank those academics and students from other institutions whom I have contacted during my research, and who took the time to respond, answer my questions, give me advice or send me soft/hard-copies of their papers:
Academiques Universite de Lausanne - Dr. Anthony Guye-Vuilleme
Czech Technical University in Prague - Michal Kraus

Institut National des Telecommunications - Prof. Marius Preda

Lulea University of Technology - Prof. Inge Soderkvist

MPEG - Dr. Leonardo Chiariglione

Swiss Federal Institute of Technology in Zurich - Dr. Manos Baltsavias, Prof. Armin Gruen, Sebestyen Susanne, Prof. Daniel Thalmann, Dr. Frederic Vexo, Mario Gutierrez

Technical University of Delft - Prof. Mathias Lemmens, Axel Smits

University of Calgary - Dr. Naser El-Sheimy, Yubin Xin

University of Cape Town - Keith Forbes, Megan Watson, Prof. Heinz Ruther

University of Leeds - Prof. David Hogg

University of Maine - Dr. Peggy Agouris, Dr. Arie Croitoru

also Point Grey Research for their outstanding customer support.

Publications

The following publications are based on the work reported in this thesis.

D. Carsky, B. Naidoo, and S. McDonald, "Tracking points in 3-D for human motion capture", in *Proceedings of the Military Information and Communications Symposium of South Africa (MICSSA)*, Pretoria, South Africa, November 2003.

D. Carsky, B. Naidoo, and S. McDonald, "Towards multi-camera human motion capture and low bit-rate video", in *Proceedings of the Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, Langebaan, South Africa, November 2003.

D. Carsky, B. Naidoo, and S. McDonald, "From digital image matching to tracking and low bit-rate video", in *Proceedings of the Southern African Telecommunication Networks and Applications Conference (SATNAC)*, Stellenbosch, South Africa, September 2004.

D. Carsky, B. Naidoo, and S. McDonald, "Tracking points in video sequences with an adaptive least squares matching tracker", in *Proceedings of the Military Information and Communications Symposium of South Africa (MICSSA)*, Pretoria, South Africa, July 2005.

Contents

Abstract i

Preface iii

Acknowledgements iv

Publications vi

Contents vii

List of Figures xii

List of Tables xx

List of Acronyms xxi

1 Introduction 1

1.1 Model-Based Coding 2

1.2 Human Body Animation 4

1.3 System Definition 5

1.4 Thesis Outline 9

2 Human Motion Capture 11

2.1 History 12

2.2 Applications 14

2.3 Methods of Human Motion Capture 15

2.4 Commercial Systems 17

2.5 Prior Work 21

2.5.1 2-D Approaches without Explicit Shape Models 23

2.5.2 2-D Approaches with Explicit Shape Models 25

2.5.3 3-D Approaches with Explicit Shape Models 28

2.6 Conclusion 35

3 Fundamental Theory **39**

3.1 Notation 39

3.2 Camera Model 40

3.3 Camera Parameters 41

3.4 The Perspective Camera 43

3.5 Camera Calibration 43

3.6 Epipolar Geometry 46

3.6.1 Epipolar Constraint 46

3.6.2 The Essential Matrix 48

3.6.3 The Fundamental Matrix 50

3.7 3-D Reconstruction 52

3.8 Summary 54

4 Digital Image Matching **56**

4.1 Overview 56

4.2 Grey Value Correlation 58

4.3 Least Squares Matching 61

4.3.1 Mathematical Model 62

4.3.2 Computational Aspects 65

4.3.3 Typical Results 71

4.4 Summary 79

5 Tracking Algorithm **83**

5.1 LSM Adaptation to Tracking 84

5.1.1 Shaping Parameter Tests 85

5.1.2 Convergence 87

5.1.3 Result Analysis 89

5.1.4 Adjustment of Matching Parameters 92

5.1.5 Convergence Radius 93

5.2 2-D Tracking 95

5.3 3-D Tracking 99

5.3.1 Skeleton Model 101

5.3.2 Initialisation 103

5.3.3 Prediction 106

5.3.4 Tracking in Multiple Views 107

5.3.5 Tracking Verification 108

5.3.6 Advantages of Multiple Views 110

5.4 Summary 113

6 Virtual Humanoid Animation with MPEG-4 **115**

6.1 MPEG-4 Overview 116

6.2 MPEG-4 Human Body Animation Tools 118

6.3 MPEG-4 Body Animation Parameters 119

6.4 Pose Parameter Extraction 125

6.5 Summary 130

7 Results and Discussion **131**

7.1 Camera Calibration 133

7.2 Image Matching 135

7.3 3-D Reconstruction 139

7.4 3-D Tracking 143

7.4.1 Sub-Pixel Accuracy for Tracking 143

7.4.2 3-D Tracking Precision 146

7.4.3 3-D Tracking Limitations 149

7.4.4 Human Motion Data 152

7.5 Pose Reconstruction 155

7.6 Complete System 159

7.7 Conclusion 164

8 Conclusions **167**

8.1 Conclusions 167

8.2 Future Work 169

A Homogenous (Projective) Representation of Lines **172**

B Rodrigues' Rotation Formula **174**

C Definition of Skeleton Model Segments **175**

Bibliography **191**

List of Figures

1.1 Structure of a general model-based coding scheme. 3

1.2 MPEG-4 human face and body animation. (customised male model images from Thalmann and Vexo [1]) 5

1.3 Implemented model-based coding scheme. 6

1.4 Comparison of the "region of interest" size, (a) head and shoulders scenario, (b) full body motion analysis scenario. 7

2.1 Muybridge's sequence of photographs called "Annie galloping". (from Wikipedia [2]) 12

2.2 Marey's motion capture suit and captured walking sequence. (from the Australian Centre for the Moving Image [3]) 13

2.3 Polhemus motion capture devices, (a) LIBERTY, (b) FASTRAK, (c) PATRIOT. (from Polhemus [4]) 18

2.4 ReActor2 from Ascension Technology Corporation. (from Ascension Technology Corporation [5]) 18

2.5 (a) InertiaCube3, (b) Animazoo Gypsy4 system. (from Intersense [6] and Animazoo [7]) 19

2.6 (a) Actor with retro-reflective markers, (b) VICON MX40 camera. (from Leeds Met [8] and VICON [9]) 20

2.7 Pedestrian contour representation by B-splines with shape variation. (from Baumberg and Hogg [10], ©1994 IEEE) 24

2.8 (a) Video input, (b) segmented subject, (c) 2-D blob representation. (from Wren *et al.* [11], ©1997 IEEE) 25

2.9 2-D stick figure model fleshed out with ribbons. (from Leung and Yang [12], ©1995 IEEE) 26

2.10 The cardboard person model, human representation by planar patches. (from Ju *et al.* [13]) 27

2.11 Stick figure model. (from Holt *et al.* [14], ©1994 IEEE) 29

2.12 (a) Input video from one view with 2-D blobs overlayed, (b) corresponding dynamic skeleton model. (from Wren and Pentland [15]) 30

2.13 Elliptical cylinder model. (from Hogg [16]) 31

2.14 Female and male 3-D human models using tapered superquadratics. (from Gavrila and Davis [17], ©1995 IEEE) 33

2.15 (a) Visual hull from multiple silhouettes, (b) skeleton fitted to visual hull. (from Theobalt *et al.* [18]) 33

2.16 Layered human body model; skeleton, ellipsoidal metaballs representing flesh, polygonal surface representation of skin, shaded renedering. (from Plankers *et al.* [19]) 35

3.1 Perspective camera model. 40

3.2 Similar triangle relationship under perspective projection (shown only for y-component). 41

3.3 Relationship between reference frames of a visual system. 44

3.4 2 out of 20 calibration images for left, middle and right view cameras. . . . 45

3.5 (a) Calibration pattern region of interest, (b) detected corners. 46

3.6 Epipolar geometry. 47

3.7 Epipolar Constraint: the set of possible matches for the point $\mathbf{p}_l$ is constrained to lie on the associated epipolar line $\mathbf{l}_r$. 47

3.8 Epipolar Pencil. 48

3.9 Coplanarity condition. 49

3.10 Triangulation with non-intersecting rays. 54

4.1 Searching for best match using similarity measures. 57

4.2 An example of the weight distribution of pixels in a 5×5 patch. 59

4.3 Matching points in a stereo pair using normalised cross-correlation. 60

4.4 Using epipolar geometry to reduce the search space from a 2-D to a 1-D problem. 61

4.5 Resampling a transformed image using different interpolation methods, (a) original image, (b) nearest neighbour, (c) bilinear interpolation, (d) bicubic interpolation. 70

4.6 Bilinear interpolation considers the four neighbouring points of the input sampling grid I_0 around each point of the projected output sampling grid I. 71

4.7 LSM using synthetic data, (a) template image with patch, (b) search image with patch, (c) search image and patch deformation after iterations 1 to 5. . 73

4.8 Synthetic data matching results. 75

4.9 Left (template) and right (search) view of a stereo system. The original patch in the search image is shown in black, and the final deformed patch in white. 76

4.10 Real data results. 77

4.11 Visual overview of the least squares matching process. 81

5.1 Information flow of the automated LSM module. 84

5.2 Examples of undeterminable parameters. Dotted lines indicate convergence thresholds of respective updates. 86

5.3 Determination of convergence by setting thresholds for updates (a), (b) and by setting a threshold for change in σ_0 (c). 87

5.4 Pixel error of matching experiment used to generate convergence data for Figure 5.3. 88

5.5 Images demonstrating incorrect convergence, (a) template image with point of interest indicated by "+", (b) search image with initial estimate indicated by "o" and converged result by "+". 89

5.6 Convergence of the updates (a), (b) and the change in standard deviation (c). Thresholds are indicated by dotted lines. 90

5.7 Assessing the quality of result by (a) cross-correlation coefficient R_{ts}, (b) standard deviation σ_0, (c) standard deviations of the shifts σ_x, σ_y. Thresholds are indicated by dotted lines. 91

5.8 (a) Flow diagram of the matching parameter adjustment process, (b) hierarchical structure of parameter modification. 93

5.9 Examples of the convergence radius in two matching experiments. 94

5.10 2-D tracking algorithm flow diagram. 96

5.11 2-D linear prediction. Using prior 2-D velocity to reduce cross-correlation search space. 97

5.12 Prediction error, (a) point on the mouth, (b) wrist, (c) elbow. 98

5.13 Selected frames from 2-D tracking experiments, (a) tracking facial expressions, (b) arm tracking. 99

5.14 3-D tracking algorithm flow diagram. 100

5.15 3-D skeleton model. 103

5.16 Key point locations on the human body. 103

5.17 Initialisation stage flow diagram. 104

5.18 Correspondence verification in a trinocular system with epipolar geometry. 105

5.19 Predicting 2-D locations in multiple images based on the estimated motion of the 3-D model projected into the image planes. 106

5.20 Pixel error of the 2-D velocity-based and 3-D velocity-based prediction. . . 107

5.21 Strategy for point tracking in multiple views. 108

5.22 Verification of tracking results, temporal matching location is indicated by a "×", spatial matching location is indicated by a "+". (a) successful tracking, (b) error detection by epipolar geometry. 110

5.23 Data output of the 2-D and 3-D tracking algorithm for an arm tracking sequence, (a) frames from the input sequence, (b) 2-D output trajectories, (c) 3-D output trajectories. 111

5.24 Smoothing of the z-velocity component. 112

6.1 Example of an MPEG-4 scene. 117

6.2 Example of a scene graph corresponding to Figure 6.1. 117

6.3 FBA scene graph. 118

6.4 (a) Anatomical rotation planes, (b) shoulder rotation in the sagital plane (flexion), (c) shoulder rotation in the coronal plane (abduction). 121

6.5 Suggested joint centre locations of the complete MPEG-4 skeleton showing front view, rotated view and close-up of the hand. 122

6.6 Model-based coding human model, a) front view, b) rotated view. 122

6.7 Rotation normals of the right arm, a) front view, b) side view. 123

6.8 Skeleton model topology. 125

6.9 Pose parameter extraction flow diagram. 126

6.10 BAP extraction for joints with multiple DOF, (a) compound rotation of the right shoulder, (b) recovery of r_shoulder_flexion, (c) recovery of r_shoulder_abduct. Rotation normals are indicated as either going into, or out of the page. . . . 127

6.11 Generation of a synthetic pose of the right arm. On the left is the right arm in the default position, in the middle are the applied pose parameters, on the right is the transformed arm (solid line) as well as the default pose (dotted line). 128

6.12 Steps 1 (a), 2 (b) and 3 (c) of the BAP extraction process for the right arm. A close-up of the current joint and segment under examination is shown on the left, the recovered BAPs are given in the middle, and the effect of the inverse rotation is shown on the right. Default posture of the arm is indicated by dotted lines. 129

7.1 Error propagation in the complete system. 132

7.2 Images used in matching experiments. (a) Template Image, (b) Search Images 1, 2, 3, 4 (rotated), (c) Search Images 5, 6 (rotated, scaled), (d) Search Images 7, 8, 9 (rotated, scaled, zero mean Gaussian noise), (e) Search Images 10 (horizontal shear), 11 (rotated, brightness & contrast modified, zero mean Gaussian noise). 136

7.3 Surface of the cross-correlation coefficient over the search space for (a) good correlation (search image 1), (b) bad correlation (search image 11). 137

7.4 Regions of reconstruction uncertainty depending on angle between rays. . . 139

7.5 Effects of correspondence error on the 3-D reconstruction process. 140

7.6 Selected images from the 3-D reconstruction accuracy test (for middle camera). 142

7.7 (a) Tracking failure due to inaccurate matching, (b) successful tracking with LSM. 145

7.8 (a) Tracking divergence due to inaccurate matching, (b) successful tracking with LSM. 146

7.9 Evaluation of tracking accuracy by tracking a target over known distances. 147

7.10 Test sequences used for tracking accuracy evaluation, (a) motion in the x-direction, (b) motion in the y-direction, (c) motion in the xz-direction. . . 148

7.11 Image patches in a stereo pair violating the assumption of locally planar surfaces. 150

7.12 Blurring effect of fast human motion. 150

7.13 Tracking divergence due to large scale reduction. 151

7.14 3-D tracking of human motion. Only the middle view of the multi-view video sequence is shown, with the corresponding 3-D model fit to the tracked 3-D points. 152

7.15 3-D path of the sacroiliac in the "walking" sequence, (a) rotated view, (b) top view (x-z plane). 154

7.16 Velocity of the HumanoidRoot, (a) x-component, (b) y-component, (c) z-component, (d) smoothed z-component (using the Savitzky-Golay smoothing filter). 155

7.17 Front and rotated views of (a) the tracked 3-D skeleton and (b) the reconstructed model at the decoder. 156

7.18 Difference between tracked model (a) and H-Anim based model (b) caused by lack of volumetric data. 157

7.19 Recovered BAPs (in degrees) of the 6th thoraic vertebrae, (a) vt6_ roll (rotation in the coronal plane), (b) vt6_torsion (rotation about the vertical body axis), (c) vt6_tilt (rotation in the sagital plane). 159

7.20 Frame 1 of the "walking" sequence. 160

7.21 Frame 35 of the "walking" sequence. 161

7.22 Frame 115 of the "walking" sequence. 162

7.23 Frame 185 of the "walking" sequence. 163

7.24 Frame 205 of the "walking" sequence. 164

A.1 (a) Two points define a line, (b) two lines define a point. 173

C.1 Definition of segment names. 175

List of Tables

2.1 Cost comparison of motion capture systems 20

3.1 Degrees of 3-D reconstruction vs *a priori* knowledge of camera parameters . 53

4.1 Comparison of ground truth and matching results (after 5 iterations) 74

4.2 Indicators of quality of real data matching result 79

6.1 BAPs associated with the joints of the skeleton model 124

7.1 Intrinsic camera parameters of the multi-view setup 134

7.2 Extrinsic camera parameters of the multi-view setup 134

7.3 Comparison of least squares matching and normalised cross-correlation matching results 138

7.4 Results of the 3-D reconstruction experiment 141

7.5 3-D tracking error 147

7.6 Statistics of segment length recovery in the "walking" sequence (segment names are defined in Appendix C) 153

7.7 Extracted BAPs describing the subject's pose in one frame 158

List of Acronyms

1-D	One-Dimensional
2-D	Two-dimensional
3-D	Three-dimensional
AC	Alternate Current
ASM	Active Shape Model
BAP	Body Animation Parameter
BAPU	Body Animation Parameter Unit
BAT	Body Animation Table
BBA	Bone Based Animation
BDP	Body Definition Parameter
BIFS	Binary Format for Scenes
BVH	Bio Vision Hierarchical data
CCD	Charge Coupled Device
COP	Centre of Projection
DC	Direct Current
DCT	Discrete Cosine Transform
DOF	Degree of Freedom
DSP	Digital Signal Processor
DTM	Digital Terrain Model
DTW	Dynamic Time Warping
FAP	Face Animation Parameter
FBA	Face and Body Animation

FDP	Face Definition Parameter
fps	Frames per Second
H-Anim	Humanoid Animation Group
HCI	Human-Computer Interaction
HMC	Human Motion Capture
HMM	Hidden Markov Model
Hz	Hertz
JTC	Joint Technical Committee
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
ISO	International Standards Organisation
IR	Infra Red
kbps	Kilobits per Second
LED	Light Emitting Diode
MIT	Massachusetts Institute of Technology
LSM	Least Squares Matching
MLD	Moving Light Displays
MPEG	Motion Pictures Expert Group
NCC	Normalised Cross-Correlation
RMS	Root Mean Square
SNHC	Synthetic and Natural Hybrid Coding
SO(3)	Special Orthogonal Group in Three Space

Chapter 1

Introduction

Human vision is a complex system that enables us to perceive and act in an equally complex environment. Computer vision aims to provide artificial systems with human-level ability to extract information from image data. Computer vision is a relatively young area of research that is closely related to fields such as image processing, pattern recognition and photogrammetry. In today's information age the sharing of visual information constitutes a large portion of traffic in communication systems. Nowhere is this more evident than in the case of the ever evolving internet and the newly introduced 3G mobile cellular networks that provide the consumer market with technology such as video conferencing. The large amount of data associated with images result in storage burdens and heavy bandwidth utilisation when exchanged between users. These problems are further amplified when video sequences are used instead of still images, and/or transmission is performed over wireless networks where bandwidth is further limited. As a result, data compression is absolutely necessary in order for visual information transmission to be practical and cost effective.

The research group at the University of KwaZulu-Natal sponsored by Thales Advanced Engineering and Armscor has been guided by the topic of low bit-rate video. Previous research focused mainly on wavelet compression [20] [21] although a new avenue was examined by Murugast [22], who adopted a more content driven approach when segmenting

moving objects of interest in video sequences. Following the ideology of video compression, the research presented in this thesis explores the relatively new and promising method of *model-based coding*[1] that is capable of achieving very low bit-rates [23]. The method is applied to a specific case of human motion. Under the concept of model-based coding, the research fields of computer vision and computer graphics are brought together. Computer vision deals with the recovery of motion data from the input video sequence, a process referred to as human motion capture (HMC). Computer graphics come in at the output stage of the system which entails rendering of photorealistic human models performing the desired movement. In essence the output is not a real image, rather a synthetically generated scene. The level of realism or believability of the output depends on the quality of the human motion capture system (for human like motion) and human body modelling stage (for realistic humanoid models).

1.1 Model-Based Coding

Conventional coding techniques (predictive coding, transform coding, vector quantisation) belong to information theory based methods which compress each frame by exploiting its stochastic properties. Model-based coding however regards the image as a 2-D snapshot of a 3-D world rather than a group of random signals. Low bit-rates are achieved by exploiting knowledge of the scene content and modelling the 3-D objects in terms of shape and motion. Figure 1.1 depicts a diagram of a general model based coding scheme. Assuming the emphasis is on human motion, the input is a sequence of images of full body movements of a person. The 3-D model encapsulates the geometrical and textural features of the person. Although the appearance (size, texture, etc.) will differ from subject to subject the structure of the human body (hierarchical setup of the joints and bone segments) is common. A default 3-D model may exist at both the encoder and decoder end of the system. Typically, the appearance description will not change throughout the sequence, hence those parameters need to be encoded only once in order to customise the default model (to resemble the subject). Using the 3-D model, the encoder analyses

[1]Also referred to as "content-based coding", "analysis-synthesis coding", "object-based coding", "knowledge-based coding" or "semantic coding"

each frame and collects information about the person's motion and deformation. The extracted parameters collectively describe the 3-D pose of the subject in each frame in a very compact manner. At the decoder, in general terms, synthesis is the reconstruction of the image using the scene parameters. When applied to HMC, the human model is transformed by the extracted pose parameters to represent the person's motion in that frame. Model-based coding can be used for compression of visual information because

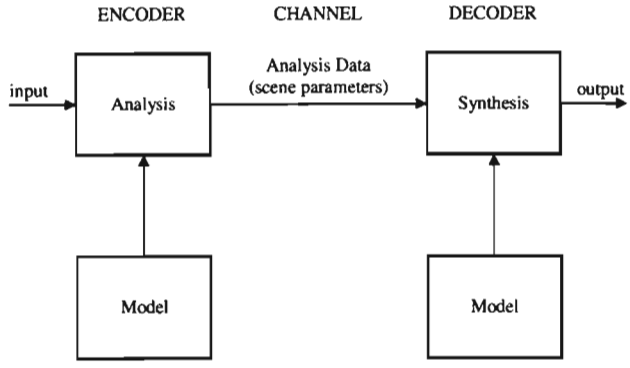


Figure 1.1: Structure of a general model-based coding scheme.

a small number of model parameters are able to quantify the object's temporal changes in a video sequence. The high coding efficiency does however come at a cost, that is, a reduction in generality since the application is restricted to scenes composed of objects known by the encoder/decoder. Previous work contributing towards 3-D model-based coding has concentrated on the scenario of head-and-shoulders [23] [24] since the facial images are important for broad applications, but also because the analysis and synthesis of the human head presents a simpler problem than that of the whole body. The concept was first proposed by Forchheimer *et al.* in 1983 [25] and applied to a model-based video-phone system. Since then, there has been a great interest in the subject [26] [27] [28] [29] [30] [31] [32] [33]. The ratification of the MPEG-4 standard in 1999 stimulated research into model-based coding. MPEG-4 is the first content-based audio-visual coding standard, part of which is specifically devoted to the human face and body. The introduction of this standard has opened the door to new research topics such as low bit-rate video phone for the deaf [34] and very low bit-rate compression of virtual characters [35] [36].

1.2 Human Body Animation

The goal of human body animation is twofold; to produce photorealistic models, as well as realistic motion, of the character. The human body is a complex articulated object and our eyes are especially sensitive to the human figure and the subtle detail of its motion. Realistic animation is a complicated task due to our sensitivity to errors in animated human motion [37] [38]. For this reason, human motion capture has emerged as the preferred method instead of keyframing and physical simulation.

In keyframe animation, the animator specifies keyframes in which the pose of the model is set manually. The intermediate frames are generated by the computer by interpolating between the two key positions. Although this saves a considerable amount of time, it is still a labour intensive technique, particularly for models with a high number of degrees-of-freedom.

Methods based on physical simulation compute movement using simulations of Newtonian physics. This method is typically used for animation of cloth deformations, rigid objects or fluids, however it is deemed impractical for the animation of articulated figures. One reason is the complexity of the physics-based model of such an object, but also, to produce the desired motion the animator would have to be familiar with the human body (i.e. configuration of the muscles and joints, and the energies involved in moving them).

Human motion capture on the other hand records the movements of an actor and maps that data onto the model. More importantly it is able to capture the subject's motion texture. Motion texture is the term (originally suggested by Prof. Ken Perlin of New York University) that describes a person's unique way of movement, i.e. the differences when the same movement is performed by different people, and the variations in an individual's repetitive motions. In this way, life-like and realistic animation is generated, with the virtual character also acquiring some of the actor's personality.

The Face and Body Animation (FBA) subgroup of the Synthetic and Natural Hybrid Coding Group (SNHC) of Motion Pictures Expert Group (MPEG) has addressed the subject of virtual character animation, gesture synthesis and compression/transmission.

The appropriate framework has now been standardised under the MPEG-4 international standard. The standard defines two sets of parameters for the face and body. The Face Definition Parameter (FDP) set and the Body Definition Parameter (BDP) set describe features particular to the subject, while the Face Animation Parameter (FAP) set and Body Animation Parameter (BAP) set are used to animate the virtual humanoid. The decoder uses these parameters to transform the default, H-Anim [39] compliant human model and renders a virtual representation of the subject (Figure 1.2).

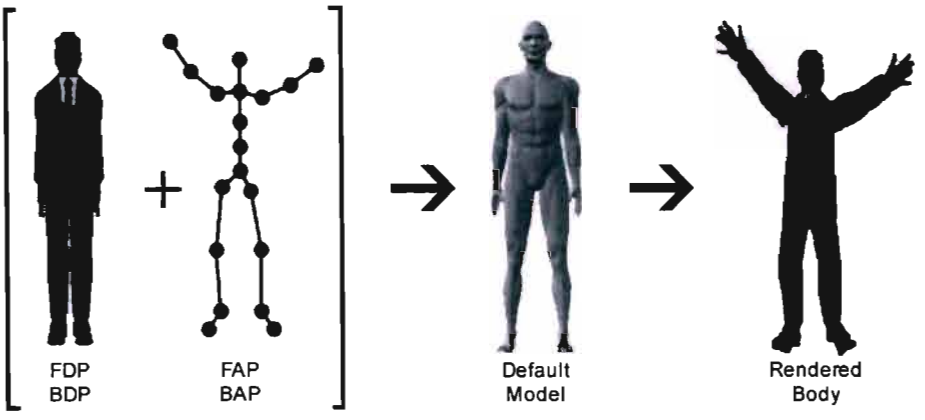


Figure 1.2: MPEG-4 human face and body animation. (customised male model images from Thalmann and Vexo [1])

1.3 System Definition

The scope of the implemented model-based coding scheme is depicted in Figure 1.3. The aim is to reconstruct general human body motion, hence the analysis of the hands and head is not dealt with. The focus is placed on the computer vision aspect of the problem, consequently modelling of the human body is performed only to such a degree as to adequately represent the corresponding movement. Human motion capture is used to estimate the subject's pose in each frame. Moeslund and Granum [40] define pose estimation as *"the process of identifying how a human body and/or individual limbs are configured in a given scene"*. Once the configuration of the individual limbs is known, the pose can be described

by a relatively small number of parameters. The pose parameters employed in this work have been defined according to the MPEG-4 standard, i.e. the pose is defined by the Body Animation Parameter set. The process of rendering human animation needs to tackle two aspects; motion of the hierarchical skeleton and deformation of the body [41]. Only the motion of the underlying skeleton is required for the animation of a model [19]. Thus, the human motion reconstruction from a video sequence is performed by means of a simplified skeleton model with 31 degrees-of-freedom (DOF) and 18 joints as defined in [42]. Since accurate modelling of the body is not carried out (i.e. modelling of the flesh and skin, customising the model to the subject's dimension/texture or animating the deformation of the body), only the animation parameters (BAPs) need to be extracted and passed onto the decoder (i.e. BDPs are not used). Automatic interpretation of human movements is

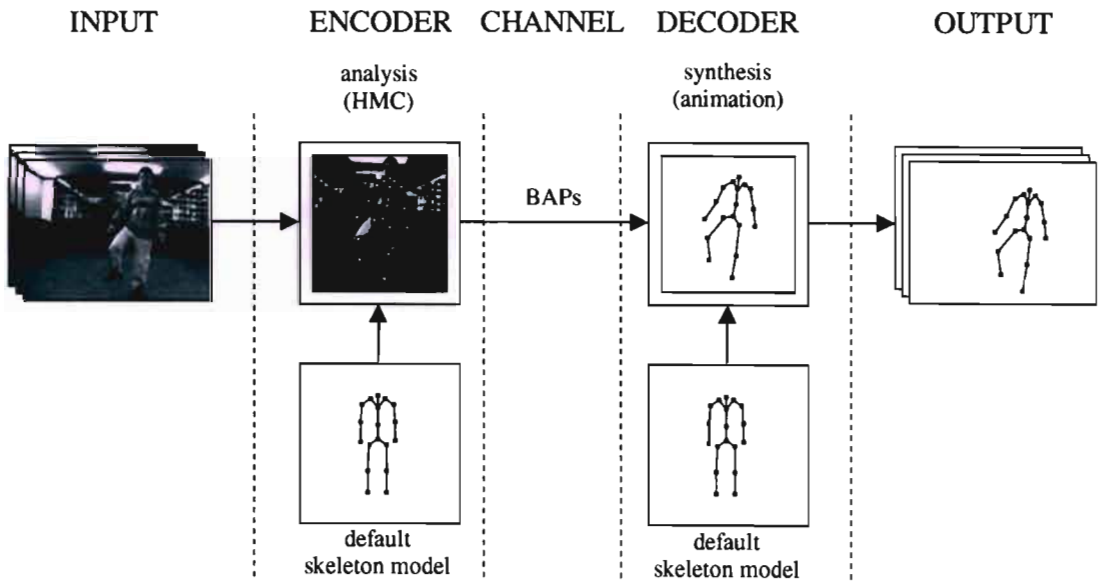
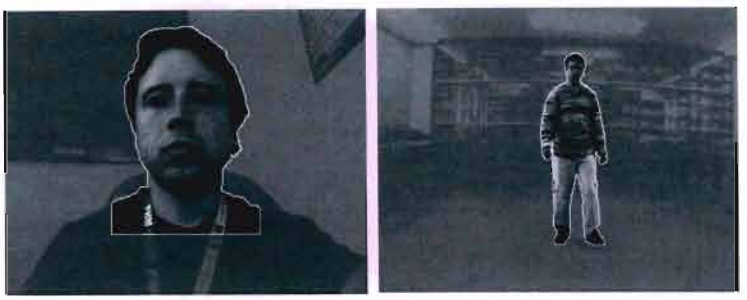


Figure 1.3: Implemented model-based coding scheme.

one of computer vision's most challenging tasks [43]. The head and shoulders scenario has enjoyed more attention from researchers partly due to certain characteristics that make the problem simpler when compared to the analysis of the whole body. The first difference is in the input data. Problems involving motion of the head typically involve the subject positioned close to the camera(s) with the head occupying a large portion of the overall image. This results in better quality data and a larger "region of interest", i.e. signal

belonging to the object under analysis (the head). Figure 1.4 shows the difference in size of the "region of interest" in case of analysis of the head and shoulders, and the full body. Depth estimation is the most challenging task of 3-D reconstruction. In contrast to full body motion, the head undergoes small depth displacements throughout the sequence, offering potentially more accurate depth estimates. However the greatest difference between



(a) (b)

Figure 1.4: Comparison of the "region of interest" size, (a) head and shoulders scenario, (b) full body motion analysis scenario.

the two scenarios is that the global motion of the head is rigid and little occlusion occurs during the sequence. The human body is an articulated object consisting of a number of segments linked by joints. During unconstrained motion, various body parts occlude each other making the tracking process very complex.

Thus far, researchers have failed to produce a full-body tracker general enough to handle realistic real-world applications [44]. Commercial optical systems employ sophisticated and expensive hardware in controlled environments. Occlusion poses the greatest challenge, which is partially overcome by deploying a large number of cameras distributed around the workspace. Retro-reflecting markers are placed on the subject to further simplify the tracking problem. The constantly increasing computational power of off-the-shelf hardware is allowing researchers to focus on marker-free vision-based motion capture. These systems commonly simplify the problem by constraints and assumptions which are discussed in more detail in Chapter 2.

This human motion capture system must provide 3-D locations of key points on the human body from which the pose can be inferred. The problem was tackled by selecting the key

points at the locations of articulation, i.e. joint locations, and tracking them throughout the video sequence using image matching techniques. The whole process can be broken down into the following components:

- Data acquisition
- Image matching
- Tracking
- 3-D reconstruction

This system is non-invasive, and no markers are placed on the subject. In addition, there are no restrictions on the background (static, monochrome etc.), and no major restrictions on the clothing worn by the subject (tighter fitting clothes with good texture produce more accurate and robust motion data).

In this research, the input data is acquired by three synchronised and calibrated CCD Dragonfly cameras from Point Grey Research [45]. These cameras are sensitive to visible light and orientated in a fixed, centralised configuration. The use of multiple cameras aids the tracking process and simplifies the depth estimation problem. The cameras themselves produce colour images at a resolution of 640x480 pixels and frame rate of 30 fps, however only the grey level intensities are utilised in the tracking process. These hardware specifications fall well below those used in the commercial systems and result in lower accuracy. It is however sufficient to capture visually accurate enough data to express a subject's motion.

Image matching is a process that finds correspondences in two images. The tracking algorithm automates the matching process so that following initialisation, no user intervention is required to track selected points throughout the video sequence. The tracking algorithm can be applied to both 2-D (monocular sequence) and 3-D (multi-view sequence) tracking. In the latter case, the additional available information is used to verify tracking results and reconstruct a 3-D model of the subject. The tracking strategy is similar to that of D'Apuzzo [46] who uses least squares matching for multi-view human motion capture.

The reconstructed model is a 3-D stick figure (skeleton) consisting of 18 joints and 17 segments. Although the structure of the model is the same as the one animated at the decoder, it must be noted that this is not always the case, as the two models serve different functions. The model animated at the decoder is used purely for display purposes which present the output of the system. Models in human motion capture are used to aid the tracking process, which in this case means the prediction of future states. The reason for choosing the same model for human motion capture and animation is twofold. Firstly, accurate human body modelling was not dealt with in this work and the skeleton model is the simplest way to adequately represent human motion. Secondly, the reconstructed motion is evaluated by visual inspection (due to the lack of ground truth data) and the use of the same model for both components allows for an easier assessment of the system's performance.

In order to achieve the results presented at the end of this thesis, several constraints/limitations were placed on the system. An initialisation step is required before tracking commences. The user must manually select the 2-D locations of the required joints in the first frame of the one view before automatic tracking in all three views can begin. The selected joints must be visible throughout the sequence, i.e. the tracking algorithm does not handle occlusion of the chosen features. Although this is quite a restrictive constraint, it is not uncommon. Lastly, the process is performed offline, and is not optimised for real-time applications.

1.4 Thesis Outline

The topic of HMC is discussed in Chapter 2. A brief history of the subject is followed by the introduction of the various methods of generating motion data. Several applications of HMC are mentioned, and a number of commercially available systems are examined. The second part of the chapter surveys past research into computer vision based HMC, which is dealt with in three categories.

Chapters 3, 4 and 5 pertain to the computer vision based HMC system implemented as

part of the presented research, which constitutes the analysis part of the model-based coding scheme.

The system is required to generate 3-D locations of the various joints on the human body from the input image data. The theory which maps the pixel locations of points to their actual locations in 3-D space is introduced in Chapter 3. This chapter also explains the various camera parameters used in the camera model, as well as their recovery through camera calibration. A widely used property of stereo view systems, the epipolar constraint is also discussed. The chapter concludes with the method of triangulating the 3-D location of a point from its 2-D correspondences in stereo image pairs.

The HMC system relies on image matching to track desired joint locations. Chapter 4 examines in greater depth two area-based matching techniques that have been implemented, and presents typical matching outputs of each method.

The actual tracking algorithm is discussed in Chapter 5. In order to automatically track points, the algorithm must adaptively tune the various matching parameters and be able to assess the quality of matching results for accurate and robust tracking. The tracking algorithm may be applied to both 2-D and 3-D tracking, and the prediction schemes under both conditions are put forward. In the case of 3-D tracking, additional information is available which is used in verification of tracked data.

Chapter 6 focuses on the MPEG-4 standard and the tools it provides for the efficient modelling and animation of the human body. This chapter identifies the appropriate pose parameters and their extraction process, as well as the model they animate to represent the subject's pose. The pose parameters form the analysis data transmitted by the encoder.

Chapter 7 presents the results and discussion of the accuracy associated with each step leading to the desired output. Conclusions are drawn in Chapter 8 and directions for future work are proposed.

Chapter 2

Human Motion Capture

This chapter begins by defining what is meant by the term human motion capture. A brief history of the topic examines its evolution over almost 120 years. An explanation of the various methods of capturing human motion follows, numerous applications for the acquired data are given and an analysis of commercially available systems provides an insight into the current state of the technology. Finally, previous research into computer vision based methods of HMC is surveyed to provide grounds for the choice of approach as adopted in this research.

The term human motion capture is generally only associated with body analysis, but in fact also covers topics related to face and gesture analysis. Under the face analysis subdivision, one can further branch out into applications concerned with face detection, tracking, face recognition, expression recognition and modelling. Gesture recognition is concerned with extracting meaning out of the motion of hands and arms. Currently the main drive behind this area of research is the development of systems that are able to understand and generate sign language. Body analysis studies the larger body movements. It looks at how people move around and what they do. Depending on the application, the person may be considered as a single blob (surveillance systems) or as a complex articulated object (gait analysis, virtual reality, model-based coding).

2.1 History

Human motion capture starts with Eadweard Muybridge and his famous experiments entitled Animal Locomotion [47] in 1887. Muybridge was initially commissioned by Gov. L. Stanford to demonstrate that, indeed, galloping horses did lift all four feet off the ground at the same time. He proved this theory by using a bank of cameras to capture a series of still images in quick succession (Figure 2.1).

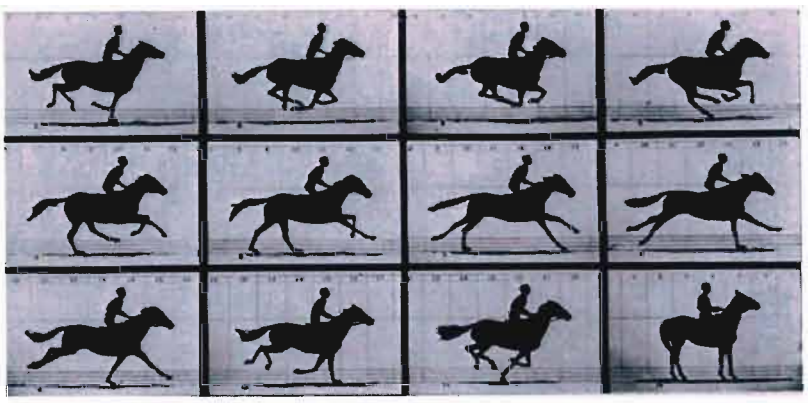


Figure 2.1: Muybridge's sequence of photographs called "Annie galloping". (from Wikipedia [2])

At the same time Muybridge was developing the sequence photography to study the movement of both animals and humans, a French physiologist, Etienne-Jules Marey was searching for new ways of studying movement through photography. To analyse human movement, he took multiple images on one plate, so all the movement was recorded on one print (Figure 2.2). The subject wore a black suit with metal strips or white lines which under correct exposure became the only visible features.

In 1915, Max Fleischer used live action film as the basis for drawing animation by inventing the method called rotoscoping. This method was used by many to get convincing motion for the human characters, including Disney studios in Snow White. In the 1970's, psychologists extensively studied motion perception using Johansson's moving light displays (MLD) technique [48]. This technique involves filming moving subjects with reflective pads or lights attached to their joints. When performed in a dark environment, only the

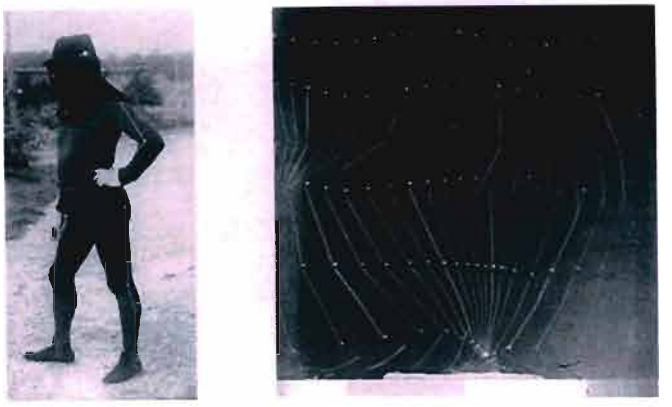


Figure 2.2: Marey's motion capture suit and captured walking sequence. (from the Australian Centre for the Moving Image [3])

motion of highlighted points is displayed.

The main application that has driven human motion capture is character animation. Although rotoscoping can be thought of as a primitive form of HMC, it was only in the late 1970's, when computers became a feasible option to character animation, that HMC for character animation was born. In the early 1980's, professor Tom Calvert made use of techniques previously developed for biomechanics studies and used potentiometers built into an exoskeleton frame in order to record joint movements. At around the same time, MIT Architecture Machine Group and the New York Institute of Technology Computer Graphics Lab experimented with optical tracking of the human body. The systems were based on the same principle as MLDs, with the edition of special hardware and software that calculated the 3-D position of each marker through time. From the mid 1980's animation systems started using computer puppets and more sophisticated suits/exoskeletons. Some examples of these are "Mike the Talking Head" by deGraf/Wahrman and "Waldo C. Graphic" by Pacific Data Images in 1988. Pacific Data Images went on to develop an upper body suit able to track the motion of the torso, arms and head, and in 1992 SimGraphics used sensors on the chin, lips, cheeks and eyebrows for a face tracking system called "face Waldo". In 1989, Kleiser-Walczak produced "Dozo", a non-real-time optical based computer animation of a dancing actor. This system used multiple cameras and reflective tape to triangulate control points in 3-D space. Acclaim managed to greatly

improve this method, releasing a system able to track many points simultaneously in real time in 1993. Since the mid 1990's, there has been a tremendous increase in human motion capture systems, ranging from mechanical, to magnetic, to acoustic as well as optical methods of tracking motion. Researchers have started looking at the non-invasive techniques of capturing motion, bringing the field a step closer to its ultimate goal of purely passive and unconstrained human motion capture.

2.2 Applications

The information gained from capturing and analysing the way people move has a number of promising applications. These include human-computer interface, smart surveillance, motion analysis, virtual reality and model-based coding.

In today's world, the computer has become a common and essential part of everyday life. Although the past two decades have seen rapid development and the computational power of computers has increased by several orders of magnitude, the computer systems of today are still, as Alex Pentland from MIT Media Lab puts it, *"blind and deaf"* to their users. This has lead to the new research field known as Human-Computer Interaction (HCI). The basic idea behind this research is to develop computers able to communicate on human terms, since many people find it difficult to learn the language of computers. Computers with these interfaces will be able to understand gestures, transforming them into something more than mere screens with attached keyboards.

An important application of HMC can be found in smart surveillance systems. Ability to recognise humans may prevent false alarms triggered by animals or natural influences, whilst face identification can be used for access control and behaviour analysis may identify potential threats before a crime is committed [49].

In motion analysis, accurate 3-D pose is utilised in the medical field, sport, dance choreography and video indexing. Accurate motion analysis helps in diagnosing patients with locomotion difficulties, as well as rehabilitation. Stroke patients require multi-disciplinary assessments and appropriate rehabilitation once released from hospital [50]. Systems such

as those proposed by Eriksson and Mataric [51] aim at bringing the rehabilitation task into the patient's home, making rehabilitation more accessible while at the same time taking the burden off healthcare services. Sportsmen use HMC to identify possible improvements in their performance. An exciting new application based on motion analysis dealing with content-based video indexing has been proposed. Sequences are indexed according to the type of motion within the clip reducing the effort involved in searching for specific events, particularly useful application in sporting footage.

Virtual reality is currently a common application of HMC. With the improving technology and decreasing costs, HMC is becoming a viable option for virtual character animation. Data describing the subject's motion is mapped onto a virtual character, generating a smooth and realistic animation. These animations are widely used for special effects in movies and computer games, but also in interactive virtual worlds and teleconferencing.

Finally, a promising new application of HMC is in model-based coding, the subject of this research. This field is centred around the new MPEG-4 standard which looks at compression of model based media in order to achieve low bit-rate transmission. Although most existing research is focused on the human head, the MPEG-4 standard is still new and an increasing amount of work dealing with the full body motion has begun to emerge.

2.3 Methods of Human Motion Capture

Human motion capture systems are composed of two parts, sensing and processing. There is an inverse relationship between the operational complexities of the two subsystems [40], i.e. high complexity in one allows for corresponding simplicity in the other. Perhaps the first decision to be made when choosing HMC systems is whether to use active or passive sensing. Active sensing involves sensors placed on the subject capable of describing their relative orientation with varying degrees-of-freedom. In some cases, the wearable devices work in conjunction with devices placed in the surroundings, where one transmits a signal, and the other uses that signal to generate data describing its orientation. Although active sensing allows for simpler processing of data, it often requires a well controlled workspace.

Passive sensing works with natural signal sources such as visible light, infra-red etc. As such, wearable devices are not required, unless markers are used to simplify tracking. In the case where markers are used, the method is still considered less intrusive and less movement restricting then active sensing. The main methods of HMC are mechanical, optical, or magnetic and these are discussed next.

Mechanical methods use an exoskeleton with built-in sensors (potentiometers, optical encoders etc.) that measure the various angles of articulation. The exoskeleton is worn by the subject, and may in some cases restrict the range of motion. Although this is a relatively simple, robust and accurate method, it is very intrusive.

Optical methods do not require an exoskeleton and as such enjoy a less intrusive and unhindered motion. However, it is common for the motion to be captured under controlled conditions. The subject often wears tight fitting clothes matching the uniform background colour, with reflective markers placed in specific locations on the body which strongly contrast the rest of the scene. Tracking is performed using a number of cameras distributed around the workspace. In most cases, high resolution and high speed cameras are used providing high data rates, an important factor in applications such as biomechanics. Disadvantages of these systems included intensive post-processing, occlusion and capturing multiple actors. Because of the computational requirements, many systems are unable to produce data in real-time. Occlusion is dealt with by employing a large number of cameras to ensure that all markers are visible by at least some cameras in each frame. However in practice this does not eliminate the problem completely. When all the information is derived from the images alone without the use of markers or special setups, the method is often referred to as image-based. Although this approach has clear advantages by being completely passive, it suffers from high computational costs and the unsolved problem of occlusion.

Magnetic systems overcome many of the flaws associated with optical systems, but also introduce limitations of their own. In a magnetic system, a known magnetic field is set up in the workspace. The actor wears sensors that detect their location and orientation based on the magnetic field. Data is captured in real-time, and occlusion and multiple subjects

do not pose any problems. On the other hand, this method is sensitive to its surroundings in particular any metal objects. Furthermore, a field of high enough quality for accurate data collection can only be created in a relatively small space, considerably limiting the size of the workspace.

2.4 Commercial Systems

Human movement tracking systems can be classified as inside-in, inside-out, or outside-in systems. Inside-in systems are those that employ sensors and sources that are both on the body. These systems are not limited by workspace, however in general do not provide 3-D world-based information. Inside-out systems make use of sensors on the body that sense external sources (externally generated electromagnetic field etc.). Although this method provides world-based information, it is restricted by the workspace. Outside-in systems are based on an external sensor that senses artificial (markers) and natural (body texture) sources on the body. Although these systems are the least intrusive (or not at all), they are limited by the workspace, and suffer from occlusion. In the following section several examples of commercial systems are given.

Polhemus [4] is the industry leader in the area of six DOF (position (x, y, z) and orientation (azimuth, elevation, roll)) motion tracking using electro-magnetic technology. Polhemus provides three "inside-out" systems, LIBERTY, FASTRAK and PATRIOT (Figure 2.3). It incorporates a DSP in conjunction with AC magnetics for real-time tracking of virtually anything non-metallic. LIBERTY is the fastest scalable electromagnetic tracker available, able to track objects at 240 Hz for all sensors simultaneously at an effective range of about 2 metres. FASTRAK is the most accurate electromagnetic motion tracking system available, tracking at a range of 1.3 to 2 metres and updating measurements at 120 Hz. PATRIOT is the cost effective solution for six DOF, tracking at 60 Hz with a range of about 1.6 metres.

Ascension Technology Corporation [5] provides two products, a magnetic system MotionStar and an optical system ReActor2 (Figure 2.4). MotionStar uses DC magnetic track-

(a) (b) (c)

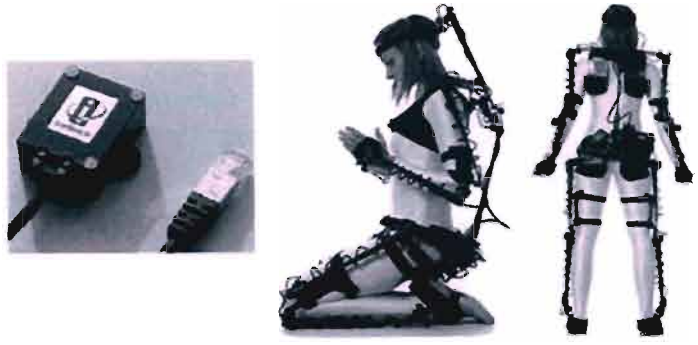
Figure 2.3: Polhemus motion capture devices, (a) LIBERTY, (b) FASTRAK, (c) PATRIOT. (from Polhemus [4])

ing, which is considerably less susceptible to metallic distortion than AC electromagnetic tracking. It generates real-time 6 DOF data of up to six performers at 120 Hz. ReActor2 captures the movement of an untethered subject fitted with infrared markers, moving in a capture area bordered by modular bars. Up to 544 sensors embedded in the frame and over 800 active LEDs provide complete tracking coverage. The Instant Marker Recognition instantly reacquires blocked markers virtually eliminating occlusion drop outs and ensuring clean data.

Figure 2.4: ReActor2 from Ascension Technology Corporation. (from Ascension Technology Corporation [5])

Animazoo [7] has two main "inside-in" systems for full body motion capture, the GypsyGyro-18 and Gypsy4 (Figure 2.5(b)). The GypsyGyro-18 is based on 18 inertial sensors, attached to a lycra suit. These InertiaCube3 sensors (Figure 2.5(a)) provide 3 DOF (azimuth, elevation, roll) measurements at 120 Hz. The sensors do not require an external source, hence the workspace is limited only by the wireless link. The generated data in each measurement represent the change in the actor's BVH (Bio Vision hierarchical data) form. BVH is

a commonly used motion capture file format, explained in more detailed by Meredith and Maddock [52]. Gypsy4 makes use of an exoskeleton that needs to be worn by the actor. The 17 joint lightweight exoskeleton has 42 potentiometers and 1 gyro sensor built into it, all running at 120 Hz. The output data format is the same as for the GypsyGyro-18, with somewhat lower accuracy and a shorter operational range.



(a) (b)

Figure 2.5: (a) InertiaCube3, (b) Animazoo Gypsy4 system. (from Intersense [6] and Animazoo [7])

VICON [9] is the global leader in 3-D optical motion capture and analysis. The success of VICON "outside-in" systems is predominantly due to their high speed, high resolution cameras, designed, developed and built specifically for motion tracking. The Mx range of cameras supports resolutions from 640x480 to 2352x1728, with maximum resolution frame rates ranging from 166 to 484 fps. By reducing the resolution, some of the models are able to achieve frame rates of 10 000 fps. The cameras also serve another purpose, illuminating the retro-reflective markers on the actor with high power strobe LEDs, using either infra-red, near infra-red or visible red light. Together with the VICON tracker, targets are tracked, their 3-D locations are computed and 6 DOF information is outputted in real-time.

The above mentioned systems are but a few examples of the commercially available hardware and software for motion capture. They demonstrate the diversity of approaches to this problem, and although the short reviews give an idea of each system's capability, they

(a) (b)

Figure 2.6: (a) Actor with retro-reflective markers, (b) VICON MX40 camera. (from Leeds Met [8] and VICON [9])

by no means provide all information required to evaluate the individual performance. The reader is thus referred to Mulder [53] [54] for a more in depth review of human motion tracking technology and resources, including further criteria such as accuracy, resolution and lag which are used in measuring the performance of these systems [55].

Table 2.1: Cost comparison of motion capture systems

System	Cost
LIBERTY (16 sensors)	$31,595
FASTRAK (4 sensors)	$8,600
PATRIOT (2 sensors)	$3,200
MotionStar Wired (18 sensors)	$57,600
MotionStar Wireless (18 sensors)	$82,635
GypsyGyro18	$80,000
Gypsy4	$20,000
ReActor2	$100,000
Vicon (6 cameras)	$87,770
Vicon (18 cameras)	$206,630
D. Carsky System (3 cameras)	$2,400

The prices of the discussed systems are shown in Table 2.1. It is evident that motion tracking is not a cheap affair. The tracking algorithm presented in this work tracks 18 joints. It would cost in excess of $30,000 had a commercial product been used. The optical systems in particular are the most expensive, starting at $87,770 and reaching $206,630. This is an indication of the high cost of accuracy and robustness of an optical human motion capture system. In comparison, the 3 camera setup used in this research cost only $2,400 and required no markers or a special workspace.

2.5 Prior Work

This section surveys the previous research into computer vision based human motion capture. To be able to compare the different approaches they need to be grouped into classes having the same characteristics. Various classes that define a particular system have been identified in [40] [49].

- Tracking dimensionality (2-D vs 3-D)
- Model based vs Non-model based
- Sensor modality (visible light, infra-red, range)
- Sensor multiplicity (monocular vs stereo)
- Sensor placement (distributed vs centralised)
- Sensor mobility (stationery vs moving)
- Tracking vs recognition
- Pose estimation vs tracking
- Pose estimation vs recognition
- Application
- Single person vs multiple person
- Number of tracked limbs

However, these classes are not both detailed and broad enough to effectively categorise previous work. As a result, researchers have introduced their own taxonomies into their surveys.

Cedras and Shah [56] overviews motion extraction methods prior to 1995. The methods are classified as either optical flow or motion correspondence. They define three ways of viewing the human motion tracking and recognition problem, as action recognition, recognition of the individual body parts, and body configuration estimation. Aggarwal and Cai [57] describe work prior to 1998. They follow the taxonomy of Cedras and Shah [56], although the three classes are given different labels. Furthermore, each class is divided into subclasses, separating the approaches into model-based/non-model-based methods, single/multiple camera systems and state-space/template matching methods respectively. Gavrila [49] adopted a different taxonomy to [56] and [57]. His survey looks at work prior to 1998, with the main focus on the dimensionality of tracking. Approaches are divided into three groups, 2-D approaches with and without explicit shape models, and 3-D approaches. Recognition is dealt with within the scope of each group. Although different taxonomies are applied to [49] and [56] [57], there are similarities between the classes and essentially same papers are reviewed. Moeslund and Granum [40] take a completely different approach when reviewing papers prior 2001. They use the stages that need to occur in order to solve the general problem of motion capture, rather than the approach or technique used. The structure consists of initialisation, tracking, pose estimation and recognition. Zhou and Hu [58] review the progress in human movement tracking prior to 2003, with a special focus on patient rehabilitation. For vision based tracking systems they follow the taxonomy of Gavrila [49].

In the following review the taxonomy of Gavrila [49] was adopted, with prior work being discussed under one of the three headings, 2-D approaches without explicit shape models, 2-D approaches with explicit shape models, and 3-D approaches. Separating prior work into these categories clearly distinguishes methods which are applicable to the presented application and which are not.

2.5.1 2-D Approaches without Explicit Shape Models

2-D motion tracking is only concerned with the human movement in the image plane. Approaches without explicit shape models bypass the pose recovery step, and describe human movement in terms of simple, low-level, 2-D image features such as points, grids, contours or trajectories. The human body tracking is normally achieved by determining the region of interest through background subtraction, skin colour detection or independent motion.

Cootes *et al.* [59] use statistical models of shape variation. These Active Shape Models (ASM) are computed from a set of training images, where example shapes are described in terms of known feature locations. Applying limits to the parameters of the model enforces global shape constraints, hence the deformable models maintain the essential characteristics of the class of objects they represent while being able to deform to fit a range of examples. The hand model in [59] was used to detect and track the contours of a hand. The compact parameterised model achieves apart from efficiency, also a degree of generalisation over the training set. The drawbacks of this approach include occlusion as features must be present at all times. Furthermore, a good initial estimate is required in order for this method to converge properly, and its dependency on edge points means that the input images must contain good contrast between the object and background. Baumberg and Hogg [60] use ASMs to track the contour of pedestrians in real-time. The contour (silhouette) is represented by a closed uniform B-spline (Figure 2.7). The subject is extracted from the image by background subtraction and similar to [10], a Kalman filter is utilised to achieve spatio-temporal control. The tracker performs reliably only for poses and views sufficiently represented by the training set.

Freeman *et al.* [61] developed a vision-based user input for computer games. A special CCD chip with onboard processing was constructed in order to implement two algorithms that respond to the user's hand or body. One algorithm used x-y image moments to calculate an equivalent rectangle, with same first and second order moments as the image, and the other used orientation histograms to select the body pose from a menu of templates. Both of these algorithms respond to the user in real-time and serve as inputs to computer

Figure 2.7: Pedestrian contour representation by B-splines with shape variation. (from Baumberg and Hogg [10], ©1994 IEEE)

games. Quek [62] uses shape and motion features alternatively to interpret hand-gestures. A moving edge detector that accentuates moving edges, suppresses stationery ones is employed which enforces directional variance, spatial cohesion, directional cohesion and path cohesion constraints within the dynamic image analysis algorithm. The system is simplified by considering only single-hand gestures directed at the machine vision's system. Although recognition results are not given, plots of typical outputs of the system which validate the computational approach are presented.

Darrel and Pentland [63] build a set of model views for hand gestures using normalised correlation. The system automatically adds views to the model set when the correlation of the tracked object and the existing views falls below a predetermined threshold. Dynamic time warping (DTW) is used to match gestures with stored patterns learned from examples. This approach is sensitive to backgrounds. Polana and Nelson [64] use optical flow and down-sampling for low level recognition of periodic human actions. The resulting system is robust to illumination and contrast changes, because the exploited motion information is invariant to these. 2-D optical flow is also used in the proposed solution by Fablet and Black [65] who aside from modelling the human motion also model the motion of generic scenes (background). Detection and tracking was posed in a principled Bayesian framework, with the performance of the system demonstrated on real data of people under different viewpoints with complex backgrounds.

2.5.2 2-D Approaches with Explicit Shape Models

This section discusses the work that uses explicit *a priori* knowledge of how the human body appears in 2-D to segment, track and label body parts. Self-occlusion is always a factor when dealing with human motion. To simplify the problem, generic postures or constrained movements are generally assumed. Typical models include the stick figure, ribbon model or blobs. These models dictate the type of features used for tracking, whether they are edges or ribbons, blobs, or points.

Wren *et al.* [11] present a region-based approach, where a person is represented by a collection of coloured blobs (Figure 2.8). This real-time person finder system, Pfinder, is referred to in almost every human motion tracking paper. The scene is modelled as a static background and a dynamic foreground, with background pixels modelled using a Gaussian distribution, and the foreground modelled as a number of blobs each sharing statistically similar colour (Y,U,V) and spatial (x,y) properties. The initial background

(a) (b) (c)

Figure 2.8: (a) Video input, (b) segmented subject, (c) 2-D blob representation. (from Wren *et al.* [11], ©1997 IEEE)

model is learned from a sequence of images without a subject, and then used to identify the foreground region once a person enters the scene. Both contour and colour methods are used to create a model of the human body, which consists of separate blobs for the hands, head, feet, shirt and pants. New statistics are calculated for each blob and predicted into the next frame using a Kalman Filter. When a blob can find no data to describe it is deleted from the person model. When the body part reappears, a new blob is created either by the contour process or the colour splitting process. As a result, Pfinder is robust to occlusion, being able to recover lost tracks. Stable enough to support real applications,

this system has been used as a front end module in several applications at the MIT Media Lab [15] [66] [67] [68].

Akita [69] demonstrates an early attempt at segmenting and tracking body parts in common circumstances. He assumed the motion to be known *a priori*, thus the different poses of the person were known beforehand and could be modelled using keyframes. The keyframes were a set of representative stick figure poses which helped when the tracking of body parts failed. The various body parts were tracked in the following order, legs, head, arms and trunk. Due to the simplifications, this approach works only in special situations. Leung and Yang [12] describe a system for labelling the human body. The system consists of segmenting, tracking and labelling of body parts from a silhouette of a human. Segmentation is done by image subtraction in combination with edges to segment movements as described in [70]. The different regions are described by 2-D ribbons, which are U-shaped edge segments. The 2-D human model consists of five U-shaped ribbons, a torso and number of joints and mid points (Figure 2.9). Structural and shape constraints

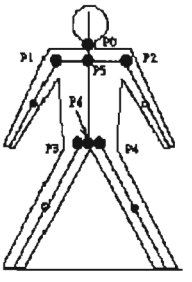


Figure 2.9: 2-D stick figure model fleshed out with ribbons. (from Leung and Yang [12], ©1995 IEEE)

are used to prune the solution space, and model patterns are defined to aid the labelling interpretation process. Long and Yang [71] use the methods described in [12] and [70] to segment the edges of the moving human and find edges which are parallel in order to track logs from frame to frame. A log is an area defined by two parallel lines and a number of attributes such as log-length, log-location, log-area, log-colour and log-orientation. Experiments were conducted to deal with occlusion (appearance, disappearance, merging and splitting of logs) however problems were present. Similar work is done by Kurakake and Nevatia [72] who obtain the joint locations in images of walking humans by establishing

correspondences between extracted ribbons.

The assumption that the motion is performed parallel to the image plane is not uncommon. Both Chang and Huang [73], and Ju *et al.* [13] have used this to simplify their problems. Chang and Huang adopted a ribbon-based motion analysis approach to describe human body movements. Moving ribbons are extracted by processing the difference between current and reference image frames, and analysed to produce motion parameter curves for each joint. Ju *et al.* [13] extend the work of Black and Yacoob [74] to deal with the articulated motion of human limbs. The *"cardboard"* human model (Figure 2.10) consists of 10 rigid planar patches, manually initialised in the first frame. The motion of each patch is defined by 8 parameters which are estimated in each frame by applying an optical flow constraint. A view-based method is used to recognise cyclic motion. Self-occlusion and clothing pose are the main problems associated with this method.

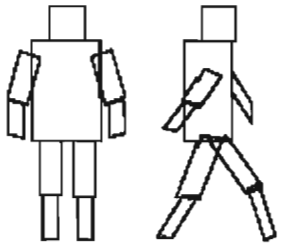


Figure 2.10: The cardboard person model, human representation by planar patches. (from Ju *et al.* [13])

Cai and Aggarwal [75] use point features belonging to the medial axis of the human upper body to track humans across multiple views. Humans are represented by a simplified head-trunk model, and segmented from the background using a double-difference-method described in their previous work [76]. Tracking is performed using the position and velocity of the feature points, as well as the average intensity of the local neighbourhood of the points. Pre-calibration of the cameras allows points on the medial axis to be brought into correspondence using epipolar constraints. Experiments were performed on an indoor sequence involving two subjects using three cameras, with results indicating a robust performance and a potential for real-time implementation.

2.5.3 3-D Approaches with Explicit Shape Models

3-D approaches aim at recovering the 3-D location and pose of a body in every frame of a video sequence. A 3-D model is usually projected into the 2-D images to aid the tracking process and/or pose recovery. Tracking can take advantage of the kinematic and shape properties of the human body to simplify the problem. Techniques used for 3-D pose recovery include inverse kinematics, analysis-by-synthesis, divide and conquer, and constraint propagation. In terms of modelling the human body, there is a trade-off between the model accuracy and its complexity. The 3-D models represent the human skeleton and flesh. Their complexity may range from simple stick figures (skeleton models) [14] [15] [68] [77] [78] [79] [80] [81], to volumetric models made up of shapes such as cylinders [16] [82] [83] [84] [85] [86] [87] [88], spheres [89], ellipsoids [19] [46] [90] [91] or superquadratics [17] [92] [93] [94]. The accuracy of the model determines its ability to predict future movements, including events like self-occlusion or self-collision. O'Rourke and Badler [89] produced some pioneering work in terms of model-based 3-D human motion analysis. Using an elaborate volumetric model for prediction, they found prediction in state space to be more stable than in image space due to the incorporated semantic knowledge. Their tracking framework consisting of four main components, prediction, synthesis, image analysis and state estimation, is followed by most tracking systems today.

The stick figure model is a representation of the human skeletal structure, typically consisting of rigid segments connected by joints with varying DOF. Work done by Chen and Lee [77] [78] aims at recovering 3-D pose using the stick figure model with 14 joints and 17 segments. Under the assumption that 6 feature points on the face are known, the 3-D position of the neck is calculated. With the neck as starting point and known segment lengths, a partial tree representing all possible poses is built up. The path through the tree is pruned by angle, distance, and collision constraints and the assumption of the subject walking, to arrive at the correct posture. [78] extends the work [77] by taking the temporal considerations into account to obtain smooth motion in the model. However, the drawback of all the joints having to be segmented beforehand still remains. Holt *et al.* [14] use a divide and conquer technique to estimate the 3-D motion of an articulated object from monocular sequences. When the general approach was applied to the analysis of

the human gait, a stick figure model (Figure 2.11) was used to represent the articulated human body. The object is decomposed into simpler parts containing a small number of links. Motion of the simplest part is then estimated and propagated to the remaining parts of the object. Attractive in its simplicity, this approach however does not exploit the fact that different components belong to the same model. Zhao *et al.* [79] find the pose

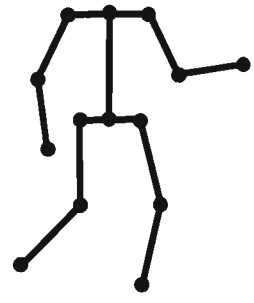


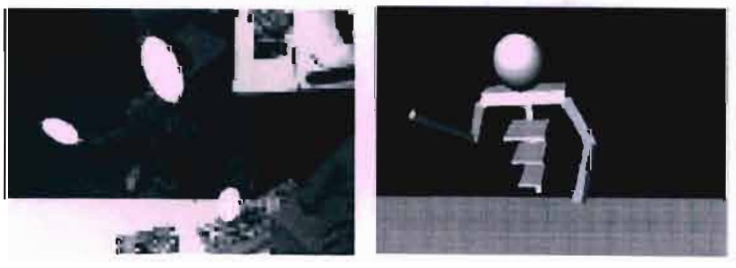
Figure 2.11: Stick figure model. (from Holt *et al.* [14], ©1994 IEEE)

of a subject by minimising an energy function that expresses the deviations of the projected and image features. A stick figure model with encoded biomechanical constraints is used and must be manually initialised in the image. Tracking is performed by means of normalised cross-correlation and least squares matching with linear prediction, although the process is not fully automated and occlusion is not handled. The user is required to reselect occluded points and adjust unnatural reconstructed poses. Barron and Kakadiaris use their early work [95] as an initialisation step for their monocular human motion tracking system [81]. The initialisation step recovers anthropometry[1] (up to a scale) and pose from a single image in which relevant points have been selected manually. The tracking method searches for the best pose in each frame by minimising the discrepancies between the original image and the synthetic image of the projected model. Penalty factors are used in their objective function to guarantee convergence to a global minimum, converting the objective function to a convex function. Poppe *et al.* [80] not only recover the pose but also describe it by MPEG-4 compliant parameters by fitting a 10 joint 16 DOF stick figure into a silhouette from a single view. The silhouette is generated by background subtraction (static background is required), and the length of the person and his/her dis-

[1]Defined in [96] as *"the branch of anthropology concerned with comparative measurements of the human body and its parts"*.

tance to the camera is calculated using *a priori* knowledge of the exact orientation of the camera. Positions of the hands and head are located by skin colour detection, locations of the feet are estimated from the silhouette, and the elbows and knees are found by inverse kinematics. The process assumes the subject is always facing the camera, and no twisting of torso occurs. The complete 3-D skeleton is then projected into the image and thickened by rounded rectangles. The final pose is obtained by a prediction and silhouette matching scheme, so that the various rotations can be recovered.

In [15] [67] [68] Wren and Pentland use the Pfinder algorithm [11] on two images from different views to produce 3-D blobs for the hands and head (Figure 2.12). Using a dynamic skeleton model and kinematic constraints, the 3-D pose of the upper body is estimated. Tracking and segmentation of the blobs is aided by predicting the model in the next frame with a Kalman Filter. Tracking performance is further boosted by including behaviour models into the control loop, modelling behaviour using Hidden Markov Models (HMM). Results demonstrate the real-time, fully-dynamic performance of this multi-view 3-D person tracking system, able to handle full (temporary) occlusion and presence of multiple people.



(a) (b)

Figure 2.12: (a) Input video from one view with 2-D blobs overlayed, (b) corresponding dynamic skeleton model. (from Wren and Pentland [15])

Volumetric models are used to represent the skeletal structure of humans, as well as the flesh surrounding the bones. Hogg [16] and Rohr [84] [87] extended the work by Marr and Nishihara [97] who used a set of fourteen elliptical cylinders to model the human body (Figure 2.13). In both cases, movement is assumed to be parallel to the image plane. Hogg's work [16] from 1983 is one of the first analysis-by-synthesis publications.

Using image subtraction to find the region of interest containing the subject, the edges of the projected model are compared to the edges present in that image region. A best fit, based on the plausibility value calculated for each possible configuration, determines the subject's pose in the image frame. Rohr [84] [87] follows Hogg's approach, adding several improvements in order to produce more robust results. The improvements include automatic determination of initial posture. Furthermore the model contours are compared with edge lines rather than grey-value edge points, and hidden model contours are removed automatically. A Kalman Filter estimates the model parameters, and a motion model tuned to walking is used.

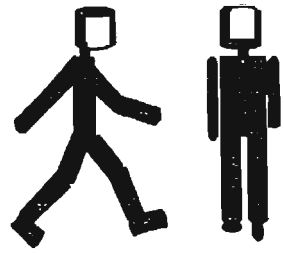


Figure 2.13: Elliptical cylinder model. (from Hogg [16])

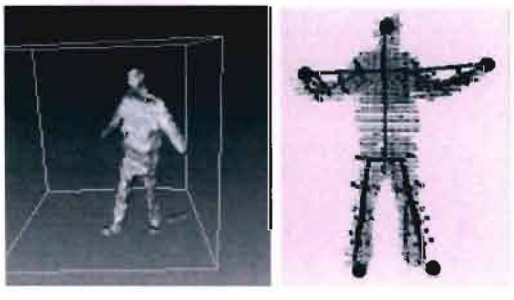
Huang and Lin [88] model the human body by 10 cylinders connected by 10 joints with 22 DOF. Assuming a static background, the object is segmented by background subtraction. The pose is estimated by projecting the 3-D model into the image and finding a best match between the projected model and silhouette. MPEG-4 BDPs are extracted in the first frame in which the subject is assumed to be standing upright. Subsequently, 22 BAPs are used to describe the pose of the model in each frame. Although successful tracking was achieved, Huang and Lin note that problems arise when limbs and body overlap. DigitEyes is a hand tracking system developed by Rehg and Kanade [82] [83]. Cylinders are used to represent the fingers, generating a 27 DOF model of a hand. The system updates the pose using inverse kinematics, and requires the knowledge of the kinematics, geometry and initial configuration to be known *a priori*. Results show the system successfully tracking two fingers occluding one another. Gonclaves *et al.* [98] use an analysis-by-synthesis approach to track a human arm, modelled as two truncated right-circular-cones with two spherical joints (elbow and shoulder), in monocular sequences. The 3-D arm model is

projected onto the image plane and fitted to the blurred and thresholded image of the real arm. Due to a dark background, the edges around the arm are very clear. The size and orientation of the arm model is adapted by minimising the error between the model projection and the real image. Moeslund and Granum [85] also try to estimate the pose of an arm in monocular sequences. Using analysis-by-synthesis they match the phase space model to real images. Unlike [98] however, their approach does not require initial parameters to be known, and is able to handle long term occlusions. In other work, Moeslund and Granum [86] use a calibrated stereo system to tackle the same problem. Investigation shows that the search space in analysis-by-synthesis can be reduced beyond the usual methods (dimensionality, kinematic and collision constraints) by using multiple cues: 3-D points and silhouette data. Results demonstrate the method's ability to estimate 3-D pose of an arm without constraints such as special coloured/textured/tight fitting clothes or markers, which are often imposed on similar systems.

Superquadratics are used when more accurate volumetric models are desired (Figure 2.14). A superquadratic is a shape that is some distortion of a sphere. It is defined parametrically in terms of longitude and latitude, with factors controlling scaling in x, y and z directions, and exponents controlling the squareness/roundness in the latitude and longitude directions. Combined in a hierarchical structure, superquadratics improve the human modelling accuracy, particularly for body parts such as the head and torso, and for regions close to articulation sites. Pentland [94] fits deformable superquadratics to range data using a physics-based method. The computational cost of the fitting process is reduced by modal analysis. Gavrila and Davis also use superquadratics. Their early work [92] [93] focuses on tracking and recognition of upper body movements using multiple views, later extending the method to cope with the full body [17]. Influenced by O'Rourke and Badler [89] the same tracking framework discussed earlier is adopted. Pose recovery is formulated as a search problem using a generate-and-test approach. A local search based on best-first search is performed, with search space dimensionality reduced by decomposition. The model proportions are derived automatically based on the subject's projection in two orthogonal views. A robust variant of chamfer matching is used to measure the similarity between the edges of the model projected into the multiple views, and the edges of the real data. To obtain better edges, subjects wear tight-fitting coloured clothes. When the best

Figure 2.14: Female and male 3-D human models using tapered superquadratics. (from Gavrila and Davis [17], ©1995 IEEE)

fit, highest similarity measure is found, the model is updated with the relevant parameters. Results show successful tracking of multiple humans performing complex motions such as dancing the Tango. Rather than searching for the best fit between the contours of a projected model, and the contours of real data, Delamarre and Faugeras [99] [100] apply physical forces that push the projection of the 3-D articulated model inside the real contour (silhouette) of the image of the person. The investigation concludes that tracking of a person from his silhouettes is possible, and even though the pose parameters of the model may not be accurate, the results are still visually good despite fast movements, self occlusion and changing light conditions. Kakadiaris and Metaxas [101] [102] [103] also use forces to align the model with image data. Additionally, their multi-view tracking system incorporates active camera selection based on the visibility of a body part and observability of its motion.



(a) (b)

Figure 2.15: (a) Visual hull from multiple silhouettes, (b) skeleton fitted to visual hull. (from Theobalt *et al.* [18])

Theobalt *et al.* [18] combine an efficient real-time optical feature tracker with visual hull reconstruction of a person, to fit a humanoid skeleton to the video footage from four calibrated views. In each time instance, the silhouette of the subject is generated by statistical background subtraction. Initially, the silhouette is separated into regions by a Generalized Voronoi Diagram Decomposition in order to locate the hands, head and feet. These body parts are tracked in 3-D by skin colour detection under the assumption that the subject is barefoot, and in parallel to this, the voxel-based approximation to the visual hull is computed. The first layer skeleton is fitted to the five 3-D points in real-time. Fitting of the second layer skeleton is done offline. This, more complex skeleton, uses the reconstructed volumetric data along with the tracked data to find the correct configuration of the limbs. Because the subject is required to always face the two front view cameras, only limited rotation about the vertical body axis is allowed.

D'Apuzzo [46] uses least squares matching to measure and track surfaces in multi-view video sequences. The user must manually determine the contour of the region to measure (silhouette of the subject), so the least squares matching process can generate a dense set of correspondences within the defined area in the three views. The correspondences are used for forward ray intersection, producing a 3-D point cloud representing the measured surface. Each surface point in the three views is tracked and verified by temporal and spatial least squares matching respectively. The surface tracking results in a dense field of trajectories with associated velocities and accelerations. This 3-D information is exploited to filter out falsely tracked points since each point on the surface must satisfy locally uniform motion. In order to describe the motion, D'Apuzzo manually selects key points in the point cloud, which are 3-D regions encapsulating portions of the 3-D vector field of trajectories that depict the motion of important joints. In sequences with complex motion this method could not successfully track all points throughout the sequence, a problem D'Apuzzo postulates could be resolved by increasing the number of cameras. In related work [19], D'Apuzzo uses the tracked data for realistic modelling of human bodies by fitting it to a simplified version of the layered model in Figure 2.16. The animation of the model is determined by the motion of the skeleton whose posture is set by the data extracted from the tracked key points. The flesh is modelled by ellipsoidal meatballs whose dimensions are generated from the 3-D surface data. The meatballs influence the

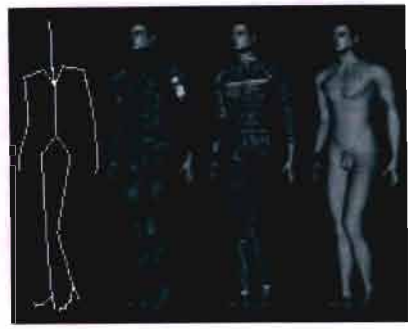


Figure 2.16: Layered human body model; skeleton, ellipsoidal metaballs representing flesh, polygonal surface representation of skin, shaded renedering. (from Plankers *et al.* [19])

skin represented by a polygonal surface, which can be shaded or texture mapped for a realistic looking human representation.

2.6 Conclusion

This chapter has discussed the topic of human motion capture from its beginnings in the late 19^{th} century to the current state of the art and ongoing research. The various methods of obtaining motion data were introduced as well as some applications that make use of such information. Out of all the possible methods of capturing motion, optical systems provide the only potential solution to non-invasive human motion capture in real-world unconstrained environments. Optical commercial systems rely heavily on very expensive hardware and artificial markers for accuracy and robustness, whilst researchers of image-based techniques formulate their approach depending on the application, introducing constraints and assumptions along the way, to achieve the final goal.

The goal of this research is to reconstruct human motion from video sequences. The problem is formulated as a model-based coding scheme in which the motion data extraction is performed at the analysis side (encoder) and the subsequent reconstruction at the synthesis end (decoder) of the system. The desired application directly influences which category (introduced in Section 2.5) the analysis approach falls under. The 2-D approach without explicit shape models bypasses the pose recovery step, immediately excluding it

as a possible solution to the problem. The 2-D approach with explicit shape models is only able to recover 2-D pose and 2-D motion, hence a 3-D approach must be adopted in order to acquire the subject's 3-D pose. It is possible to infer some 3-D information from monocular video sequences and a number of researchers have extracted pose from single views [78] [79] [80] [81] [89]. However, the recovered 3-D data is generally only known up to an unknown scale factor and the system is more susceptible to occlusion and pose ambiguity. When multiple calibrated cameras are used the 3-D reconstruction problem becomes straight forward [104]. Furthermore, the different views may be used for verification of tracked data [46], aiding in occlusion handling [17] [102] or resolving ambiguities in one view by the other views [105]. This research has recognised the benefits of multiple views and adopted this approach mainly due to the simplicity of 3-D reconstruction and the ability to verify tracked data.

The efficiency of a model-based coding system depends on how effectively it is able to describe the scene. Investigation by Preda *et al.* [34] has shown that the MPEG-4 Face and Body Animation (FBA) framework provides appropriate functionalities for virtual actor animation, gesture synthesis and compression/transmission with applicability to low bit-rate situations. This claim is further supported by Capin *et al.* [35] [36] who present favourable results from work concerning modelling and animating virtual bodies using MPEG-4. Thus the FBA framework defined in the MPEG-4 standard was a natural choice for the model-based coding system implemented in this research. The task of estimating FBA parameters from image data has thus far been tackled by only a few researchers [80] [88] [106], presenting an opportunity for novel research.

The tracking complexity is dictated by the number of DOF of the state [107]. In the case of model-based coding of the human body, a complete 3-D kinematic human model is required. To parameterise the overall pose, the locations of major joints have to be estimated, which when connected by rigid segments form a skeleton. To obtain the entire 3-D skeleton one may either track each joint separately [46] [79], fit the skeleton to a 2-D silhouette [80], fit the skeleton to a visual hull [18] or generate all possible poses from a known starting point and using various constraints to recover the most plausible solution [78]. Alternatively, one can use a volumetric model to find the overall pose by matching

the projected model to the segmented image [81] [88], extracted silhouette [100] [102] or edge data [17]. A volumetric model may not have an explicitly defined skeleton, however the underlying skeleton always exists due to the points of articulation joining each body part. However, since most of these models use a single volume to represent the torso, the DOF of the spine is limited. Similarly, the DOF is limited in the case of fitting the skeleton to the visual hull [18] and 2-D silhouette [80]. Furthermore, in the last two cases some of the actual joint locations are estimated from the relevant data introducing a level of uncertainty to the resulting pose parameters [80]. In this research 18 joints are tracked in a 31 DOF model. The additional DOF come from a more complex spine model and the introduction of the left and right clavicula joints. Since these DOF would be annulled by a single volume representation of the torso, the approach of tracking individual joints was adopted rather than matching a volumetric representation of the subject to the image data. Thus the locations representing the key joints on the human body have to be found in each frame. Theobalt *et al.* [18] track the hands, feet and head using skin detection while Wren and Pentland [15] [67] [68] and Azarabeyjani and Pentland [66] track the hands and head as blobs that not only consider colour but shape characteristics as well. These methods track relatively large regions, and assume the corresponding joint lies at the centre of that region. Neither of these methods could be applied to tracking all 18 points required in this work. Skin colour detection will work only for hands, head and feet, and only if the subject is wearing long sleeves, long pants and no shoes. Tracking using blobs has been applied to the whole body in the 2-D case [11] and could be extended to 3-D by [66]. However, blobs represent large regions which reduce the overall DOF, furthermore accurate locations of all joints cannot be estimated. The solution by D'Apuzzo [46] uses image matching techniques to track points in subsequent frames as well as within the different views. That approach was adopted because it utilises least squares matching which is able to find correspondences with sub-pixel accuracy. This not only ensures that points are correctly tracked in the sequence, but more importantly finds accurate correspondences within the multiple views which is essential for the 3-D reconstruction phase. The tracking process only requires estimates at the joint level which is adequately approximated by a skeleton model.

The two main limitations of the presented system come in the form of manual initialisa-

tion and the assumption that all tracked locations are visible throughout the sequence. Although both limitations make real world applications impractical, they are not uncommon in current research. Several systems specifically rely on key locations to be provided *a priori* [14] [46] [78] [79] [80] [81] [89] [108], and a number of approaches cannot handle or have difficulty with occlusion [13] [46] [59] [71] [79] [82] [109] [110]. Both aspects reserve room for improvement of the current system, and are discussed in that context in Chapter 8.

The task presented to the human motion capture system is to produce 3-D locations of selected joints in each frame to enable the construction of the 3-D skeleton and subsequent estimation of 31 BAPs. The key concept is acquiring 3-D information from 2-D images, and the following chapter presents the necessary theory which makes this process possible.

Chapter 3

Fundamental Theory

This chapter introduces the fundamental computer vision theory that provides the link between 2-D images and the real world. Since 3-D locations of joints are required for the pose parameter extraction process, the relationship between a point in 3-D space and its projection on the image plane needs to be established by some camera model. This relationship is expressed in terms of intrinsic and extrinsic camera parameters which are found through camera calibration. A correspondence in stereo views together with known camera parameters allows the unambiguous triangulation of the point's 3-D location in world space. Furthermore, a multiple camera system allows the exploitation of the epipolar constraint inherent in the geometry of two views, which becomes useful in the matching and tracking processes that form the human motion capture system.

3.1 Notation

The concepts introduced in this chapter are part of the foundation upon which computer vision is based. Unless otherwise indicated, the notation and derivations follow those of Trucco and Verri [104]. Boldface letters refer to matrices and vectors. Furthermore, a boldface $\mathbf{P}$ refers to a 3-D point with $[X, Y, Z]$ coordinates, and where necessary, the reference frame in which it is measured is indicated by subscripts c and w for camera and

world reference frames respectively. The projection of **P** in the image plane is represented by **p**, with a subscript *im* indicating when the measurement is in pixels. Subscripts l and r refer to the left and right camera reference frames respectively, with rl indicating a measurement in the right view with respect to the left reference frame. Column vectors discussed in the body text are written as a transpose indicated by a superscript T.

3.2 Camera Model

The most widely used geometric model of intensity cameras is the perspective model shown in Figure 3.1. Also known as the pinhole model, it consists of an image plane π and a centre of projection (COP) **O** which is the origin of the 3-D camera coordinate system. The z-axis is perpendicular to the image plane, a focal length distance f away from the centre of projection. The point at which the z-axis and image plane intersect is called the principal point **o**.

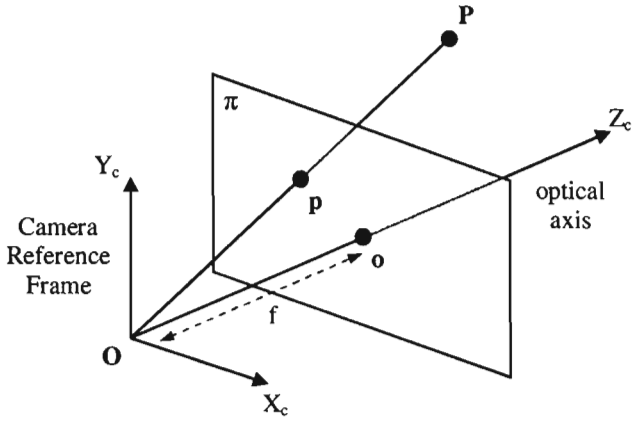


Figure 3.1: Perspective camera model.

Under perspective projection, a 3-D point $\mathbf{P} = [X, Y, Z]$ in the camera reference frame is mapped to **p** by similar triangles (Figure 3.2) resulting in a set of equations (3.1). This is a mapping from Euclidean 3-space $\mathbb{R}^3$ to Euclidean 2-space $\mathbb{R}^2$.

$$\begin{aligned} x &= f\frac{X}{Z} \\ y &= f\frac{Y}{Z} \end{aligned} \tag{3.1}$$

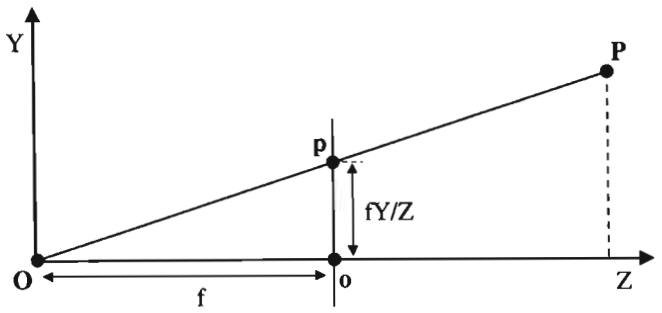


Figure 3.2: Similar triangle relationship under perspective projection (shown only for y-component).

3.3 Camera Parameters

Camera characteristics are described by extrinsic and intrinsic parameters. Extrinsic parameters define the location and orientation of the camera reference frame with respect to a known world reference frame by a translation vector $\mathbf{T}$ and an orthogonal rotation matrix $\mathbf{R}$. With these parameters, one is able to link the world coordinates $\mathbf{P}_w$ of a point $\mathbf{P}$ to its corresponding camera coordinates $\mathbf{P}_c$ by equation (3.2),

$$\mathbf{P}_c = \mathbf{R}(\mathbf{P}_w - \mathbf{T}) \tag{3.2}$$

where $\mathbf{R}$ is a 3x3 rotation matrix that aligns the corresponding axes of the two reference frames and $\mathbf{T}$ is a 3x1 vector of the distance between the two origins.

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \qquad \mathbf{T} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{3.3}$$

Intrinsic parameters describe the optical, geometric and digital characteristics of a camera. The parameters are related to the perspective projection, the transformation between camera and pixel coordinates, and the geometric distortion introduced by the optics.

The image projected onto the image plane is digitised into discrete pixel values by the sensor. The output is a matrix of intensity values, with each entry representing one

pixel. In reality, the pixels have physical size. Knowing the pixel size, together with the location of the principal point in the image plane, the relationship between image and pixel coordinates is given by equation (3.4),

$$\begin{aligned} x &= (x_{im} - o_x)s_x \\ y &= -(y_{im} - o_y)s_y \end{aligned} \tag{3.4}$$

where:

s_x is the horizontal pixel length in millimetres

s_y is the vertical pixel length in millimetres

o_x is the x coordinate of the principal point in pixels

o_y is the y coordinate of the principal point in pixels

The image coordinates are defined such that the y-axis points in the opposite direction to that of the pixel coordinates, hence the sign change of y in equation (3.4).

The camera optics often introduce radial and tangential distortions that become more evident towards the periphery of the image. Fortunately the distortion can be modelled and the distorted coordinates $[x_d, y_d]^T$ in the camera reference frame can be corrected for. The level of distortion depends on the quality of the lens used and the complexity of the distortion model is based on the desired accuracy. In some cases the distortion may be ignored altogether, however in the presented work it is modelled as in [111] by a sixth order polynomial. The coefficients (k_1,k_2,k_3,k_4,k_5) of the sixth order polynomial are stored in the five element vector **k** and are applied to the distorted coordinates according to equation (3.5).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \left(1 + k_1 r^2 + k_2 r^4 + k_5 r^6\right) \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \mathbf{dx} \tag{3.5}$$

where:

$$r^2 = x_d^2 + y_d^2 \ , \qquad \mathbf{dx} = \begin{bmatrix} 2k_3 x_d y_d + k_4 \left(r^2 + 2x_d^2\right) \\ k_3 \left(r^2 + 2y_d^2\right) + 2k_4 x_d y_d \end{bmatrix} \tag{3.6}$$

3.4 The Perspective Camera

Sections 3.2 and 3.3 linked the various reference frames via camera parameters. In general only the pixel data is available to the user, which necessitates the relationship between pixel coordinates and the world coordinates. The camera parameters can be grouped into two matrices $\mathbf{M}_{int}$ and $\mathbf{M}_{ext}$ encoding the intrinsic and extrinsic parameters respectively as in equation (3.7),

$$\mathbf{M}_{\text{int}} = \begin{bmatrix} f/_{s_x} & 0 & o_x \\ 0 & -f/_{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{M}_{ext} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -\mathbf{R}_1^T \cdot \mathbf{T} \\ r_{21} & r_{22} & r_{23} & -\mathbf{R}_2^T \cdot \mathbf{T} \\ r_{31} & r_{32} & r_{33} & -\mathbf{R}_3^T \cdot \mathbf{T} \end{bmatrix} \tag{3.7}$$

where $\mathbf{R}_i$, i = 1,2,3 is a vector formed by the i-th row of the rotation matrix $\mathbf{R}$. Hence the linear matrix equation of normalised perspective projection can be written:

$$\begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \mathbf{M}_{\text{int}}\mathbf{M}_{ext} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3.8}$$

Equation (3.8) encompasses the transformations between the various reference frames of a vision system as depicted in Figure 3.3.

3.5 Camera Calibration

Camera calibration is an essential preliminary step for most applications of 3-D computer vision. The goal of camera calibration is to recover the intrinsic and extrinsic camera parameters. These parameters estimate the geometric and optical camera characteristics, and the 3-D position and orientation with respect to a defined world reference frame according to the models described in Sections 3.2 - 3.4. There are two main categories under which camera calibration methods fall [112] viz. photogrammetric calibration and self-calibration. Photogrammetric calibration requires precise *a priori* knowledge of the calibration object's geometry in 3-D space. The calibration object generally consists of

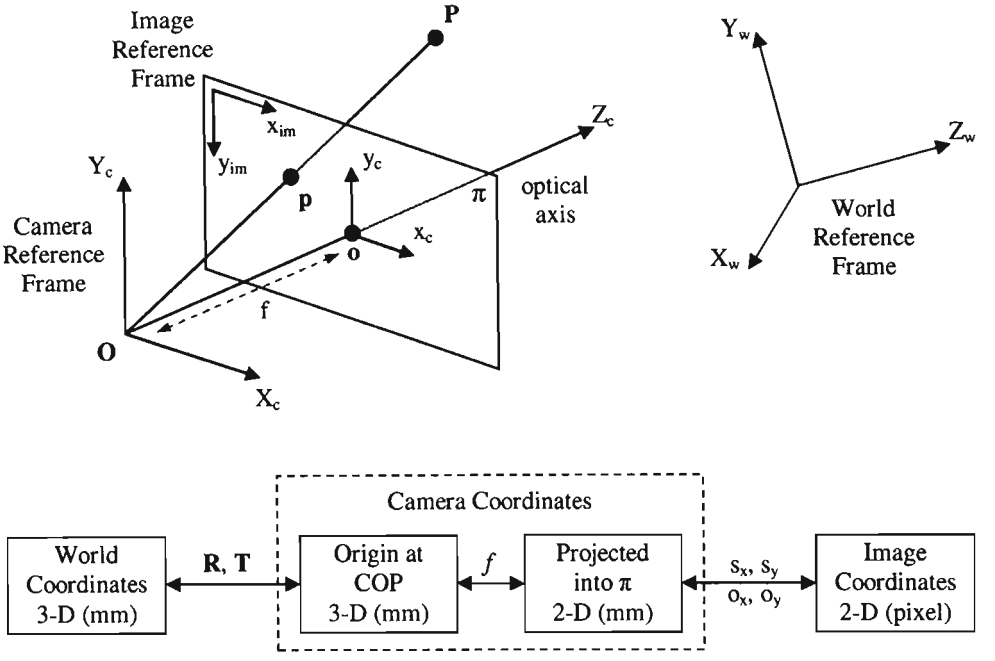


Figure 3.3: Relationship between reference frames of a visual system.

two or three orthogonal planes. Although producing good results, this method is time consuming, and requires expensive equipment and an elaborate setup. These drawbacks may make photogrammetric calibration unfeasible for some applications. Self-calibration on the other hand is simple, easy and requires no calibration object. Camera parameters are recovered from correspondences in images of a static scene. Although this method is feasible for almost any setup, due to the number of parameters to estimate, the results are not always reliable [112]. Zhang [112] has proposed a solution which is a combination of these two methods. It uses a planar pattern that is moved around in space to generate several views of the object in different poses. The pattern itself can be printed on a laser printer and attached to a planar surface and the knowledge of the motion is not required. Both intrinsic and extrinsic camera parameters are found explicitly. This method has become popular among researchers due to its simpler and cheaper setup when compared to photogrammetric calibration, whilst at the same time producing more accurate and robust results in comparison to the self-calibration technique.

The cameras used to capture data for this research were calibrated using the Matlab

Calibration Toolbox implemented by Jean-Yves Bouguet [111]. C++ implementation of this calibration is also freely available, as part of the Intel OpenCV library [113]. These two libraries are probably the most widely used tools for camera calibration. Bouguet's approach is based on Zhang's work [112] [114]. The calibration is done in two steps; initialisation and non-linear optimisation. The initialisation computes a closed-form solution for the calibration parameters excluding lens distortion. Subsequent non-linear optimisation computes all the intrinsic and extrinsic parameters by minimising the total reprojection error in the least squares sense. The main difference between Bouguet's and Zhang's approach is that Bouguet implements the intrinsic camera model by Heikkila and Silven [115]. Heikkila and Silven incorporate extra two parameters into the lens distortion model to take into account tangential distortion in addition to radial distortion.

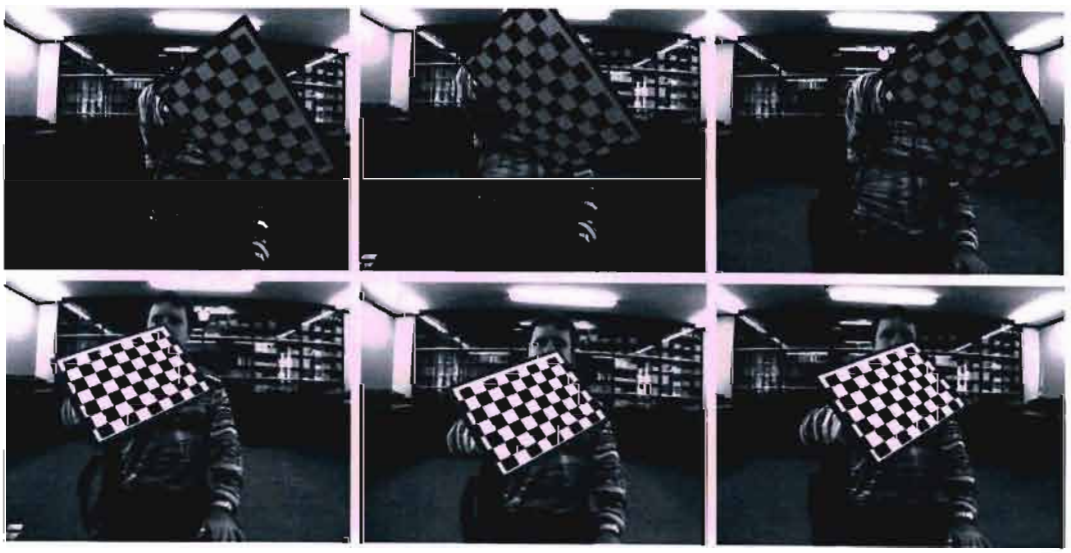
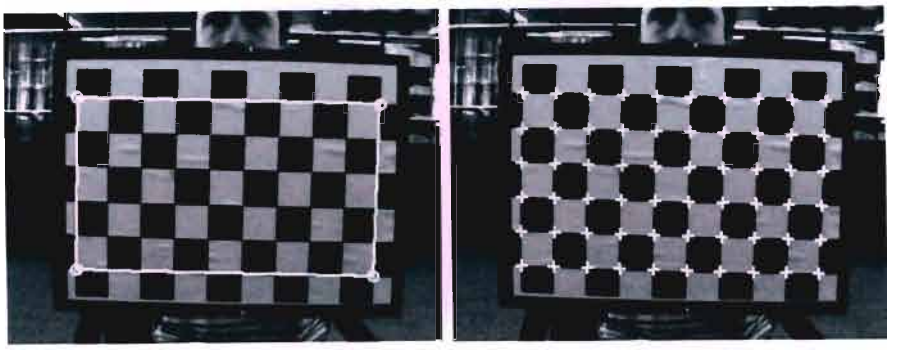


Figure 3.4: 2 out of 20 calibration images for left, middle and right view cameras.

The calibration procedure itself requires the user to capture several images of a calibration pattern in various poses for all cameras as shown in Figure 3.4. The user must manually select the four corners that define the region of interest (Figure 3.5(a)), and then the checkerboard corners are detected automatically (Figure 3.5(b)). Once each camera is calibrated separately, the toolbox allows stereo calibration, which produces the rotation matrix and translation vector that relate the two camera reference frames to each other. The relationship is such that points $\mathbf{p}_l$ and $\mathbf{p}_r$, the projections of a 3-D point $\mathbf{P}$ in the left



(a) (b)

Figure 3.5: (a) Calibration pattern region of interest, (b) detected corners.

and right camera reference frames respectively, are related in [111] by equation (3.9).

$$\mathbf{p}_r = \mathbf{R}_{rl}\mathbf{p}_l + \mathbf{T}_{rl} \tag{3.9}$$

The extrinsic parameters of a stereo system, $\mathbf{R}_{rl}$ and $\mathbf{T}_{rl}$ are calculated from the extrinsic parameters of the left and right view by equation (3.10).

$$\begin{aligned} \mathbf{R}_{rl} &= \mathbf{R}_r\mathbf{R}_l^T \\ \mathbf{T}_{rl} &= \mathbf{T}_r - \mathbf{R}_{rl}\mathbf{T}_l \end{aligned} \tag{3.10}$$

3.6 Epipolar Geometry

In the case of two cameras looking at the same point, the epipolar geometry is the basic constraint that arises from the existence of the two viewpoints. The epipolar geometry is the intrinsic projective geometry between two views. It is independent of scene structure, and only depends on the cameras' internal parameters and relative pose.

3.6.1 Epipolar Constraint

The epipolar geometry is shown in Figure 3.6. Point $\mathbf{P}$ is observed by two cameras with optical centres $\mathbf{O}_l$ and $\mathbf{O}_r$, producing images $\mathbf{p}_l$ and $\mathbf{p}_r$. These points all lie in the epipolar plane defined by the two intersecting rays $\mathbf{O}_l\mathbf{P}$ and $\mathbf{O}_r\mathbf{P}$. More importantly, the point $\mathbf{p}_r$ lies on the line $\mathbf{l}_r$ formed at the intersection of the epipolar plane and the image plane

π_r of the right camera. The epipolar line $\mathbf{l}_r$ is associated with the point $\mathbf{p}_l$, and likewise, the line $\mathbf{l}_l$ on which the point $\mathbf{p}_l$ lies is associated with the point $\mathbf{p}_r$. Both the left and right epipolar lines pass through the points $\mathbf{e}_l$ and $\mathbf{e}_r$ respectively, which are located at the intersections of the baseline joining the optical centres $\mathbf{O}_l$ and $\mathbf{O}_r$ with the respective image planes. The points $\mathbf{e}_l$ and $\mathbf{e}_r$, called the epipoles of the two cameras, are the (virtual) images of the opposite view's optical centre.

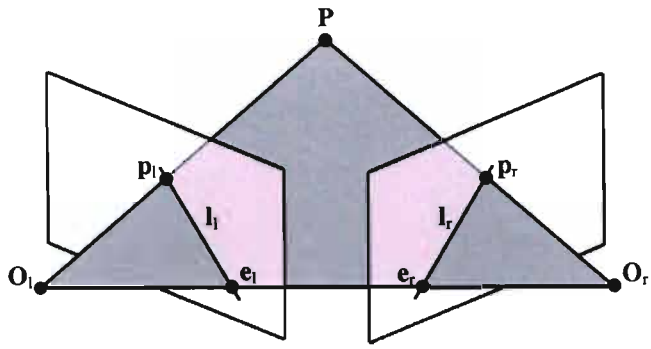


Figure 3.6: Epipolar geometry.

The practical importance of epipolar geometry stems from the fact that the epipolar plane intersects each image in a line, the epipolar line ($\mathbf{l}_l$ and $\mathbf{l}_r$). Given $\mathbf{p}_l$, the epipolar plane is determined by the baseline and the ray defined by $\mathbf{p}_l$. The line $\mathbf{l}_r$ is the image in the second view of the ray back-projected from $\mathbf{p}_l$. Thus the corresponding point, $\mathbf{p}_r$, is constrained to lie on the epipolar line $\mathbf{l}_r$ (Figure 3.7). This is known as the epipolar

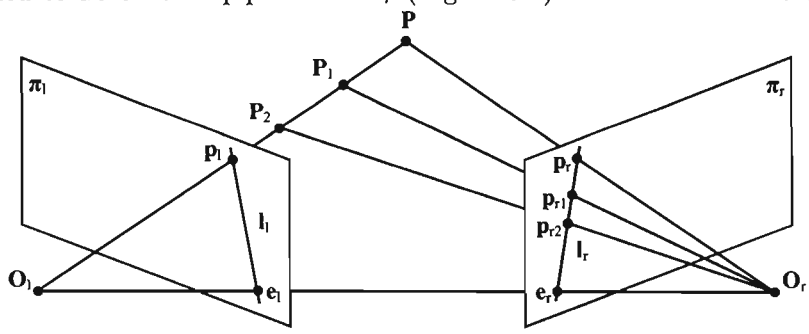


Figure 3.7: Epipolar Constraint: the set of possible matches for the point $\mathbf{p}_l$ is constrained to lie on the associated epipolar line $\mathbf{l}_r$.

constraint, which establishes a mapping between points in the left image, lines in the right

image and vice-versa. This can effectively be used in correspondence problems whereby the search for correspondences is thus reduced to a 1-D problem by searching only along the epipolar line rather than in the whole image. Alternatively, the same knowledge can be used to verify whether or not a candidate match lies on the corresponding epipolar line.

The baseline, the line connecting $\mathbf{O}_l$ and $\mathbf{O}_r$, defines a family of planes. As the position of $\mathbf{P}$ varies, the epipolar planes "rotate" about the baseline. This family of planes is referred to as an epipolar pencil (Figure 3.8). Its intersection with each image plane is a family of lines going through the points $\mathbf{e}_l$ and $\mathbf{e}_r$. These two families are pencils of lines called the pencils of epipolar lines. It has been shown in [116] that epipolar planes induce a natural correspondence between the two pencils of epipolar lines, i.e. two epipolar lines correspond to each other if and only if they belong to the same epipolar plane.

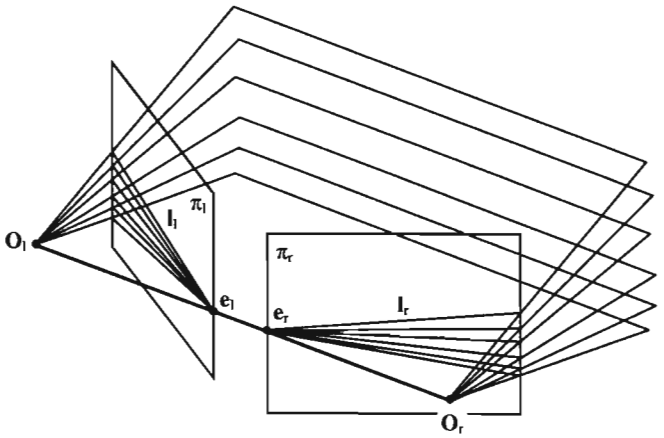


Figure 3.8: Epipolar Pencil.

3.6.2 The Essential Matrix

The essential matrix was first introduced in the now classic paper by Longuet-Higgins [117]. The essential matrix was calculated by the 8-point algorithm, and used to compute the structure of a scene from two views with calibrated cameras. The following explanation of the essential matrix (and in the subsequent section the fundamental matrix) is based on the descriptions by Trucco and Verri [104]. Trucco and Verri use equation (3.11) to relate the left and right camera reference frames, which is a slight variation of equation (3.9)

defined in the calibration section, where rotation preceded translation.

$$\mathbf{p}_r = \mathbf{R}_{rl}(\mathbf{p}_l - \mathbf{T}_{rl}) \tag{3.11}$$

While the rotation matrix remains the same in both cases, the translation vector is different, and now determined by equation (3.12).

$$\mathbf{T}_{rl} = \mathbf{T}_l - \mathbf{R}_{rl}^T\mathbf{T}_r \tag{3.12}$$

Defining vectors $\mathbf{P}_l$ and $\mathbf{T}$ as in Figure 3.9, the equation of the epipolar plane is given by the following coplanarity condition:

$$(\mathbf{P}_l - \mathbf{T})^T\mathbf{T} \times \mathbf{P}_l = 0 \tag{3.13}$$

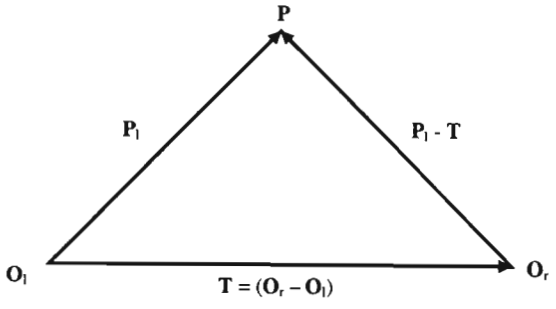


Figure 3.9: Coplanarity condition.

Using equation (3.11), equation (3.13) is rewritten as:

$$(\mathbf{R}_{rl}^T\mathbf{P}_r)^T\mathbf{T} \times \mathbf{P}_l = 0 \tag{3.14}$$

The cross product by $\mathbf{T}$ can be expressed as matrix multiplication where:

$$\mathbf{S} = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \tag{3.15}$$

so that equation (3.14) becomes,

$$\mathbf{P}_r^T\mathbf{E}\mathbf{P}_l = 0 \tag{3.16}$$

with:

$$\mathbf{E} = \mathbf{R}_{rl}\mathbf{S} \tag{3.17}$$

By construction, $\mathbf{S}$ always has rank 2. The matrix $\mathbf{E}$ is called the essential matrix and links the epipolar constraint and the extrinsic parameters of a stereo system. Recalling equation (3.1), which in vector form becomes (3.18),

$$\mathbf{p} = \frac{f}{Z}\mathbf{P} \tag{3.18}$$

by applying eqution (3.18) to both $\mathbf{P}_r$ and $\mathbf{P}_l$ and dividing by $Z_l Z_r$, equation (3.16) can be rewritten as,

$$\mathbf{p}_r^T \mathbf{E} \mathbf{p}_l = 0 \tag{3.19}$$

which defines the essential matrix in terms of camera coordinates. Equation (3.19) defines a mapping between points and epipolar lines, i.e. it is satisfied by every point $\mathbf{p}_l$ lying on the left epipolar line. The equation of the right epipolar line due to an arbitrary point $\mathbf{p}_l$ in the left image plane can be attained by equation (3.20) (see Appendix A for the homogenous projective representation of lines).

$$\mathbf{l}_r = \mathbf{E}\mathbf{p}_l \tag{3.20}$$

Properties of the essential matrix:

- encodes information on the extrinsic parameters only
- has rank 2, since $\mathbf{S}$ in equation (3.17) has rank 2 and $\mathbf{R}_{rl}$ full rank
- its two nonzero singular values are equal

3.6.3 The Fundamental Matrix

The essential matrix works in the camera reference frame which is generally hidden from the user. When one wants to proceed directly from pixel measurements, the fundamental matrix first introduced by Faugeras *et al.* [116] is used. Interestingly, because the fundamental matrix works with pixel measurements, it can be recovered from pixel matches alone without any information about the intrinsic and extrinsic parameters of the stereo system.

Suppose that $\mathbf{M}_l$ and $\mathbf{M}_r$ are the matrices of the intrinsic parameters of the left and right cameras respectively, and $\bar{\mathbf{p}}_l$ and $\bar{\mathbf{p}}_r$ are the pixel coordinates of $\mathbf{p}_l$ and $\mathbf{p}_r$ respectively,

the relationship described by equation (3.21) exists.

$$\begin{aligned} \mathbf{p}_l &= \mathbf{M}_l^{-1}\bar{\mathbf{p}}_l \\ \mathbf{p}_r &= \mathbf{M}_r^{-1}\bar{\mathbf{p}}_r \end{aligned} \tag{3.21}$$

Substituting equation (3.21) into equation (3.19),

$$\bar{\mathbf{p}}_r^T\mathbf{F}\bar{\mathbf{p}}_l = 0 \tag{3.22}$$

where:

$$\mathbf{F} = \mathbf{M}_r^{-T}\mathbf{E}\mathbf{M}_l^{-1} \tag{3.23}$$

$\mathbf{F}$ is called the fundamental matrix. $\mathbf{F}\bar{\mathbf{p}}_l$ in equation (3.22) equivalently to $\mathbf{E}\mathbf{p}_l$ in equation (3.19) can be thought of as the equation of the projective epipolar line $\bar{\mathbf{l}}_r$ corresponding to the point $\bar{\mathbf{p}}_l$, or

$$\bar{\mathbf{l}}_r = \mathbf{F}\bar{\mathbf{p}}_l \tag{3.24}$$

Alternatively, $\bar{\mathbf{p}}_r$ corresponding to $\bar{\mathbf{p}}_r$ is found by equation (3.25).

$$\bar{\mathbf{l}}_l = \mathbf{F}^T\bar{\mathbf{p}}_r \tag{3.25}$$

The epipoles are located using equation (3.26).

$$\mathbf{F}\bar{\mathbf{e}}_l = 0, \qquad \mathbf{F}^\mathbf{T}\bar{\mathbf{e}}_r = 0 \tag{3.26}$$

The most important difference between equations (3.20) and (3.24), and therefore between the essential and fundamental matrices is that the former is defined in camera coordinates whilst the latter in pixel coordinates. Thus if one estimates the fundamental matrix from pixel correspondences, the epipolar geometry may be recovered without *a priori* knowledge of the intrinsic and extrinsic camera parameters.

The fundamental matrix is a generalisation of the essential matrix, and is commonly used with uncalibrated systems. It is important because it captures, in a very compact manner, the epipolar geometry of the two images [118]. The fundamental matrix is used in many applications such as affine and projective reconstruction, computation of projective invariants, uncalibrated stereo matching, calibration and auto-calibration, image rectification, and image synthesis.

The fundamental matrix may be computed from a certain number of point correspondences obtained from the two images of the stereo setup. An often cited method for computing the fundamental matrix is the 8-point algorithm, introduced by Longuet-Higgins [117], which requires 8 or more point matches. It has the advantage of simplicity of implementation, because the linear criterion leads to a non-iterative computation method. However, the algorithm has been found to be quite sensitive to noise even with numerous data points [118], resulting in the algorithm being viewed as inadequate and inferior to the iterative methods. Recently Hartley [119] has challenged this view, basing his approach on the observation that the poor performance of the 8-point algorithm is due, for the most part, to poor numerical conditioning. He has proposed a normalised 8-point algorithm, showing results comparable, though not bettering those of the iterative techniques, reviewed in [118] and [120].

The methods in [117] [118] [119] [120] require point correspondences in the image pairs. The data points are always corrupted by noise, and sometimes the matches are even spurious or incorrect. As a result, Csurka *et al.* [121] have modelled the uncertainty of the estimated fundamental matrix in order to exploit its underlying geometric information correctly and effectively. The uncertainty is given by the covariance matrix which is calculated by either the statistical or analytical method. Once obtained, it may be used to define the epipolar band for stereo matching, computation of uncertainty in projective reconstruction as well as the improvement of self-calibration techniques based on Kruppa equations [116].

3.7 3-D Reconstruction

3-D reconstruction generates a three dimensional representation of a scene from two dimensional images. The degree of reconstruction that can be obtained from stereo images depends on the amount of available *a priori* knowledge of the system, summarised in Table 3.1.

In this work both the intrinsic and extrinsic parameters have been recovered through

Table 3.1: Degrees of 3-D reconstruction vs *a priori* knowledge of camera parameters

a priori knowledge	3-D reconstruction
intrinsic and extrinsic parameters	unambiguous (absolute coordinates)
intrinsic parameters only	up to an unknown scaling factor
no parameters	up to an unknown global projective transformation

calibration, hence 3-D locations of points can be obtained from their projections in the left and right views through triangulation. Assuming two corresponding points $\mathbf{p}_l$ and $\mathbf{p}_r$ in the left and right views respectively are known, the rays originating at each centre of projection and passing through the respective image points should theoretically intersect in 3-D space, at the location of the real point. In practice however this will not happen due to several reasons:

- Models describing the camera geometry and distortion are approximations and not exact
- Calibration parameters carry uncertainty due to approximate camera models and noisy image data
- Correspondence match inaccuracy

As a result, the 3-D point is estimated at the point of minimum distance between both rays. The triangulation problem is represented in Figure 3.10 and solved according to [104]. Let l, the ray through $\mathbf{O}_l$ and $\mathbf{p}_l$ be described by $a\mathbf{p}_l$ ($a \in \mathbb{R}$) and the ray r going through $\mathbf{O}_r$ and $\mathbf{p}_r$ be described in the left camera reference frame by $\mathbf{T}_{rl} + b\mathbf{R}_{rl}^T\mathbf{p}_r$ ($b \in \mathbb{R}$). The vector $\mathbf{w}$ is orthogonal to both l and r. The task now is to determine the location of the point $\mathbf{P}$, which lies at the mid-point of the shortest segment joining l and r, defined by $\mathbf{P}_1$ and $\mathbf{P}_2$, and parallel to $\mathbf{w}$. Solving the linear system of equations (3.27) for a_0, b_0 and c_0,

$$a\mathbf{p}_l - b\mathbf{R}_{rl}^T\mathbf{p}_r + c\left(\mathbf{p}_l \times \mathbf{R}_{rl}^T\mathbf{p}_r\right) = \mathbf{T}_{rl} \tag{3.27}$$

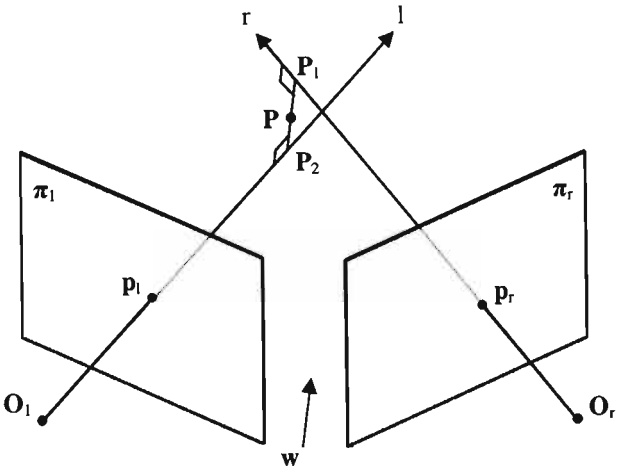


Figure 3.10: Triangulation with non-intersecting rays.

the points $\mathbf{P}_1$ and $\mathbf{P}_2$ are found by equation (3.28),

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{T}_{rl} + b_0 \mathbf{R}_{rl}^T \mathbf{p}_r \\ \mathbf{P}_2 &= a_0 \mathbf{p}_l \end{aligned} \tag{3.28}$$

and finally the 3-D location of the point $\mathbf{P}$ in the left camera reference frame is given by equation (3.29).

$$\mathbf{P} = {}^1\!/_2(\mathbf{P}_1 + \mathbf{P}_2) \tag{3.29}$$

3.8 Summary

This chapter has introduced some important concepts in terms of the work relating to the presented research as well as computer vision in general. The perspective camera model provides a link between the 2-D pixel coordinates and the 3-D world coordinates. The extra effort involved with camera calibration is justified by enabling unambiguous 3-D reconstruction, as well easy calculation of the epipolar geometry, an important relationship between multiple views. The following chapters will discuss image matching, the tracking algorithm and pose parameter extraction. Image matching finds pixel correspondences in two images. The tracking algorithm tracks the various pixel locations throughout the video sequence and produces their 3-D location through triangulation. Once these 3-D

locations are known, the pose parameter extraction process can generate the relevant information pertaining to the subject's pose. It is clear how the theory fits into the overall human motion capture system, as it generates 3-D motion data from an input array of grey levels.

Chapter 4

Digital Image Matching

Digital image matching is at the heart of the human motion capture system as implemented in this research. By finding accurate correspondences in different images it is possible to track points in subsequent frames as well as to reconstruct the 3-D skeleton. In this work two techniques have been implemented, cross-correlation and least squares matching (LSM). This chapter presents the respective mathematical models together with the advantages and disadvantages of each method and some typical results.

4.1 Overview

Digital image matching is a crucial component of almost any image analysis process both in photogrammetry and computer vision. Matching is necessary for the generation of Digital Terrain Models (DTM), scene reconstruction and often a prerequisite for object detection, classification and identification. The correspondence problem can be broadly classified into two main categories, feature-based and correlation-based matching techniques. The former method matches features that have been extracted from the images in a pre-processing step, while the latter works directly with the intensity profiles of the images.

Feature matching methods first extract salient features from the images using feature extraction algorithms. The types of features used depend on the particular problem, but

may include one or more low-level features such as edges, corners, line segments, curve segments, and/or higher-level features such as circles, ellipses or polygonal regions. The matching process uses the attributes (feature descriptors) associated with each feature to find a corresponding pair. First, a preliminary list of candidate corresponding features is built up. This list is then thinned through assumptions (e.g. object's local geometry) or constraints (epipolar, uniqueness or continuity). The best match is found using the measure of the distance between the feature descriptors. Feature-based methods are suitable in cases when *a priori* information about the scene is available, so that optimal features can be selected.

The correlation-based techniques may also be referred to as signal, window, or area-based methods. Point matching is performed by considering a neighbourhood of pixels around the point constituting a subset of the original image, which in the subsequent discussions is referred to as an image patch. In addition, the patch defined around the point of interest whose correspondence in another image is desired is called the template patch. One correlation-based approach uses a statistical measure, shifting the template patch over the search space of the second view and calculating a similarity value at each position as in Figure 4.1. Alternatively, one may define the unknown corresponding coordinates in the search area as a function of the grey values of the target and search regions and iteratively solve the non-linear system to obtain the match location.

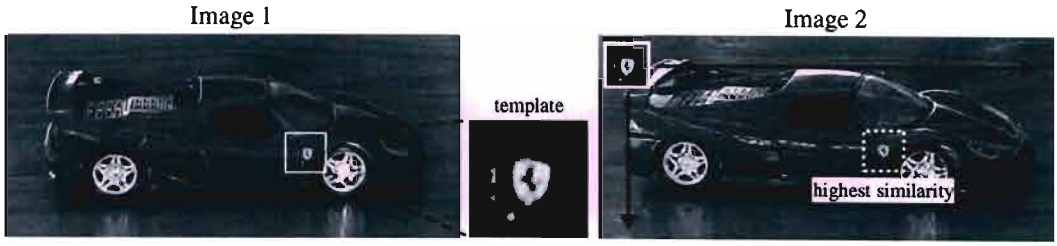


Figure 4.1: Searching for best match using similarity measures.

Although in terms of surface discontinuities and requirements for approximate values feature-based matching is more flexible, correlation methods have a higher potential for accuracy [122]. Moreover, keeping in mind the application of matching to human motion capture where specific locations have to be tracked, it is inevitable that adequate features

will be absent at some locations, hence correlation-based methods have been adopted in this work.

4.2 Grey Value Correlation

One of the most commonly used area-based matching techniques is cross-correlation. A template patch is shifted over the search space as in Figure 4.1 and the similarity at each locations is represented by the correlation coefficient R_{ts}. The correspondence in the search image is located at the position of the highest R_{ts}.

Denoting the template patch by f_i and the search area by g_i, where i = 1,2,...,n with n the number of pixels in the patch, the common similarity measure, discrete correlation, is given by equation (4.1).

$$R_{ts} = \sum_i f_i \cdot g_i \tag{4.1}$$

Normalising equation (4.1) by the means $\bar{f}$ and $\bar{g}$, and the second moments $\sum_i (f_i)^2$ and $\sum_i (g_i)^2$, the normalised cross-correlation (NCC) coefficient is given by equation (4.2),

$$R_{ts} = \frac{\sum_i (f_i - \bar{f})(g_i - \bar{g})}{\sqrt{\sum_i (f_i - \bar{f})^2 \cdot \sum_i (g_i - \bar{g})^2}} \tag{4.2}$$

with the property $-1 \leq R_{ts} \leq 1$.

By normalising the cross-correlation, the correspondence measure is insensitive to luminance scale and level. Burt *et al.* [123] showed that normalised cross-correlation gave the lowest error rates when compared to non-normalised or partially normalised correlation.

Cross-correlation treats an image as a 2-D signal disregarding the 3-D structure of the scene. Particularly in close range scenarios, when the 3-D scene is observed from different viewpoints, or the 3-D objects undergo motion, the resulting two images may differ considerably. Cross-correlation matching cannot account for this geometric distortion between the two images, but to reduce its effect a weighting function w_i may be introduced [124]. Noting that the geometric distortion is more pronounced towards the edges of an object

than its central part, the weighting function is defined such that it favours the central pixels of the patch as shown in Figure 4.2.

1	1	1	1	1
1	3	3	3	1
1	3	5	3	1
1	3	3	3	1
1	1	1	1	1

Figure 4.2: An example of the weight distribution of pixels in a 5×5 patch.

When the weighting function is incorporated into the matching process, the normalised cross-correlation is given by equation (4.3).

$$R_{ts} = \frac{\sum_i w_i \left(f_i - \bar{f}\right)\left(g_i - \bar{g}\right)}{\sqrt{\sum_i w_i \left(f_i - \bar{f}\right)^2 \cdot \sum_i w_i \left(g_i - \bar{g}\right)^2}} \tag{4.3}$$

The matching process involves shifting the template patch over the search space and calculating the cross-correlation coefficient R_{ts} at each step. This produces correlation values at discrete locations, resulting in the matching accuracy to be within ±0.5 pixel in both x and y directions. To improve the accuracy, Webber [125] fits a second order polynomial to the discrete values in the neighbourhood of the highest R_{ts} (R_{max}).

$$\begin{aligned} h_x(r) &= d_0 + d_1 x + d_2 x^2 \\ h_y(r) &= e_0 + e_1 y + e_2 y^2 \end{aligned} \tag{4.4}$$

This leads to the sub-pixel position of R_{max} to be:

$$\begin{aligned} x_{\max} &= -d_1/2d_2 \\ y_{\max} &= -e_1/2e_2 \end{aligned} \tag{4.5}$$

Figure 4.3 illustrates the matching process using normalised cross-correlation. The location of the point of interest in image 1 defines the centre of the template patch. The template patch is shifted over the search space in image 2, and the correlation coefficient is calculated at each location by equation (4.2). In the resulting 3-D plot the correlation value is indicated on the z-axis, with the x-y axes corresponding to the x-y location in the second image. At the position of the maximum correlation equation (4.4) is used to

approximate the continuous function of R_{ts} in order to estimate the sub-pixel location of R_{max} by equation (4.5). The resulting coordinates x_{max} and y_{max} are also the coordinates of the correspondence in image 2.

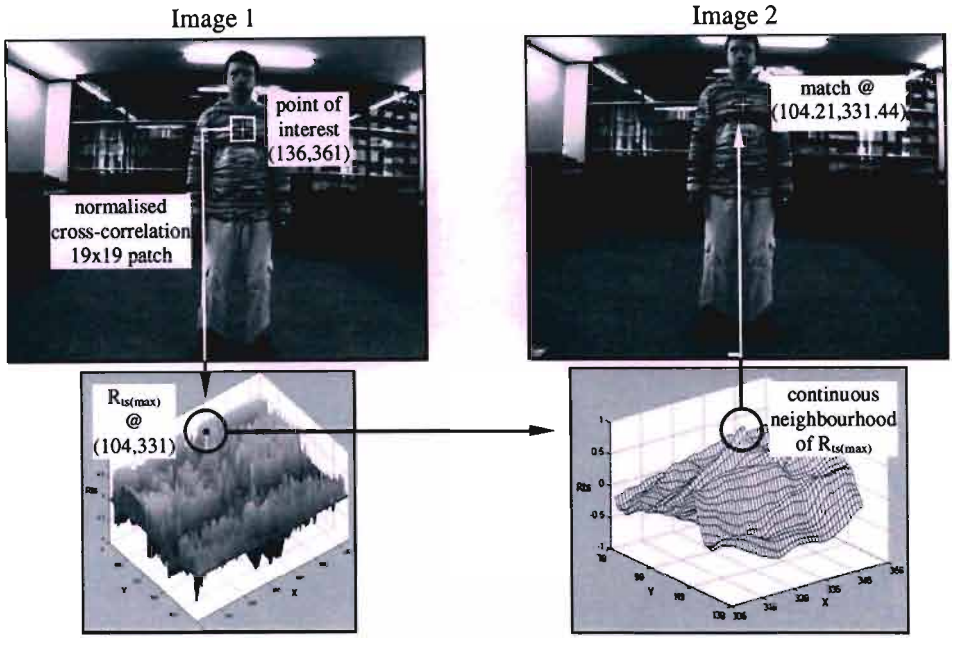


Figure 4.3: Matching points in a stereo pair using normalised cross-correlation.

The 3-D plot of the correlation coefficient in Figure 4.3 draws attention to the drawbacks of cross-correlation matching. The images in Figure 4.3 have a resolution of 640×480 pixels. The template patch was 19×19 pixels, thus the correlation coefficient had to be calculated 287364 times, making this process time consuming and computationally expensive. Furthermore, aside from the peak corresponding to the final result, the 3-D plot displays a number of other peaks, intuitively leading one to suspect that false matching may occur. This is indeed true, particularly when images contain regular patterns and/or the difference between the two images is large enough to cause low correlation at the actual match location. A partial solution to these two problems is reducing the search space in the second image. In case of stereo images, the epipolar constraint may be used to restrict the search to a line effectively reducing the task from a 2-D to a 1-D problem (Figure 4.4). Alternatively, if matching is performed in images constituting a video sequence, *a priori* knowledge of motion may be employed in predicting the object's location in subsequent

frames. In this way, the search space is reduced from the entire image to an area covering the uncertainty of the prediction.

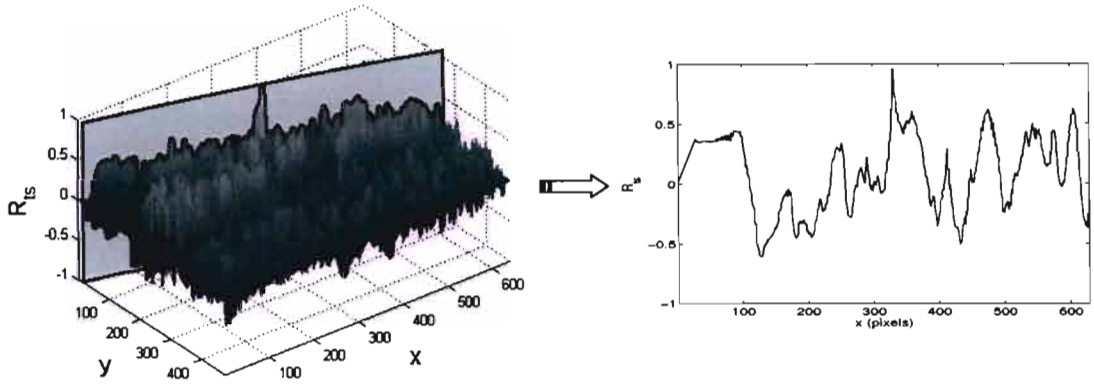


Figure 4.4: Using epipolar geometry to reduce the search space from a 2-D to a 1-D problem.

Although search space reduction improves the performance of cross-correlation matching, the problem of geometric distortion between the two images still remains. Moreover, a different viewpoint or object motion results in changing illumination and reflectance conditions distorting the images radiometrically as well. Even though the weighting function w_i can be introduced, it reduces rather than accounts for the effects of geometric and radiometric distortions. Thus, even when equation (4.5) is used to acquire a sub-pixel match location, cross-correlation cannot be considered as a matching technique with consistent sub-pixel accuracy. Nonetheless, cross-correlation remains an effective coarse matching tool, with the correlation coefficient a very useful similarity measure.

4.3 Least Squares Matching

Least squares matching is a method introduced by Ackermann [126] and Gruen [127] to overcome the limitations of cross-correlation. Similarly to cross-correlation, point matching is performed by considering an area around the point of interest in the template patch. In contrast however the correspondence is not found by searching a defined region. Instead, an estimate of the match location in the second image defines the centre of the

search patch which is shifted by the matching process to the actual coordinates of the correspondence. The mathematical model incorporates parameters that account for the geometric and radiometric differences between the template and search patches, providing least squares matching with the ability to produce results with high sub-pixel accuracy. The geometric relationship between the two patches can be described by transformations of varying degrees, depending upon the desired precision. Both Ackermann [126] and Gruen [127] have concluded that a 6 parameter affine transform is sufficient, particularly when small patches are used (e.g. 32x32 pixels or smaller). Subsequently, the affine transform has been used in other works involving least squares matching [46] [128] [129] [130] [131] [132] [133]. Radiometric correction of the search patch is achieved by introducing an offset and gain parameter to model the brightness and contrast differences. Following a least squares procedure, the residual differences between the grey values of the template and search patch are minimised to produce a transformation that provides optimal matching in the least squares sense.

4.3.1 Mathematical Model

The following derivation of the least squares estimation model follows the one given by Gruen [127], with the improved radiometric model adopted from [134]. Given two images, a template patch is defined in image 1 and a search patch in image 2. The template and search patches are given as discrete two-dimensional functions $f(x, y)$ and $g(x, y)$ respectively whose origins are located at the centre of the patch (i.e. at the point of interest and at the estimated match location). In an ideal situation correlation is established when:

$$f(x, y) = g(x, y) \tag{4.6}$$

Due to noise present in one or both images, equation (4.6) does not hold, and an additive noise vector $e(x, y)$ is introduced leading to:

$$f(x, y) = g(x, y) + e(x, y) \tag{4.7}$$

which can be rewritten as:

$$f(x, y) - e(x, y) = g(x, y) \tag{4.8}$$

Equation (4.8) gives the least squares observation equation which models the function $f(x,y)$ with the function $g(x,y)$. The location of $g(x,y)$ within the search image needs to be approximated with respect to some initial estimate $g_0(x,y)$. In addition to finding the translation between the final location of $g(x,y)$ and $g_0(x,y)$, the affine transform given by equation (4.9) compensates for geometric differences by shaping the search patch to resemble the template patch.

$$\begin{aligned} x &= a_{11} + a_{12}x_0 + a_{21}y_0 \\ y &= b_{11} + b_{12}x_0 + b_{21}y_0 \end{aligned} \tag{4.9}$$

The affine transform is performed with respect to the origin of the search patch defined at the location of the initial estimate. It accounts for translation (a_{11},b_{11}), rotation (a_{12},a_{21},b_{12},b_{21}), scaling (a_{12},b_{21}) and shearing (a_{21},b_{12}). Adding r_0 and r_1 for radiometric correction, together with equation (4.9), equation (4.8) can be rewritten as:

$$f(x,y) - e(x,y) = r_1 g(a_{11} + a_{12}x_0 + a_{21}y_0, b_{11} + b_{12}x_0 + b_{21}y_0) + r_0 \tag{4.10}$$

Gruen linearises the non-linear observation equation (4.10) by expanding it into a Taylor series and keeping only the zero and first order terms as in equation (4.11).

$$f(x,y) - e(x,y) = g_0(x,y) + \frac{\partial g_0(x,y)}{\partial x}dx + \frac{\partial g_0(x,y)}{\partial y}dy + dr_0 + dr_1 g_0(x,y) \tag{4.11}$$

Differentiating equation (4.9) gives:

$$\begin{aligned} dx &= da_{11} + x_0 da_{12} + y_0 da_{21} \\ dy &= db_{11} + x_0 db_{12} + y_0 db_{21} \end{aligned} \tag{4.12}$$

and using simplified notation:

$$\begin{aligned} g_x &= \tfrac{\partial g_0(x,y)}{\partial x} \\ g_y &= \tfrac{\partial g_0(x,y)}{\partial y} \end{aligned} \tag{4.13}$$

equation (4.11) becomes:

$$\begin{aligned} f(x,y) - e(x,y) = g_0(x,y) + g_x da_{11} + g_x x_0 da_{12} + g_x y_0 da_{21} \\ + g_y db_{11} + g_y x_0 db_{12} + g_y y_0 db_{21} + dr_0 + dr_1 g_0(x,y) \end{aligned} \tag{4.14}$$

The parameters in equation (4.14) are grouped into the parameter vector $\mathbf{x}$, the coefficients into the design matrix $\mathbf{A}$, and the vector difference $f(x,y) - g(x_0,y_0)$ into $\mathbf{l}$ according

to (4.15).

$$\begin{aligned}
\mathbf{x}^T &= [da_{11}, da_{12}, da_{21}, db_{11}, db_{12}, db_{21}, dr_0, dr_1] \\
\mathbf{A} &= \begin{bmatrix}
{}^1g_x & {}^1g_x \cdot {}^1x_0 & {}^1g_x \cdot {}^1y_0 & {}^1g_y & {}^1g_y \cdot {}^1x_0 & {}^1g_y \cdot {}^1y_0 & 1 & {}^1g_0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
{}^ng_x & {}^ng_x \cdot {}^nx_0 & {}^ng_x \cdot {}^ny_0 & {}^ng_y & {}^ng_y \cdot {}^nx_0 & {}^ng_y \cdot {}^ny_0 & 1 & {}^ng_0
\end{bmatrix} \\
\mathbf{l}^T &= \begin{bmatrix} {}^1f - {}^1g, & \cdots & ,{}^nf - {}^ng \end{bmatrix}
\end{aligned} \tag{4.15}$$

where:

${}^ix, {}^iy$ are the local coordinates of pixel i in the image patch

${}^ig_x, {}^ig_y$ are the discrete horizontal and vertical gradients at pixel i

${}^if, {}^ig$ are the grey values of the pixel i in the template and search patch

n is the total number of pixels in a $N \times M$ image patch

Using (4.15) the observation equations are obtained in the classical notation:

$$\mathbf{l} - \mathbf{e} = \mathbf{A}\mathbf{x} \tag{4.16}$$

and assuming:

$$E(e) = 0,\ E(ee^T) = \sigma_0^2 P^{-1} \tag{4.17}$$

the system is a Gauss-Markov model. The least squares estimation of equation (4.16) leads to the solution vector $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}} = (\mathbf{A}^T\mathbf{P}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{P}\mathbf{l} \tag{4.18}$$

with:

$\mathbf{v} = \mathbf{A}\hat{\mathbf{x}} - \mathbf{l}$	residual vector	(4.19)
$\hat{\sigma}_0^2 = \frac{\mathbf{v}^T\mathbf{P}\mathbf{v}}{r}$	variance factor	
$\mathbf{Q} = \hat{\sigma}_0^2 \cdot (\mathbf{A}^T\mathbf{P}\mathbf{A})^{-1}$	covariance matrix	
$\hat{\sigma}_i^2 = \mathbf{Q}_{ii}$	variance factor of single parameters	
$r = n - u$	redundancy	
n	number of observations	
u	number of transform parameters	
$\mathbf{P}$	weight matrix	

The weight matrix $\mathbf{P}$ is usually set to identity because typically the weight of each pixel in the patch is the same. It is in situations such as edge matching that $\mathbf{P}$ provides the flexibility to favour edge pixels by setting the corresponding weights depending on the value of the gradients at that point.

The functional model is non-linear and the final solution has to be found iteratively. The solution vector $\hat{\mathbf{x}}$ contains incremental updates of the affine parameters that describe the geometric difference between the template and search patch. The transform parameters are initialised with values $a_{11} = a_{21} = b_{11} = b_{12} = 0$ and $a_{12} = b_{21} = 1$ (i.e. initialised with values that do not alter the image patch). Subsequently, after each iteration these values are updated according to equation (4.20) and the search image is transformed.

$$p_1 = p_0 + dp \tag{4.20}$$

where p denotes an affine parameter and dp the incremental update.

The $\mathbf{A}$ matrix and $\mathbf{l}$ vector must be re-evaluated from the transformed image. This implies interpolation of the grey values from the transformed search patch, a process referred to as resampling, discussed in Section 4.3.2. The iteration stops when the shaping of the search patch becomes insignificant, indicated by low magnitudes of the update parameters or a small change in σ_0.

4.3.2 Computational Aspects

Section 4.3.1 presented the mathematical model and theoretical explanation of least squares matching. In this section the focus shifts to the practical implementation aspects and the decisions that directly affect the convergence rate, robustness and accuracy of the matching process.

Grey Level Derivatives

The grey level derivatives are computed from the discrete pixel values of the image patch. It is assumed that surrounding border pixels are available. The derivatives g_x and g_y at

pixel (i, j) are given by equations (4.21) and (4.22) which average the two neighbouring slopes.

$$\begin{aligned} g_x &= \frac{(g(i,j+1) - g(i,j))/\Delta x + (g(i,j) - g(i,j-1))/\Delta x}{2} \\ &= \frac{g(i,j+1) - g(i,j-1)}{2\Delta x} \end{aligned} \tag{4.21}$$

$$\begin{aligned} g_y &= \frac{(g(i+1,j) - g(i,j))/\Delta y + (g(i,j) - g(i-1,j))/\Delta y}{2} \\ &= \frac{g(i+1,j) - g(i-1,j)}{2\Delta y} \end{aligned} \tag{4.22}$$

where:

$\Delta x, \Delta y$ are the distance between two neighbouring pixels, equal to 1

i,j are the row and column indices respectively

Computation of Design Matrix A

The design matrix **A** was introduced in equation (4.15). Its elements are dependant on the image gradients defined by equations (4.21) and (4.22), and collectively embody the information used to determine the geometric relationship between the conjugate patches. The formulation and treatment of **A** thus has a direct effect on the accuracy, robustness and rate of convergence of matching. Baltsavias [134] has investigated the computational aspects of **A**, arriving at three viable variations:

(a) Computation of **A** from the search patch (**A** is updated in each iteration)

(b) Computation of **A** from the template patch (**A** remains constant throughout the iterations)

(c) Computation of **A** from the average of the template and search patch (**A** is updated in each iteration)

In the case of noise free, synthetic image matching, all three alternatives will arrive at the same result. According to the mathematical model derived in Section 4.3.1, option (a) is

the theoretically correct method and should yield the most accurate correspondences. Option (b) computes **A** from the template patch, thus bypasses the necessary recomputation of **A** after each iteration resulting in faster matching. Option (c) generates an additional patch from which **A** is computed by averaging the template and the search patch. The effect of this intermediate step is that the differences between the conjugate patches caused by noise and distortions are smoothed, thereby aiding the convergence of the transform parameters. The results of the investigation performed in [134] show that in the case of planar surfaces with good contrast and low noise the three variations indeed exhibit very similar performance. However, when larger distortion and noise is introduced, option (c) displays a more stable and faster convergence. In terms of accuracy, the final locations obtained by (c) were similar to those obtained by (a) (the difference being smaller than the x and y shift convergence criterion), and considerably better than those obtained by (b). Additionally, by smoothing the differences between the conjugate patches, option (c) is able to handle worse initial estimates than options (a) and (b).

In conclusion, when planar surfaces with low noise are being matched option (b) is preferred purely because it is faster. However this is not the case in sequences used in this work, and subsequently option (c) was adopted because it finds the correct solution even in situations when the other two methods fail.

Optimal Matching Window Size

The matching window size, i.e. the size of the image patch, is a significant parameter of area-based matching algorithms. The patch is a rectangular array of grey values symmetrical about the pixel position of the target point or the candidate matching point. The question now arises, what is the optimal window size for least squares matching. Unfortunately there is no straight forward answer because the patch size depends on factors specific to each case such as the objects present in the image as well as their texture.

The size and shape of the patch will ultimately determine the reliability and accuracy of the matching result. It must adapt to both the radiometric content of the image as well as the geometric form of the object. The patch must be large enough to contain sufficient

signal (texture) in order to determine all the affine parameters, but small enough for the assumption of a locally planar object surface to hold. The determinability of the affine parameters is dependant on the grey level gradients in the patch, their size as well as their distribution [134]. If the patch is too small, some of the shaping parameters may not be recovered, resulting in insufficient information for the similarity measurement. On the other hand, when the patch is too large, the assumption of the planar surface portion may no longer be valid, and the geometric deformation and extra features further away from the patch origin may affect the matching process and matching quality. Thus, the task is to select the smallest patch that allows for stable convergence of the affine parameters.

The shape of the patch can adapt to the matching feature to maximise the useful signal content used for the estimation of the affine parameters. An example of this is found in photogrammetry when matching features such as roads, in which case rectangular patches are preferable. Photogrammetry applications often have the advantage of being free to select the matching features, thus a selection process may be performed to select points lying along edges, improving matching success and precision [134]. Unfortunately, in the case of human motion capture, the key points were determined by the joint location rather than the presence of texture. Since there was no guarantee of long edges in either direction, the shape of the patches is always kept square.

The optimal size of the patch is determined experimentally depending on the content of the images used. Since the variation in applications of LSM is large (e.g. close range up to aerial images), it is difficult to establish even a guideline. Ackermann [126] has used patches of 64×64 pixels and more. Baltsavias [134] states that patches smaller than 7×7 to 9×9 pixels should be avoided and proceeded to use 15×15 patches in most of his experiments, rarely using patches greater than 30×30. Xin [135] also found 15×15 patches produced best results in his application. Rosenholm [136] conducted an investigation into the effect of window size on precision and reliability, concluding that patches of 20×20 and 30×30 pixels are optimal for precision. An optimal patch is large enough to contain sufficient signal for the determination of the geometric relationship between the conjugate patches, yet small enough to represent a planar surface. The optimal size in this work was found experimentally in matching scenarios typical of those faced in human motion

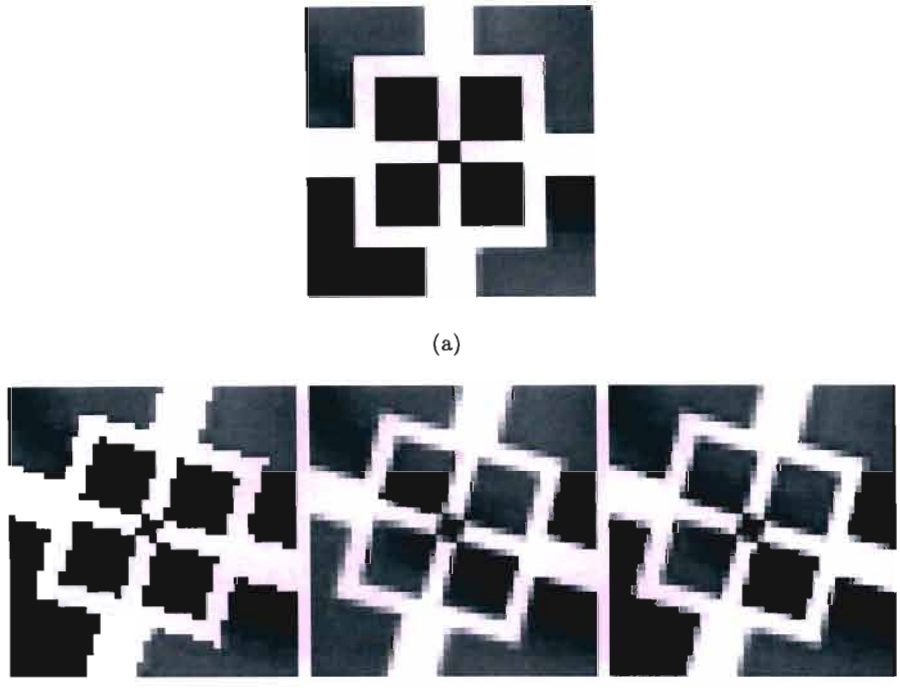
tracking, with a 19×19 patch producing most consistent results.

Resampling

Resampling is the process of transforming a sampled image from one coordinate system to another. The least squares matching procedure iteratively estimates the affine mapping function. After each iteration the original search patch is transformed by the updated affine transform to yield a patch increasingly resembling the template. In the subsequent explanation the original search patch is referred to as the *input image* and the transformed patch as the *output image*. The transformation itself is performed on the coordinate system of the input image, resulting in new coordinates for the grey values which are no longer restricted to an integer lattice. The output image is obtained by first projecting the output resampling grid into the input image by the inverse mapping function (i.e. the inverse affine transform) and then sampling the input image at these points to assign grey values to their respective output pixels. However, the projected resampling grid will generally not correspond to the input sampling grid and the grey values must be interpolated from the input image at the desired locations.

The choice of a suitable interpolation function usually involves a tradeoff between accuracy and efficiency. The simplest and fastest interpolation is the nearest neighbour algorithm which assigns a value to an output pixel of the nearest sample point in the input image. Bilinear interpolation considers the four neighbouring points around the sample location and assumes that the brightness function is bilinear in the neighbourhood, while bicubic interpolation considers a neighbourhood of sixteen points assuming that the brightness function is bicubic. Higher order functions can also be used, however will not be discussed in the scope of this work. An example of the three above mentioned methods is shown in Figure 4.5, where an affine transform was used to rotate the input image (Figure 4.5(a)) by 15 degrees. Once the output sampling grid was projected back into the input image, the output image was resampled by nearest neighbour interpolation (Figure 4.5(b)), bilinear interpolation (Figure 4.5(c)) and bicubic interpolation (Figure 4.5(d)). Nearest neighbour interpolation suffers from severe loss of quality and since it may introduce errors of up to a

half-pixel it is not suitable for problems requiring sub-pixel accuracy. Bilinear interpolation displays considerably improved results, with the problem of step like straight boundaries significantly reduced. The bicubic interpolation has less of a blurring effect than bilinear interpolation, however the difference in image quality is less pronounced and comes at a higher computational cost [137].



(a)

(b) (c) (d)

Figure 4.5: Resampling a transformed image using different interpolation methods, (a) original image, (b) nearest neighbour, (c) bilinear interpolation, (d) bicubic interpolation.

Ackermann has originally applied sophisticated interpolation techniques in his implementation of least squares matching, however through experimentation with various methods he concluded in [126] that bilinear interpolation provides sufficient results for accurate matching. The bilinear interpolation is give by equation (4.23):

$$\begin{aligned} I(r,s) = I_o(i,j) + (I_o(i,j+1) - I_o(i,j)) \cdot dx + (I_o(i+1,j) - I_o(i,j)) \cdot dy \\ +(I_o(i+1,j+1) + I_o(i,j) - I_o(i,j+1) - I_o(i+1,j)) \cdot dxdy \end{aligned} \tag{4.23}$$

where:

r,s are real

i,j are integers with $i = int(r)$, $j = int(s)$

$dx = s - j$, $dy = r - i$

Figure 4.6 shows the output resampling grid (broken lines) projected into the original image whose sampling grid is indicated by solid lines. The grey value of the output pixel is computed from the four pixels of the input image neighbouring the projected sampling location.

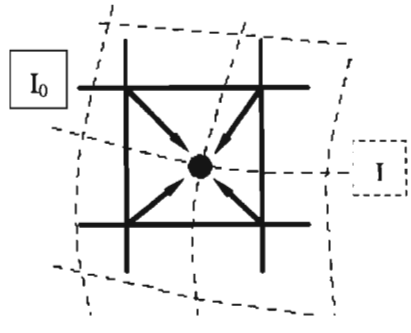


Figure 4.6: Bilinear interpolation considers the four neighbouring points of the input sampling grid I_0 around each point of the projected output sampling grid I.

4.3.3 Typical Results

The following two experiments demonstrate the least squares matching process. First, synthetic images are used to clearly illustrate the shaping of the search patch during the iterative stage, as well as to provide ground truth data to assess the matching performance. The second example is performed with real images from a stereo setup. In this case ground truth is not available and parameters used for assessing the quality of the result are discussed.

Synthetic Data

The experiment with synthetic data makes use of a simple template image consisting largely of a black circle on a white background, which provides well defined edges and consequently large gradients whose presence improves matching stability. The second view, the search image, was generated by transforming the template image by a known affine transform defined by equation (4.24) and subsequently resampling using bilinear interpolation. The two synthetic images are shown in Figures 4.7(a) and 4.7(b).

$$\begin{aligned} \begin{bmatrix} x_s \\ y_s \end{bmatrix} &= \begin{bmatrix} a_{11} \\ b_{11} \end{bmatrix} + \begin{bmatrix} a_{12} & a_{21} \\ b_{12} & b_{21} \end{bmatrix} \begin{bmatrix} x_t \\ y_t \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \end{bmatrix} \end{aligned} \tag{4.24}$$

In comparison to the real data encountered in human motion tracking, the synthetic images differ in three ways:

1. the synthetic images represent perfectly planar surfaces
2. the presence of such well defined edges is uncommon in markerless human motion capture
3. no noise was added to the search image aside from the distortion introduced by the irreversible blurring effect of bilinear interpolation

The overall effect of these factors on the matching result is that for the synthetic data experiment the achieved accuracy is unrealistic in comparison to real data. However, the experiment is beneficial in that it is able to visualise the least squares matching process, as well as to indicate the theoretically achievable precision.

Before initiating the matching process, the suitable patch size must be determined. Section 4.3.2 stated two aspects affecting the size of the patch (matching window): the patch must be small enough to represent a locally planar surface and the patch must be large enough to contain sufficient signal to allow stable and accurate determination of the six

affine parameters. In this case the former constraint is always fulfilled because the whole image represents a planar surface. The latter constraint can thus be easily satisfied as the maximum patch size is not limited apart from the dimensions of the actual image. The *"sufficient signal"* refers to the regions of high value gradients which for the template image are encapsulated by a 69×69 patch. Figure 4.7(c) shows the shaping of the search patch through five iterations. By visual inspection there seems to be little change after the third iteration and by the fifth iteration the search image seems identical to the template.

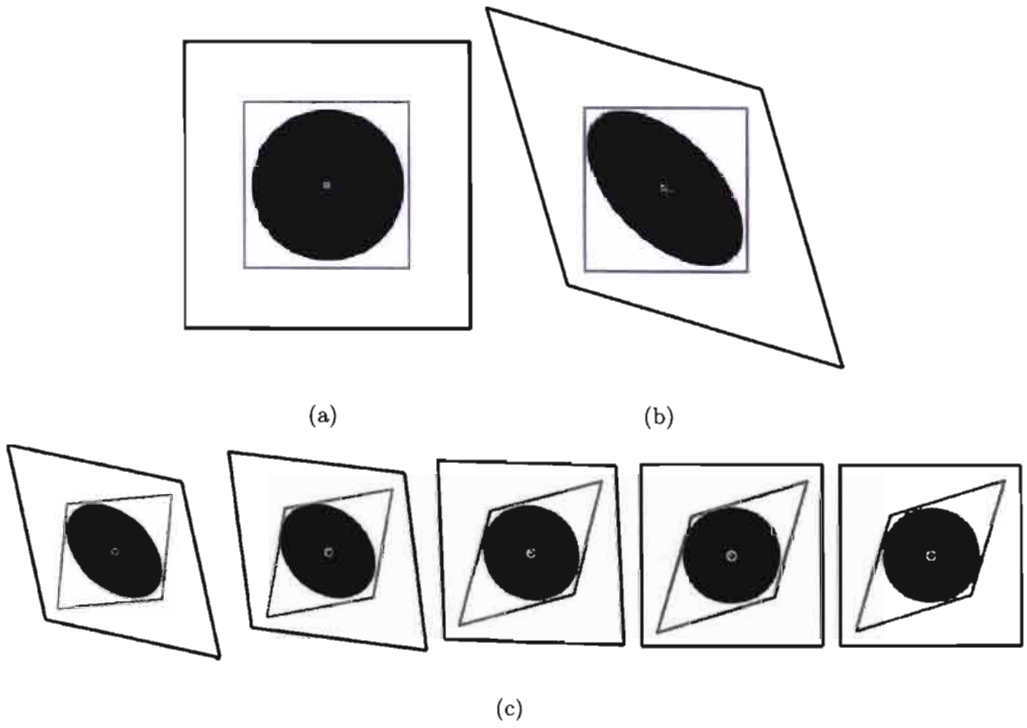


Figure 4.7: LSM using synthetic data, (a) template image with patch, (b) search image with patch, (c) search image and patch deformation after iterations 1 to 5.

To evaluate the matching result quantitatively, one must look at the values of the updates, the resulting affine parameters and the pixel error of the final result. The shaping of the search image in Figure 4.7(c) occurs because the matching process incrementally updates the estimate of the affine transform describing the geometric difference between the conjugate images. Figures 4.8(a) and 4.8(c) show the behaviour of the x and y coordinate updates of parameters in equation (4.9) respectively. Ideally the system converges when

the updates tend to zero as is the case from iteration 7 onwards. In practice however, suitable thresholds are selected for the magnitudes of the updates to establish convergence. The typical values of 0.05 for the shifts ($|\Delta a_{11}|, |\Delta b_{11}|$) and 0.1 for the shaping parameters ($|\Delta a_{12}|, |\Delta a_{21}|, |\Delta b_{12}|, |\Delta b_{21}|$) used in [134] were also adopted in this work. These thresholds are indicated in Figures 4.8(a) and 4.8(c) by dotted lines, according to which convergence occurred after the 5th iteration. Figures 4.8(b) and 4.8(d) display the x and y coordinate affine parameters converging onto the ground truth indicated by dotted lines.

Table 4.1: Comparison of ground truth and matching results (after 5 iterations)

parameter	ground truth	measurement	% error
a_{11}	1.0000	0.9991	0.09
a_{12}	1.0000	0.9993	0.07
a_{21}	0.3000	0.2982	0.60
b_{11}	1.0000	0.9963	0.37
b_{12}	0.3000	0.3016	0.53
b_{21}	1.0000	0.9985	0.15

The accuracy of the affine parameters estimation is more clearly expressed in Table 4.1. The purpose of recovering the affine parameters is to align the template and search patch coordinates of corresponding pixels by shifting and warping the search patch coordinate system. This results in the origin of the search patch coordinate system that is initially defined at the location of the estimate to be shifted to the correct location of the correspondence. Intuitively, as the estimation of the affine parameters improves, the resemblance between the patches becomes more evident and the accuracy of the result increases. This relationship is supported by Figures 4.8(e) and 4.8(f), which show the cross-correlation coefficient and the pixel error after each iteration respectively. As the similarity between the template and search patch increases, the error of the match location decreases. The final matching result lies at the origin of the search patch and hence is only dependant on the x and y shifts a_{11} and b_{11} respectively. These two parameters were estimated with

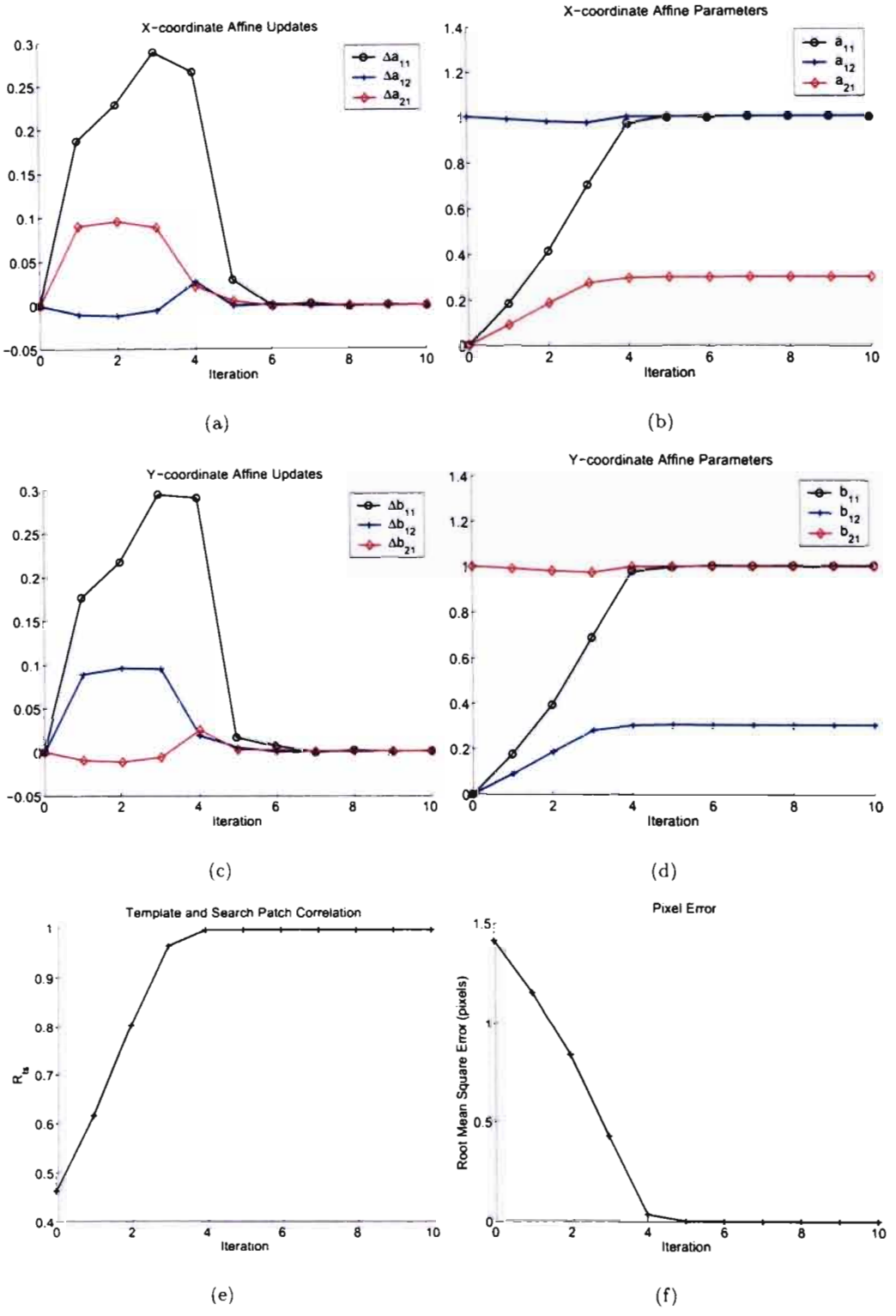


Figure 4.8: Synthetic data matching results.

errors of 0.09% and 0.37% resulting in precision of 1/250 pixel. The high accuracy can be attributed to the three aspects discussed at the beginning of this section, which set the synthetic images apart from the typical real data encountered in this research. Ackermann indicates in [126] that precision of 1/180 pixel is possible for aerial photographs with high contrast, however in low contrast situations it drops down to 1/10 pixel. Baltsavias [134] achieved accuracy of 1/10 to 1/50 pixels in close range experiments and reports that the maximum attainable accuracy in real images lies in the range 1/20 to 1/100 pixels. Lemmens [124] establishes LSM accuracy to be 1/20 pixel which is a representative value when comparing LSM to other methods and not considering specific cases.

Real Data

The experiment with real data was performed on images constituting a stereo pair. Portions of these images are shown in Figure 4.9 together with the corresponding patches which have been enlarged for the sake of clarity. The image content is typical of the captured data for the human motion capture system in this work.

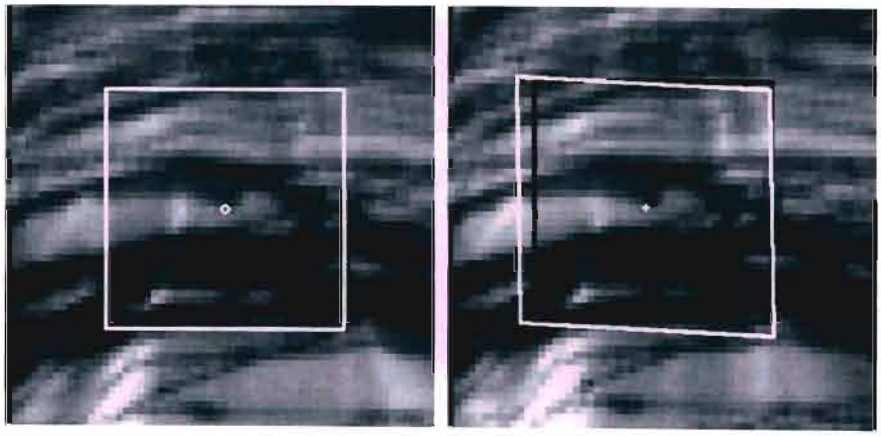


Figure 4.9: Left (template) and right (search) view of a stereo system. The original patch in the search image is shown in black, and the final deformed patch in white.

The multi-view camera setup is such that the baseline between the cameras is relatively short resulting in small geometric differences and consequently similar images in all views. This leads to an argument that a simpler transform could be used to describe the geometric

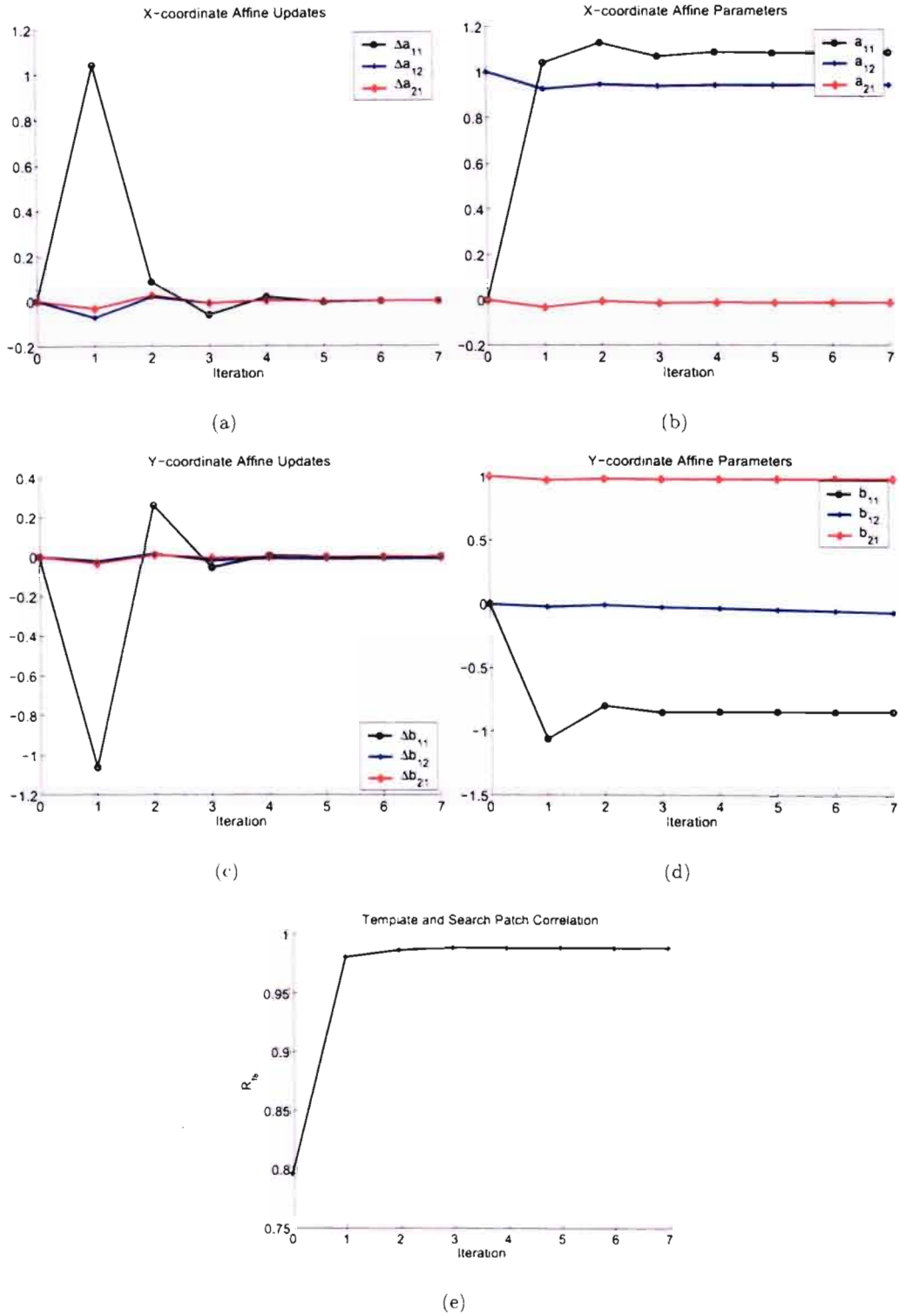


Figure 4.10: Real data results.

relationship between the conjugate images. Ackermann [126] has investigated this possibility by experimenting with a 4-parameter similarity transform. The experiments were performed on aerial images and results indicated that in areas of flat ground (i.e. planar patches) the 6-parameter affine transform did not yield considerably better results. However, Ackermann concluded that the affine transform should still be used as a safeguard.

The experimental results are shown in Figure 4.10, where 4.10(a) and 4.10(c) display the updates together with the convergence criterions (thresholds), 4.10(b) and 4.10(d) display the estimated affine parameters and 4.10(e) displays the cross-correlation between the two patches. Indeed the affine parameter estimation process is dominated by shifts and scales, with only small shearing occurring. The relatively high similarity between the images results in fast convergence (4 iterations).

In the absence of ground truth, the confidence in the result is assessed by analysing parameters produced as a consequence of the matching process. These parameters include the cross-correlation coefficient R_{ts} calculated by equation (4.2) and the standard deviations σ_0, σ_x and σ_y obtained from equation (4.19). R_{ts} is a useful indicator of the matching quality because it quantifies the similarity between the two patches. Since the matching process tries to minimise the residual differences between the patches, a high value R_{ts} suggests quality estimation of the transform function parameters and consequently good matching results. The standard deviation σ_0 is an *a posteriori* estimator for the difference in grey levels between the template and search patch. It is highly correlated with the cross-correlation coefficient. Serving as an indicator of the error between the patches, σ_0 decreases with increasing similarity. The standard deviations of the x and y shifts, σ_x and σ_y ($\sigma_x = \sigma_{a_{11}}$, $\sigma_y = \sigma_{b_{11}}$) are of particular interest because the two shifts are ultimately responsible for the accuracy of the final result. Large values of σ_x and σ_y indicate low precision in a_{11} and b_{11} and consequently low precision in the resultant match location.

The four parameters R_{ts}, σ_0, σ_x and σ_y are utilised in result assessment by comparing them to suitable thresholds. The difficulty of selecting the thresholds stems from the fact that characteristic values of these parameters are dependant on the image content and quality. The typical threshold criteria used in this work are displayed in Table 4.2

together with the corresponding values from the real data experiment. A high threshold for the cross-correlation coefficient is appropriate since the conjugate images begin the matching process with high similarity due to the 30 fps frame rate and centralised camera setup. The thresholds for σ_0, σ_x and σ_y have been adopted from D'Apuzzo [138] who found these values to be appropriate for markerless human motion capture.

Table 4.2: Indicators of quality of real data matching result

parameter	measurement	threshold
R_{ts}	0.9893	0.9000
σ_0	10.925	25.000
σ_x	0.0374	0.1800
σ_y	0.0159	0.1800

4.4 Summary

This chapter has presented two digital image matching techniques. Cross-correlation is a simple method in terms of implementation, however because it does not model the geometric and radiometric differences between the images it does not provide adequate accuracy required by the tracking algorithm discussed in Chapter 5. Furthermore, unless the search space is limited to a small area, cross-correlation is susceptible to false matching. Despite these deficiencies, cross-correlation has been widely used in the tracking algorithm as a similarity measure, as well as a coarse matching technique for initial estimate generation in a search space reduced by prediction or epipolar geometry.

The second matching technique introduced was least squares matching. Least squares matching models the geometric difference between images by a 6-parameter affine transform, and the radiometric difference by an additive and multiplicative coefficient. Least squares matching is an iterative process that can be summarised by the 11 steps shown below. A schematic representation of the algorithm is provided in Figure 4.11.

Initialisation:

(1) Define template patch

(2) Define search patch

(3) Initialise the affine transform

Iterate:

(4) Subtract search patch from template patch

(5) Construct $\mathbf{l}$

(6) Average template and search patch

(7) Calculate x and y gradients

(8) Construct $\mathbf{A}$

(9) Estimate affine parameter updates

(10) Update affine parameters

(11) Transform original search patch

repeat steps 4-11 until $|updates| < threshold$ or $\Delta\sigma_0 < threshold$

Experiments have demonstrated the high accuracy of least squares matching, within 1/250 pixel for synthetic data. Matching precision in real data close-range experiments is expected to lie between 1/10 and 1/50 pixel. In this work no ground truth data is available for real images, and the confidence in the result is assessed by looking at how well the affine parameters have been estimated. This is indicated by either the similarity or the error between the patches described by R_{ts} and σ_0 respectively. The final result is obtained by shifting the initial estimate to the correct location by parameters a_{11} and b_{11}. The precision of these parameters is indicated by σ_x and σ_y respectively, which are also employed in the analysis of the result. After the matching process is complete, the above mentioned parameters are compared to predefined thresholds, and if the relevant criteria are satisfied the estimated match location is accepted.

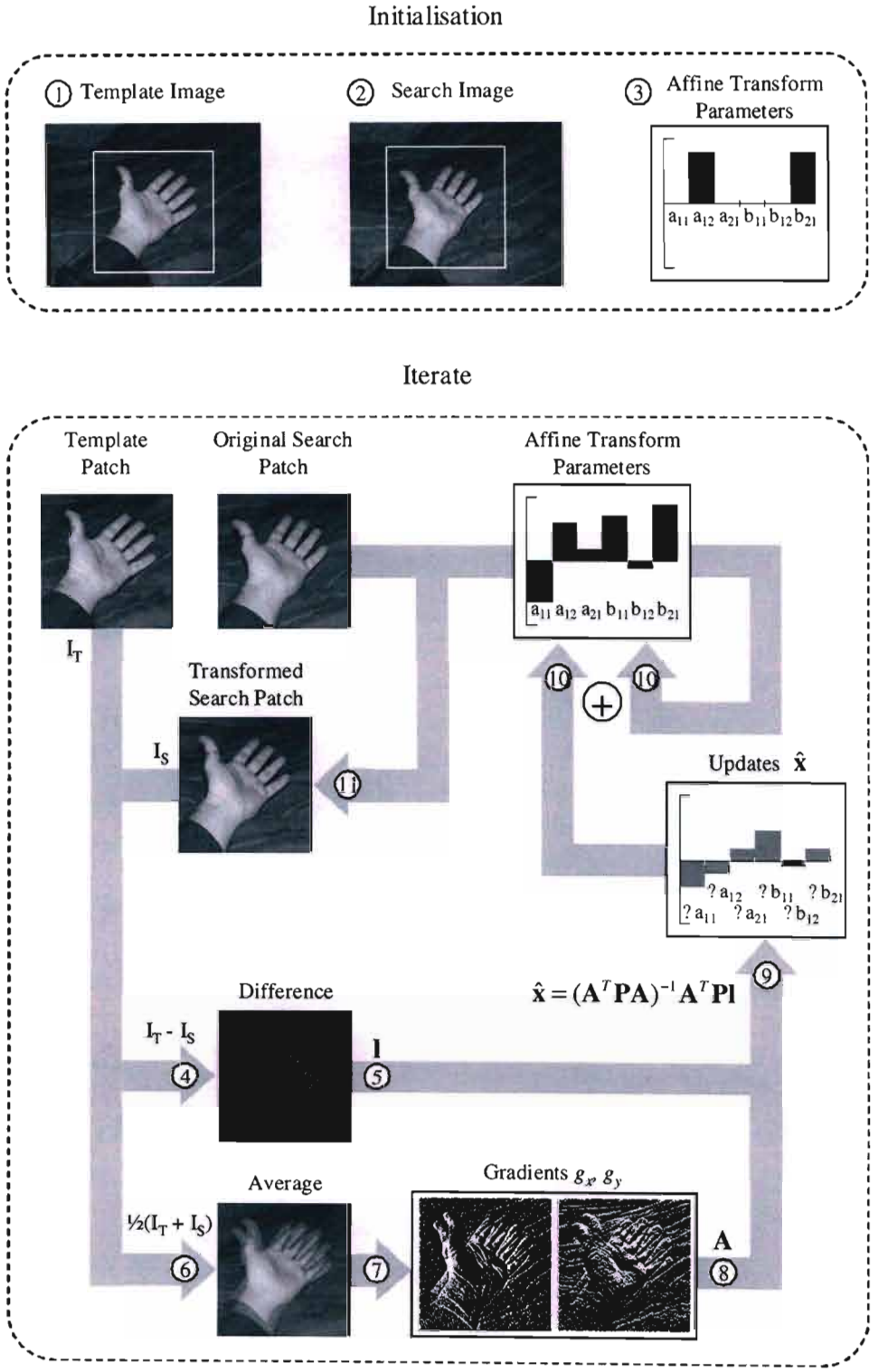


Figure 4.11: Visual overview of the least squares matching process.

The least squares matching technique is at the heart of the tracking algorithm discussed in Chapter 5. By automating the matching process, scrutinising the matching results and automatically tuning the matching parameters, the tracking algorithm tracks selected points throughout the sequence as well as within the multiple views with sub-pixel accuracy.

Chapter 5

Tracking Algorithm

The tracking algorithm developed in this work completes the human motion capture system required at the encoder stage of the model-based image coding scheme. Selected points are tracked in 3-D by automating the image matching process over multiple views. Following manual initialisation, tracking is performed automatically without any intervention from the user. Automation of the image matching process involves the introduction of a prediction scheme that generates initial estimates of correspondences for least squares matching. Furthermore, the least squares matching process is monitored to detect undeterminable affine parameters, to determine convergence, assess results and if necessary repeat matching with modified parameters.

This chapter starts by discussing the issues relating to the automation of the least squares matching process. First, the convergence radius, also called the pull-in range is investigated to establish the necessary accuracy of the prediction scheme. Then, tests for the detection of undeterminable parameters are proposed and methods of determining convergence and assessing results, already introduced in Chapter 4, are discussed further. The automatic least squares matching process is first applied to tracking in 2-D, following which the concept is extended to 3-D tracking. The advantages of multiple views and 3-D information are described and incorporated into the final tracker.

5.1 LSM Adaptation to Tracking

Figure 4.11 in Section 4.4 provides a schematic description of the least squares matching model introduced in Section 4.3. The process iteratively estimates the geometric and radiometric relationship between two images in order to yield the location of corresponding points. To utilise LSM in an automated system such as a 2-D/3-D tracker, several additional components have to be added as shown in Figure 5.1. The input to the system consists of two images and the locations of the point of interest and the initial estimate. The estimation of the affine parameter updates is performed according to Figure 4.11, which is represented in Figure 5.1 by the LSM box. In addition to the estimation of the incremental updates, the shaping parameters are examined to detect undeterminable cases, the convergence status is established by inspecting the behaviour of the estimation process and the quality of results is assessed using parameters introduced in Section 4.3.3. A failure of any one of the above conditions results in the matching process repeating

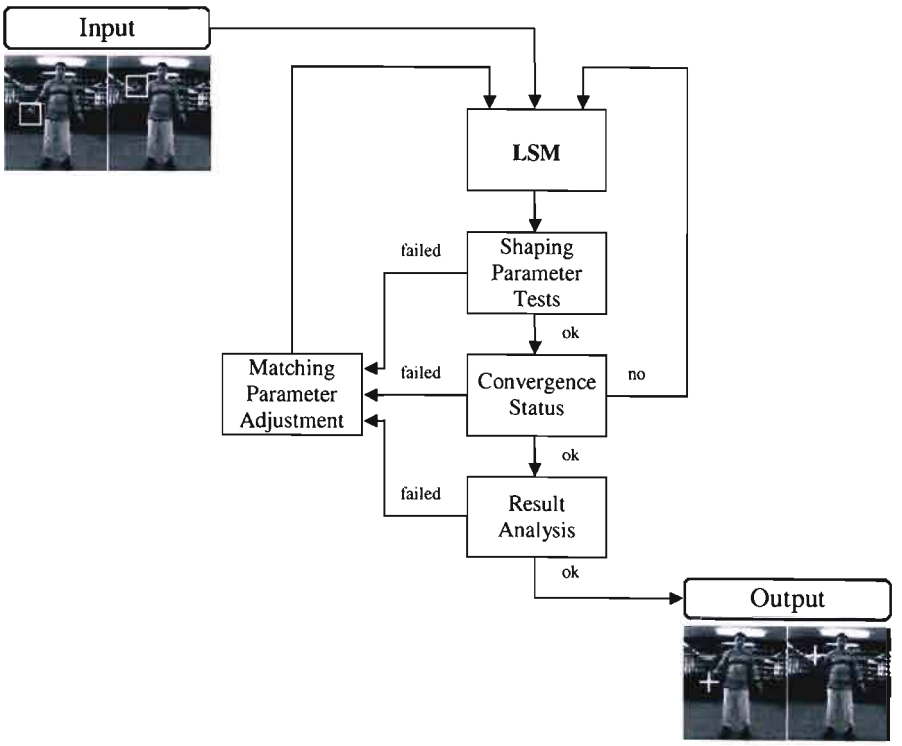


Figure 5.1: Information flow of the automated LSM module.

with modified parameters, thus ensuring a high rate of matching success in the tracking procedure.

5.1.1 Shaping Parameter Tests

The behaviour of the shaping parameters during the matching process is analysed in order to detect undeterminable cases. The issue of affine parameter determinability has already been touched upon in Section 4.3.2 which dealt with matching patch size. The patch must ideally be large enough to encompass enough image content to guarantee stable and accurate recovery of all affine parameters. In a practical tracking application the user cannot select the best suitable patch size for each matching case. On the contrary, the user can only select a patch size that provides good results in most cases, and the tracking algorithm must fine tune this parameter in those instances when matching does not yield a satisfactory result. Under these conditions, it is probable that throughout the process there will be situations when the patch does not contain enough information for LSM to determine all the affine parameters. The undeterminable parameters usually exhibit oscillating or diverging characteristics during the iterative estimation process. An example of these two cases is shown in Figure 5.2. The oscillation of the update and resulting oscillation of the transform parameter is displayed in Figures 5.2(a) and 5.2(b) respectively, and similarly in the case of divergence in Figures 5.2(c) and 5.2(d). The tests for oscillation and divergence are performed during the estimation process. Initially there may be a considerable difference between the patches causing large update changes, therefore the tests are carried out from the 3^{rd} iteration onwards once the system has begun to stabilise. Each shaping parameter update $(\Delta a_{12}, \Delta a_{21}, \Delta b_{12}, \Delta b_{21})$ is checked, and if its magnitude is greater than the convergence criteria threshold, it must satisfy the set of rules suggested in [134]:

- If in three consecutive iterations the update changes sign, its magnitude must decrease
- If an update does not change sign, its magnitude must be decreasing over the last three iterations

- If the update's magnitude decreases in two successive iterations, it is not allowed to increase
- The magnitude of the update must decrease in the last 2-3 iterations (performed at the end of the estimation process)

Failure to meet any one of the above criteria may lead to the corresponding parameter to be excluded from the mathematical model. Naturally, the two shifts (Δa_{11} and Δb_{11}) cannot be excluded without loosing accuracy and confidence in the result.

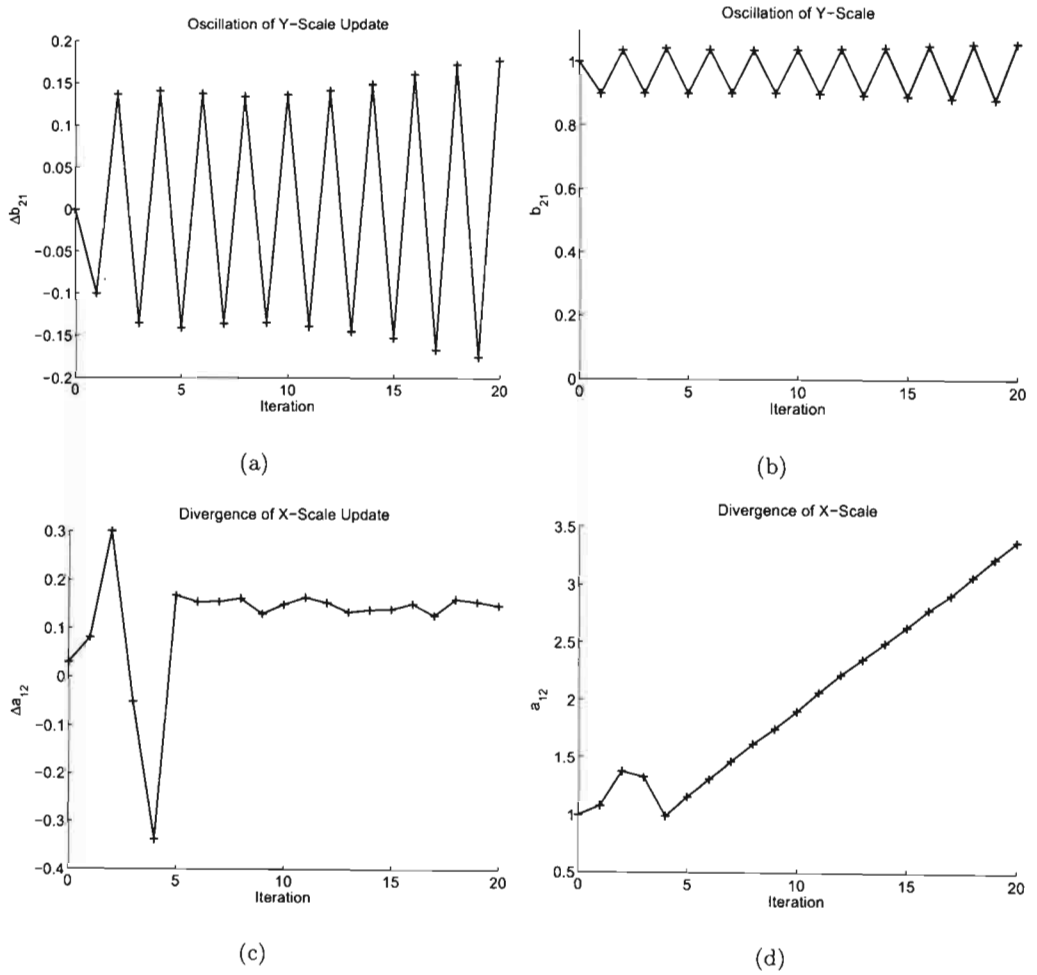


Figure 5.2: Examples of undeterminable parameters. Dotted lines indicate convergence thresholds of respective updates.

5.1.2 Convergence

LSM convergence is established when the shaping of the search patch becomes so small that it has no real effect on the matching result. Such a point in the estimation process may be indicated either by the updates as done in Section 4.3.3 or by the change in σ_0. In either case, the corresponding values are compared to pre-selected thresholds. In the case of the updates, convergence occurs only once the magnitudes of all the incremental changes fall below their thresholds. Figure 5.3 shows an example of successful convergence.

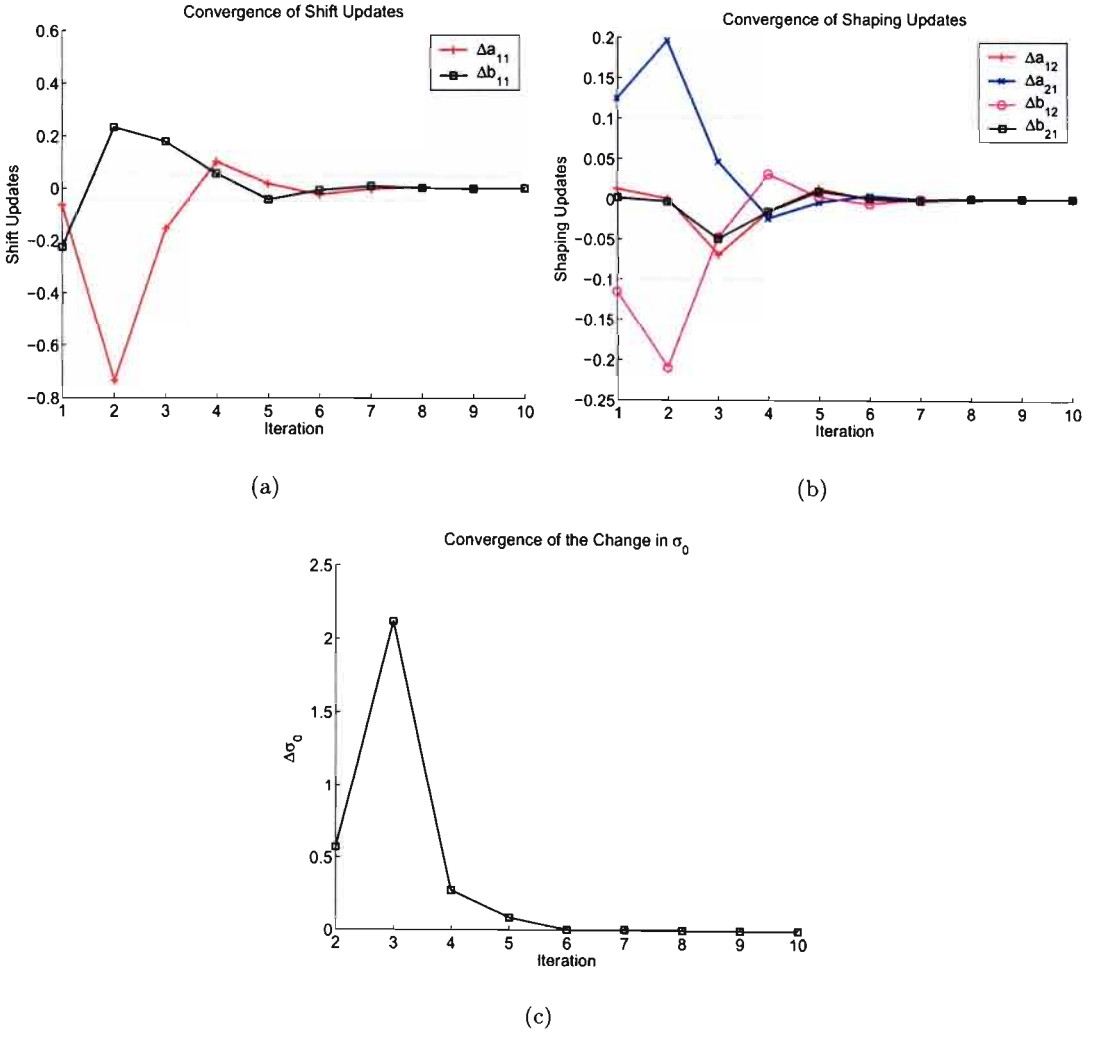


Figure 5.3: Determination of convergence by setting thresholds for updates (a), (b) and by setting a threshold for change in σ_0 (c).

Convergence determination by updates is shown in Figures 5.3(a) and 5.3(b) which display the incremental estimation of the shifts and shaping parameters respectively, together with their corresponding thresholds ($|\Delta a11| = |\Delta b_{11}| = 0.05, |\Delta a_{12}| = |\Delta a_{21}| = |\Delta b_{12}| = |\Delta b_{21}| = 0.1$). The behaviour of $\Delta\sigma_0$ and its threshold (equal to 0.02) for the same matching example is shown in Figure 5.3(c). The task is now to assess whether the chosen threshold values are sufficient for accurate results. Using the 1/20 pixel guideline accuracy of LSM as reported by Lemmens [124], the plot of the pixel error of the matching experiment of Figure 5.3 and the accuracy "threshold" was generated (Figure 5.4). Figure 5.4 shows that the desired accuracy was reached in the forth iteration. According to the updates, convergence was established in the fifth iteration, while $\Delta\sigma_0$ fell below its threshold in the sixth iteration. Thus it can be concluded that the thresholds for both criteria satisfy the accuracy demands. Experiments have shown that both methods of determining convergence produce very similar results, however the $\Delta\sigma_0$ criterion was preferred since only one parameter instead of six needs to be monitored.

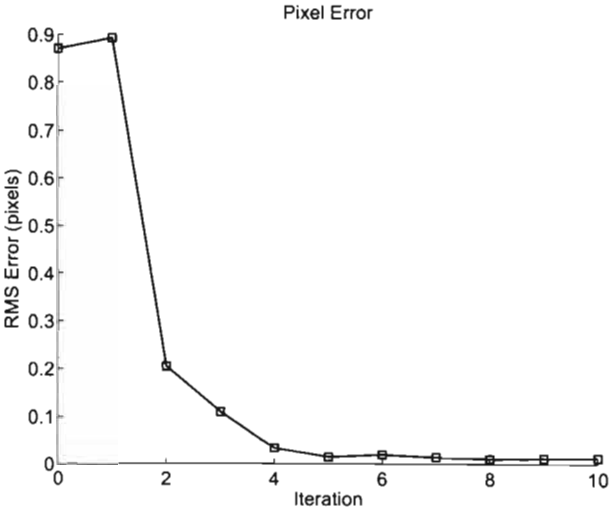


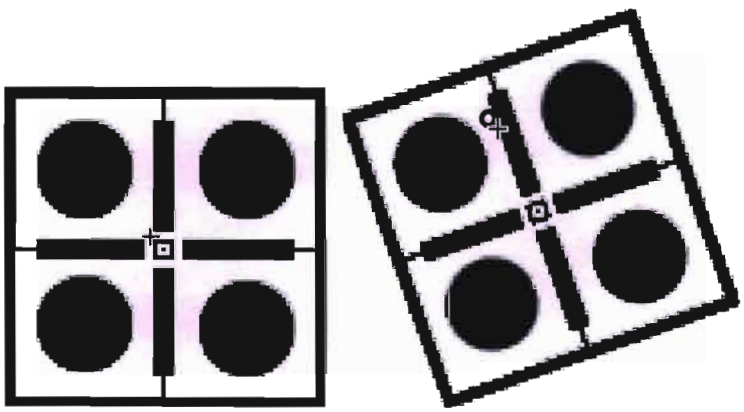
Figure 5.4: Pixel error of matching experiment used to generate convergence data for Figure 5.3.

The *"convergence status"* component of the automatic LSM algorithm (Figure 5.1) has three outputs; convergence not yet reached, convergence reached and convergence failed. The first two instances have already been covered by the discussion of convergence determi-

nation. Failure occurs when the convergence criterion is not satisfied within the maximum number of allowed iterations. In this work, the maximum number of iterations was set at 15. The temporal changes and spatial differences in the multi-view video sequences were relatively small resulting in an average convergence rate of 4-5 iterations. Thus, if LSM is unable to converge onto a solution in 15 iterations, it can be said with confidence that this case will not yield a solution with the particular matching parameters.

5.1.3 Result Analysis

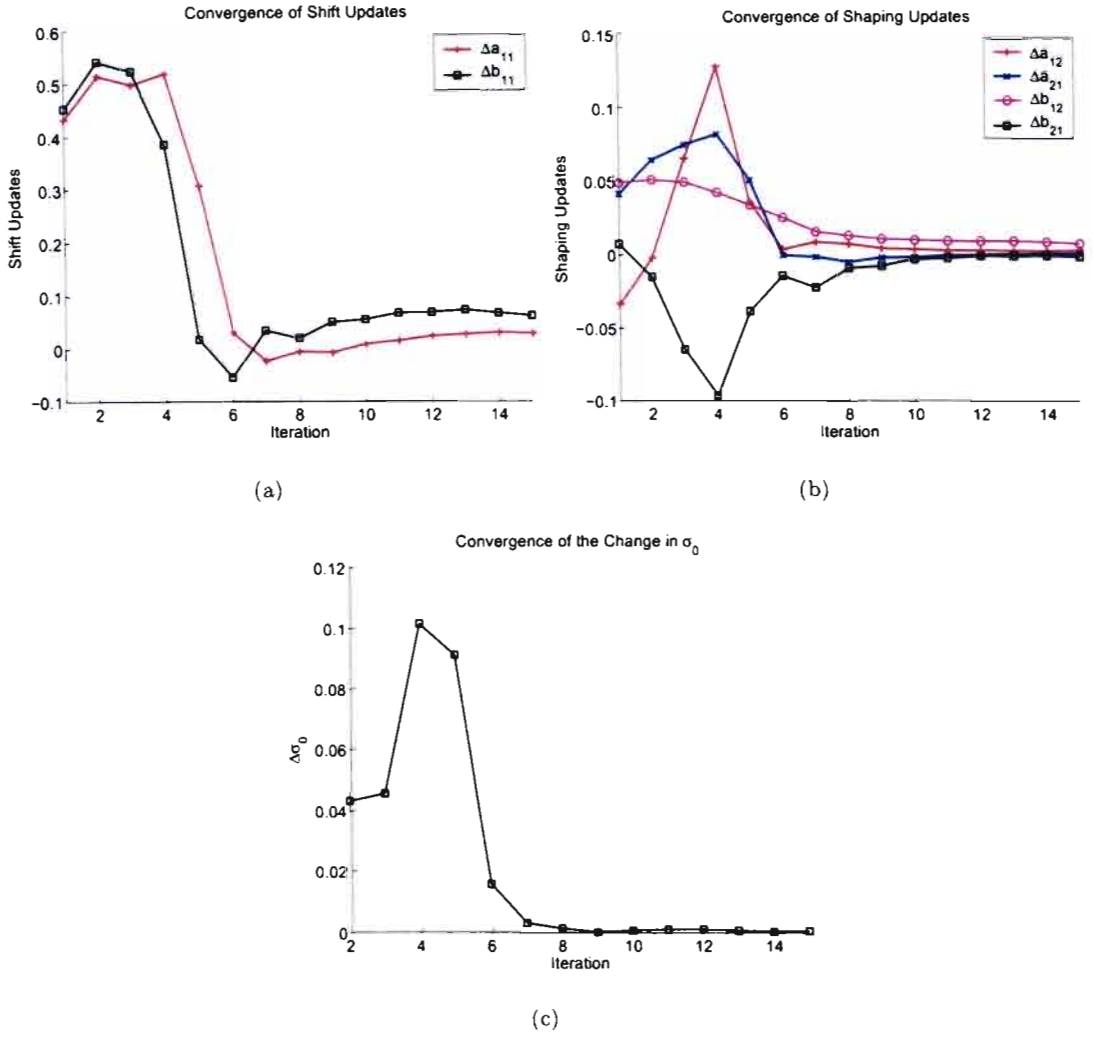
Convergence on its own does not guarantee correct results. Both methods of determining convergence merely monitor the shaping of the search patch, stopping the estimation process once significant transformation of the patch has ceased. The convergence tests cannot detect when LSM converges onto a wrong solution, hence the result needs to be analysed to establish whether it is acceptable or not. When working with real data, in the absence of ground truth, the parameters used to analyse the quality of the result are the cross-correlation coefficient R_{ts} and the standard deviations σ_0, σ_x and σ_y as introduced in Section 4.3.3. An experiment that demonstrates the importance of the result analysis uses images shown in Figure 5.5, where the point of interest lies at the location denoted



(a) (b)

Figure 5.5: Images demonstrating incorrect convergence, (a) template image with point of interest indicated by "+", (b) search image with initial estimate indicated by "o" and converged result by "+".

by a "+" in the template (Figure 5.5(a)), and the corresponding estimate in the search is denoted by an "o" (Figure 5.5(b)). The estimate is purposefully at a completely wrong location to ensure that the process does not converge onto the correct solution. Figure 5.6 displays the resulting updates in (a) and (b) and the change in standard deviation in (c).



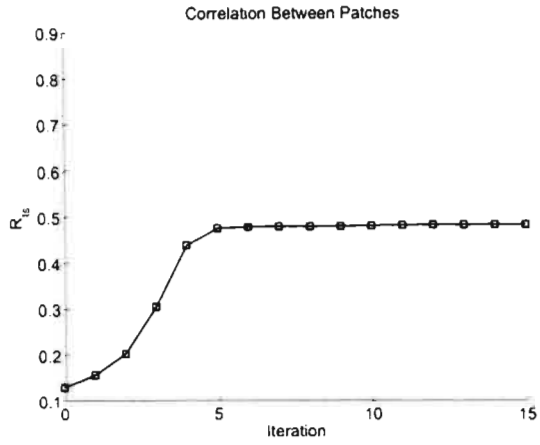
(a) (b)

(c)

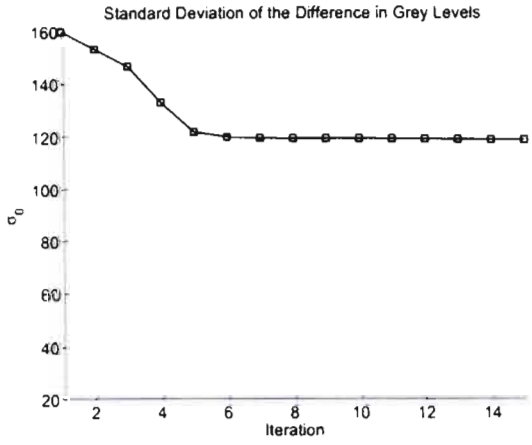
Figure 5.6: Convergence of the updates (a), (b) and the change in standard deviation (c). Thresholds are indicated by dotted lines.

Although the shift updates start to diverge as the iterations continue, according to the threshold criteria the systems has converged after 7 iterations in the case of the updates and after 6 iterations when σ_0 is used as the criterion. It is obvious to a human observer

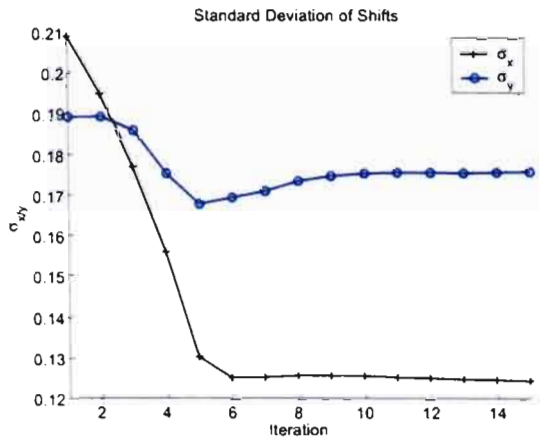
that the result denoted by "+" in Figure 5.5(b) is incorrect and the automatic tracking algorithm must also be able to make this judgement in order to successfully track points in video sequences.



(a)



(b)



(c)

Figure 5.7: Assessing the quality of result by (a) cross-correlation coefficient R_{ts}, (b) standard deviation σ_0, (c) standard deviations of the shifts σ_x, σ_y. Thresholds are indicated by dotted lines.

Figure 5.7 displays the cross-correlation coefficient, the standard deviation of the difference between the two patches (error) and the standard deviations of the shifts during the matching process. Both R_{ts} and σ_0 indicate a large difference between the patches and

consequently low similarity. Although the standard deviations of the shifts do fall below the threshold, the result would be rejected based on R_{ts} and σ_0. Ideally LSM transforms the search patch to look exactly like the template and thus such a low value of R_{ts} and conversely a high value of σ_0 are a clear indication of an incorrect convergence.

5.1.4 Adjustment of Matching Parameters

The matching parameters are modified when undeterminable parameters are detected, the system fails to converge or the result does not exhibit the desired characteristics of a good match. Once one of these three components generates an error, the automatic LSM algorithm enters the *"Matching Parameter Adjustment"* phase and adjusts either the patch size, the mathematical model (by excluding undeterminable parameters) or the initial estimate. The sequential process is demonstrated in Figure 5.8(a). The adjustment changes only one of the parameters at a time, then reinitialises and repeats the matching process. If matching continually fails, the adjustment proceeds down the decision tree resulting in a combination of the modifiable aspects that affect the matching process. Modifying the patch size is the first attempt at improving the matching result because it is the most probable reason for matching failure. If however this does not have a positive influence on the matching, any undeterminable affine shaping parameters are excluded from the mathematical model, following which the patch size is once again modified if errors persist. In the case that excluding undeterminable parameters and modifying the patch size fails to produce an acceptable result, a new estimate is generated and the adjustment process is once again free to modify the patch size and model if necessary. This strategy structures the three modification processes in a hierarchical fashion illustrated in Figure 5.8(b). In the simplest case only the patch size is changed, while in the extreme case, a new estimate is generated, undeterminable parameter(s) is/are excluded from the model and the patch size is adjusted. The modification of the patch size is performed within a minimum and maximum size range as long as the adjustment is improving the result. The improvement is judged by the values of R_{ts}, σ_0, σ_x and σ_y which must be better than in the previous adjustment. Increasing the patch size attempts to satisfy the first constraint of LSM, i.e. the patch must contain enough signal to allow the determination of all six

affine parameters. On the other hand, if increasing the patch size does not eliminate the matching error, the size of the patch is decreased, favouring the second constraint of LSM which states that the patch must be small enough to represent a locally planar surface. The undeterminable parameters are only excluded once patch size adjustment fails to improve accuracy. Once all the possibilities are exhausted, the matching is considered failed which may amount to tracking failure.

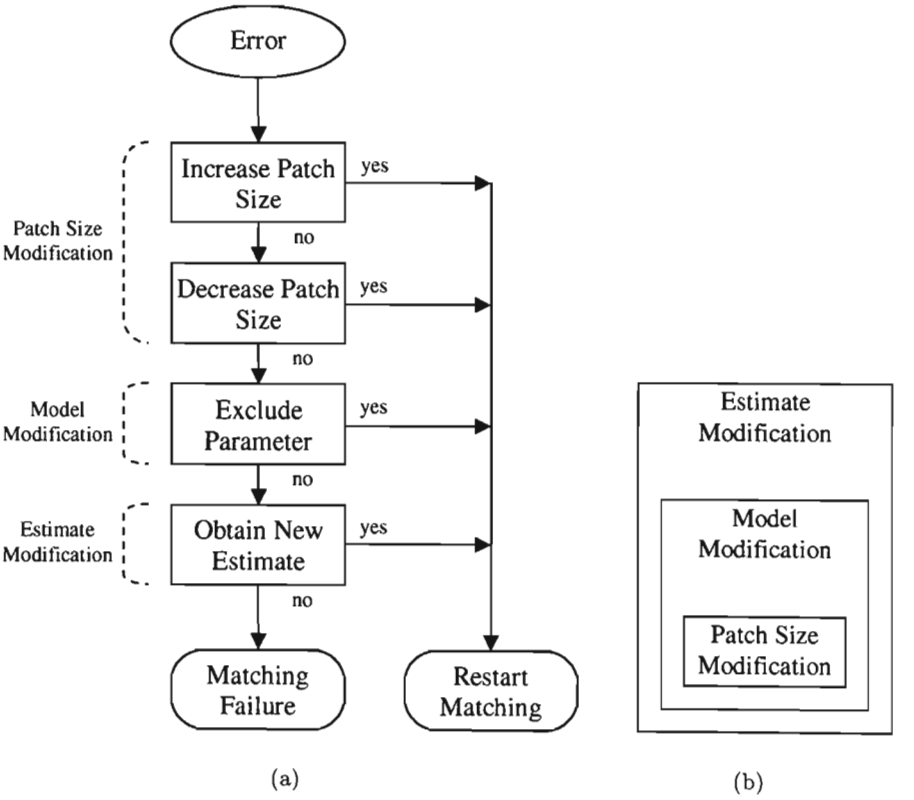


Figure 5.8: (a) Flow diagram of the matching parameter adjustment process, (b) hierarchical structure of parameter modification.

5.1.5 Convergence Radius

The convergence radius of LSM is a crucial aspect because it dictates the permissible uncertainty in the initial estimate. It describes the maximum error of the estimate which LSM is able to deal with to arrive at the correct solution. The convergence radius is determined by the signal present in the patch and in most cases is not a circle as one would

presume from the term *"radius"*, rather it is direction dependant. To avoid confusion, some researchers refer to the convergence radius as the *"pull-in range"*. Although the convergence radius has a direct affect on the convergence rate, stability and accuracy, the main motivation behind its analysis is to determine the necessary accuracy of the prediction scheme implemented in the tracking algorithm.

Expressing the convergence radius by a fixed number of pixels is inconsequential and does not provide any useful information from a theoretical point of view. The problem with defining the convergence radius stems from the fact that it is dependant on many factors and cannot be fixed. It changes from image to image and is influenced by the patch size, signal content, filtering, the method of calculating the **A** matrix and image gradients to name a few examples. In photogrammetry, some investigations have been done into the convergence radius of least squares matching. Ackermann [126] and Forstner [139] stipulate that an approximation within 3 pixels is required for fast convergence when no smoothing is performed. Pertl [140] states that the estimates should be within 1.5 pixels because LSM is used only for fine matching. However both Ackermann [126] and Baltsavias [134] point out that successful matching was achieved with estimates ten or more pixels off the target, thus the suggested estimate range may not necessarily dictate the maximum convergence radius in all matching cases. On the contrary, a 1.5 pixel limit may not indicate sufficient estimate accuracy either. In cases where the image consists of repetitive patterns separated by 2 pixels, the initial estimate must be within 1 pixel to achieve correct matching.

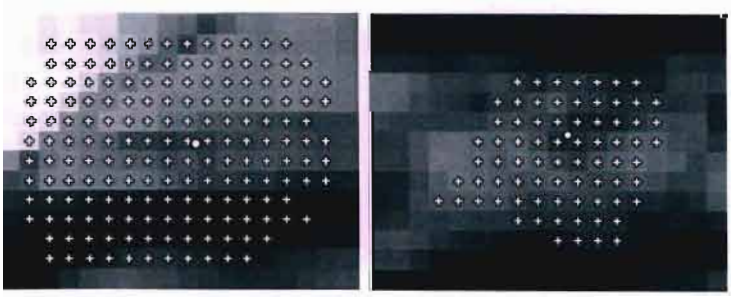


Figure 5.9: Examples of the convergence radius in two matching experiments.

The convergence radii of two matching experiments using images taken from the captured

motion data are displayed in Figure 5.9, where the estimates leading to a successful convergence are denoted by crosses and the target locations by circles. These two cases clearly illustrate the variable and directional nature of the convergence radius. In the left image, all points within approximately 6 pixels from the target were correctly matched, whilst in the right image the maximum error that guaranteed convergence was only 3.5 pixels. The prediction scheme implemented in the tracking algorithm used the 3 pixel error as a conservative guideline when estimating the locations of points in subsequent frames. Since the motion data did not contain closely spaced repetitive patterns, there was no danger of false matching with estimates within that range.

5.2 2-D Tracking

The 2-D tracking algorithm based on least squares matching consists of an initialisation step and a prediction scheme added onto the automatic LSM module described in Section 5.1 (Figure 5.10). The template patch is obtained from the current frame, while the search patch is defined at the predicted location in the next frame. Successful matching effectively tracks the point in the two frames and by looping this process, tracking is achieved throughout the monocular sequence. The initialisation step generates the 2-D locations of the desired points in frame 1 that are tracked throughout the video sequence. This may be an automatic process that selects key features, or a manual step which relies on the user to input the key locations (as implemented in this work).

The prediction stage is vital for successful matching. The investigation into the convergence radius of least squares matching produced a conservative limit to the prediction error of 3 pixels. Section 4.2 introduced normalised cross-correlation, a coarse matching technique with a maximum accuracy of ± 0.5 pixel. Although NCC accuracy satisfies the initial estimate requirements of LSM, to avoid lengthy computations and to reduce the possibility of false matching NCC must be performed in reduced search space rather than the whole image. In monocular sequences epipolar geometry cannot be exploited. Instead, prior motion of the point in the 2-D image plane is used to predict the point's location in the subsequent frame. The prediction process is depicted in Figure 5.11.

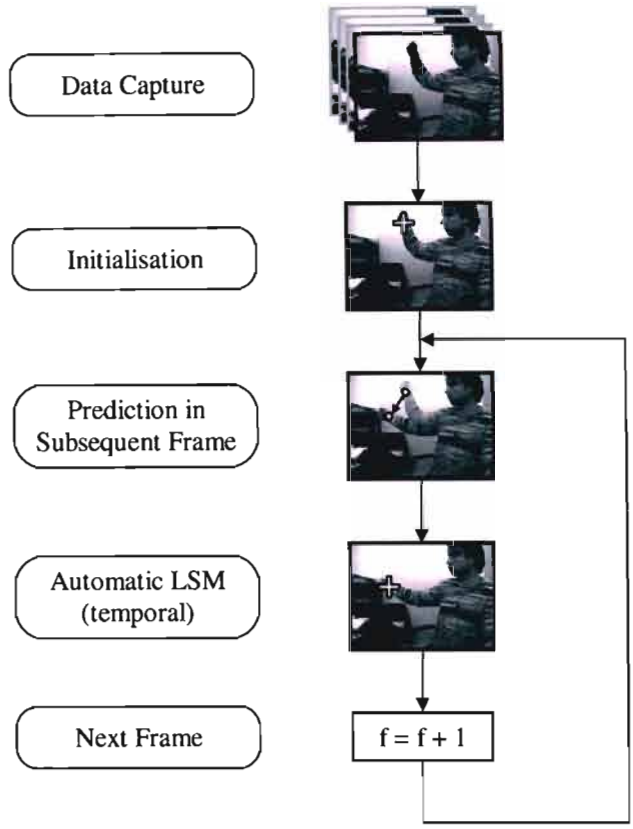


Figure 5.10: 2-D tracking algorithm flow diagram.

The instantaneous velocity of point $\mathbf{p}_0$ is calculated from its location in the current and previous frame. Adding the change in displacement $[\Delta x\ \Delta y]$ to the current location $[x_0\ y_0]$ results in $\mathbf{p}_{+1}$, a 2-D velocity-based linear prediction of $\mathbf{p}_0$ in the subsequent frame. The cross-correlation search space is defined around $\mathbf{p}_{+1}$ and the estimate is established at the location of best cross-correlation. The size of the cross-correlation search space is dictated by the accuracy of the 2-D velocity-based estimation. The 2-D velocity-based estimation can provide accurate enough estimates (within 3 pixels) for slow changing inter-frame motion in which case cross-correlation improvement is not required. However the estimate's uncertainty increases with increasing change in velocity (i.e. acceleration), and correspondingly the cross-correlation search space grows.

Figure 5.12 shows the pixel error of the 2-D velocity-based prediction in three cases with

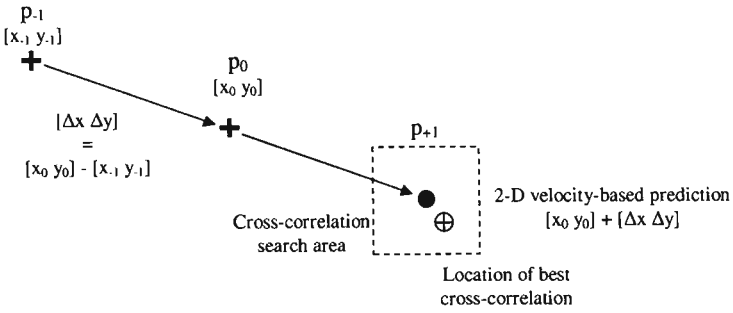
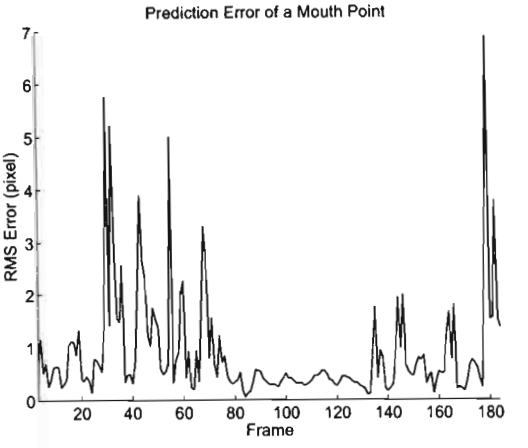


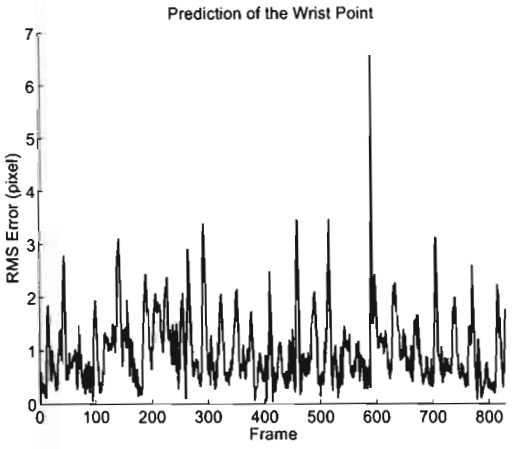
Figure 5.11: 2-D linear prediction. Using prior 2-D velocity to reduce cross-correlation search space.

varying degrees of inter-frame motion, together with the 3 pixel limit of the prediction error. Figure 5.12(a) refers to a point on the mouth, taken from a 2-D tracking of facial expressions experiment (Figure 5.13(a)). Changes in facial expressions occur rapidly resulting in sudden changes in velocity and consequently a number of prediction errors greater than 3 pixels. Thus normalised cross-correlation is required to ensure estimates remain within 3 pixels. Due to the size of the errors the search space must have a minimum radius of approximately 5 pixels. Figures 5.12(b) and 5.12(c) refer to points taken from the arm tracking experiment shown in Figure 5.13(b). The motion of the arm is much smoother than that of facial features (i.e. changes in motion do not occur as suddenly) thus the 2-D velocity prediction is sufficient in most instances. The prediction of the elbow Figure 5.12(c) is always within acceptable limits, while the prediction of the wrist Figure 5.12(b), which undergoes greater motion changes, requires NCC improvement in a 3 pixel radius search area.

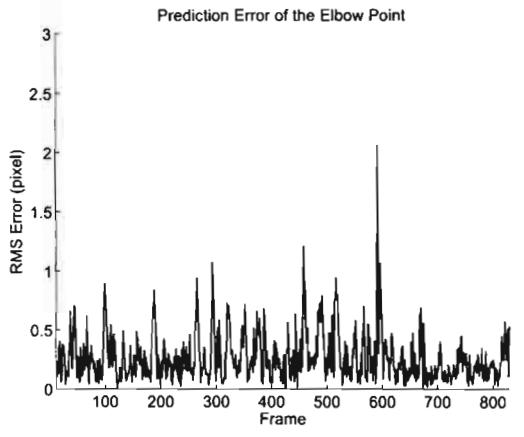
Although with hindsight it becomes evident that in some cases the 2-D velocity-based prediction is sufficient, one can not foretell this before the points are tracked. For this reason the 2-D tracking algorithm prediction scheme always made use of normalised-cross correlation to improve the estimate. This method not only eliminates estimates that fall outside the 3 pixel safety range, but it improves the accuracy of the estimates in general. As an example, NCC reduced the average estimate error of the wrist point in Figure 5.12(b) from 0.83 pixels ($\sigma = 0.56$) to 0.39 pixels ($\sigma = 0.15$). In situations where one can anticipate relatively smooth motion such as the case of tracking the arm, the



(a)



(b)



(c)

Figure 5.12: Prediction error, (a) point on the mouth, (b) wrist, (c) elbow.

cross-correlation search space can be defined smaller than in cases with rapidly changing motion as was experienced in tracking facial expressions. Although the addition of cross-correlation matching to the prediction scheme increases the computational requirements of the tracking algorithm, it is offset to a degree by the faster convergence rate of LSM resulting from the improved estimate.

(a)

(b)

Figure 5.13: Selected frames from 2-D tracking experiments, (a) tracking facial expressions, (b) arm tracking.

5.3 3-D Tracking

The 3-D tracking algorithm follows the same basic strategy of the 2-D algorithm, whereby tracking is achieved by looping a matching process in two successive frames aided by an appropriate prediction scheme. However, the 3-D tracking algorithm incorporates data from multiple views and tracks the points in subsequent frames (temporally) as in the 2-D case, as well as within the multiple views (spatially). The additional information provided

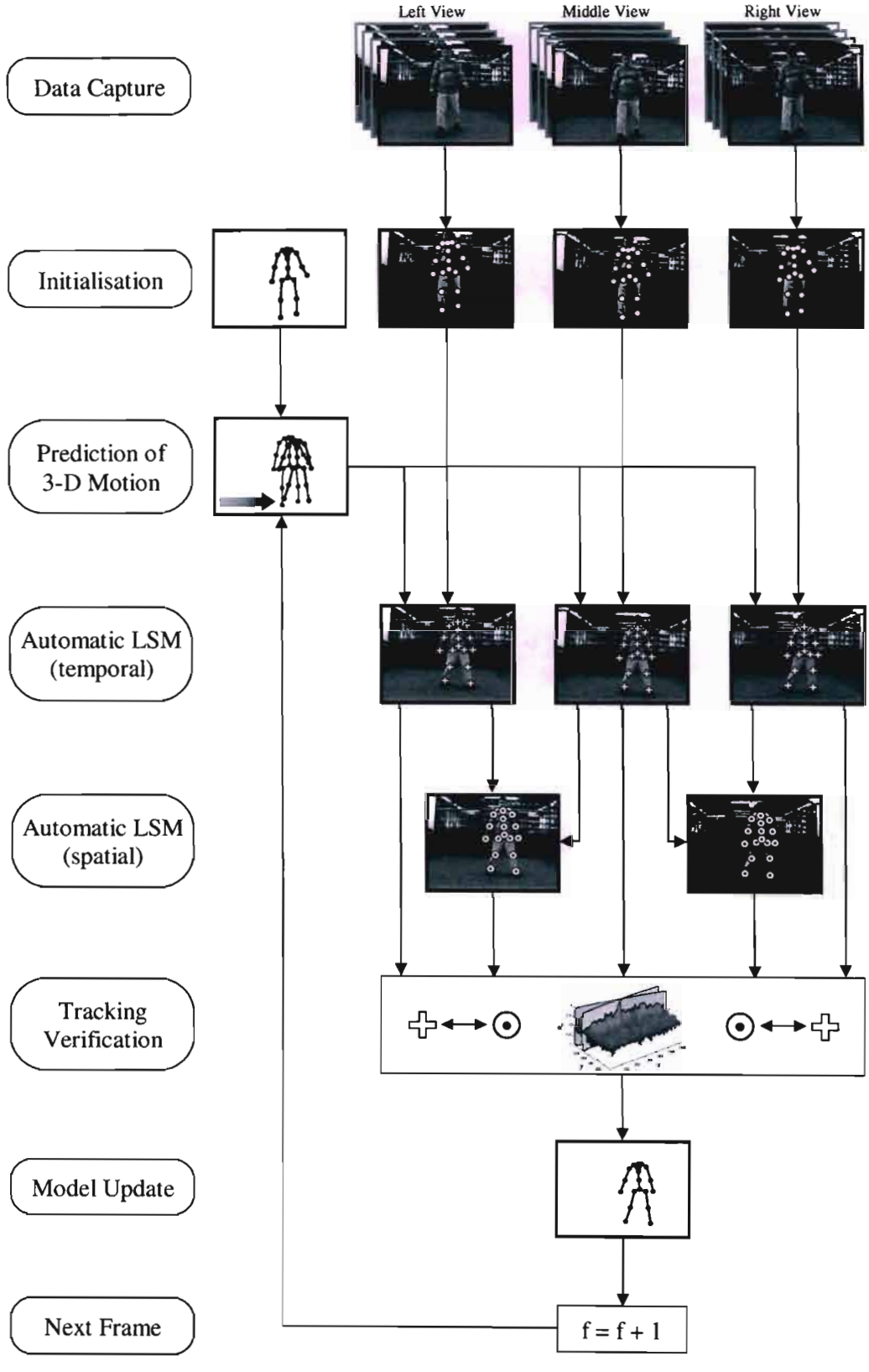


Figure 5.14: 3-D tracking algorithm flow diagram.

by the calibrated multiple views improves tracking robustness and enables the recovery of 3-D information. The following discussion focuses on the specific application to human motion capture, however the tracking algorithm itself may be easily adapted to tracking other points/objects.

Figure 5.14 displays the flow diagram of the 3-D tracking algorithm. The multi-view video sequences were acquired by three synchronised Dragonfly IEEE-1394 cameras from Point Grey Research [45], which produce images at 30 fps with a resolution of 640×480 pixels. The calibration step discussed in Section 3.5 was performed before capturing the motion data in order to estimate the camera parameters. The initialisation step requires the 2-D pixel locations of the desired points in the first frame of one of the three views, following which the correspondences in the other views and the 3-D model are generated automatically. The prediction process is bound by the same constraint as in the 2-D case (maximum 3 pixel error), however prediction is no longer made in 2-D, but in 3-D with the aid of a model. Point tracking in successive frames involves finding temporal correspondences in parallel for all three views and subsequently spatial correspondences in the middle-left and middle-right image pairs. The tracking verification stage analyses the spatial and temporal matching results. Provided the difference between the two locations is within preset limits and the location of the point is verified by epipolar geometry, tracking can be considered successful. After a successful completion of each tracking loop (tracking in successive frames) the correspondences in the three views are used to update the 3-D model (by triangulation, equation (3.29)) and the process repeats until the end of the sequence.

5.3.1 Skeleton Model

The purpose of modelling the human body in human motion capture applications is to assist the tracking process by aiding segmentation, prediction of future states, occlusion or collision, or helping to recover pose. Various models and their function have already been described in Chapter 2. In this work the model is used in the prediction stage of the tracking process and subsequently in pose recovery. The tracking algorithm only requires

prediction at the joint level which is effectively achieved by the skeleton model. There is no need to model the flesh because segmentation of the subject from the background is not required and neither is the estimation of the subject's physical dimensions for the purpose of pose recovery by fitting the model into the body contour.

The complexity of the skeleton model is dictated by the desired output. The decoder of the model-based coding scheme synthesises the subject's motion using an 18 joint, 31 DOF model. As a result the analysis stage must be able to extract all 31 parameters from the tracked data, which in turn necessitates for the human motion capture system to track all 18 joints. Consequently the skeleton model is defined by the 18 joints required by the pose estimation process in order to provide prediction for the corresponding points in the three views.

Figure 5.15 shows the 3-D skeleton model used in the 3-D tracking algorithm and in the pose extraction process which is dealt with in Chapter 6. The 18 points of articulation (joints) are connected by 17 segments and correspond to the following anatomical joints:

- left and right wrists
- left and right elbows
- left and right shoulders
- left and right sternoclaviculars
- cervical vertebrae (4)
- thoracic vertebrae (6)
- lumbar vertebrae (3)
- sacroiliac
- left and right hips
- left and right knees
- left and right ankles

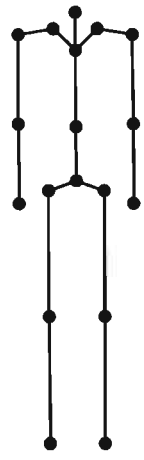


Figure 5.15: 3-D skeleton model.

5.3.2 Initialisation

The initialisation step of the 3-D tracking algorithm requires the same input from the user as the 2-D tracking algorithm. In addition to the 2-D pixel locations in one view, an automatic process follows which generates correspondences in the other views as well the 3-D model in the first frame. To initialise the tracking algorithm for human motion capture, 18 points that define the skeleton model are selected in the middle view, which correspond to the locations indicated in Figure 5.16.

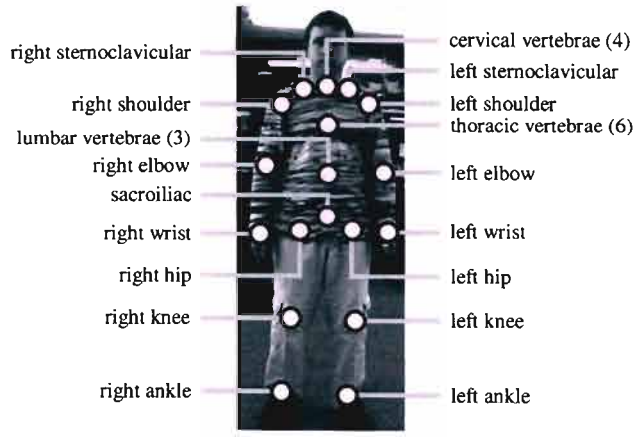


Figure 5.16: Key point locations on the human body.

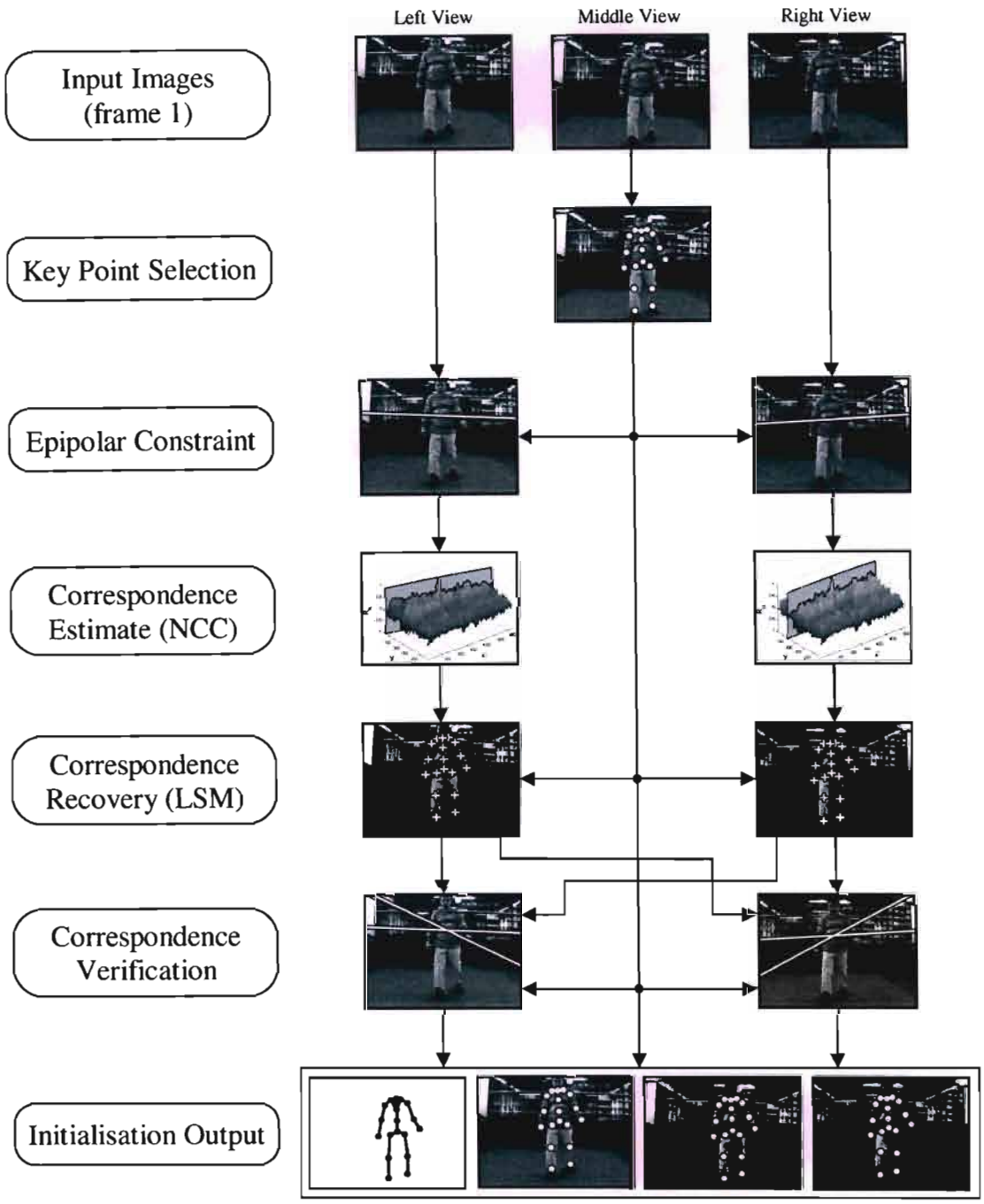


Figure 5.17: Initialisation stage flow diagram.

Figure 5.17 shows the flow diagram of the steps that follow the manual point selection in order to complete the initialisation stage. After the manual identification of the 18 joint

locations in the middle view, corresponding epipolar lines are generated in the left and right images in order to reduce the cross-correlation search space. Normalised cross-correlation along the epipolar lines provides correspondence estimates for the 18 points in the two images, which are subsequently refined by least squares matching. Before commencing the tracking process the recovered correspondences are verified by epipolar geometry to ensure that indeed they correspond to the locations defined in Figure 5.16.

The discussion of epipolar geometry thus far focused on reducing the search space by the constraint arising from the geometry of two views. The introduction of a third view increases the constraint even further, from a line to a point. Thus, a correspondence in two views can predict the location of the correspondence in the third view at the intersection of the epipolar lines (Figure 5.18). In an ideal situation, with a perfect camera model, exact camera parameters and precise point correspondences, the epipolar lines in the left view resulting from the points p and p' in the middle an right views respectively would intersect at p'', the location of the correspondence. Accordingly, the matched points in the left and right views should lie at the intersections of the epipolar lines provided they represent the same point in 3-D space. Of course, factors such as image noise, imperfect camera model and errors associated with calibration and matching will introduce a level of inaccuracy into the epipolar geometry. As a result the correspondences generally do not line up exactly with the epipolar line intersection and the verification process must allow for a certain distance from the epipolar lines. Once the data is verified, the 3-D locations of the joints are triangulated and the automatic tracking of the 18 points in the 3 views can commence.

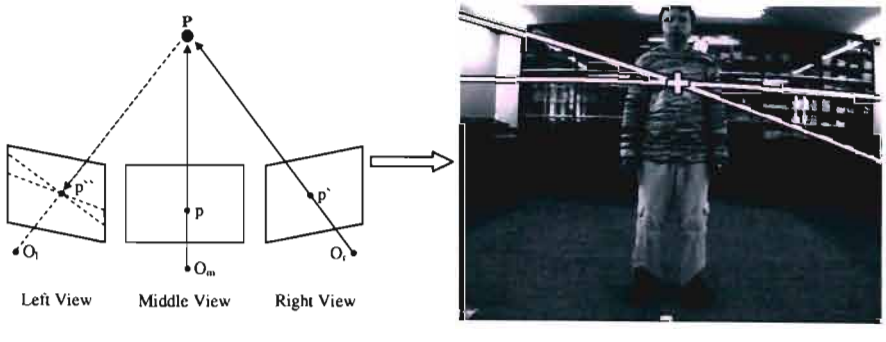


Figure 5.18: Correspondence verification in a trinocular system with epipolar geometry.

5.3.3 Prediction

The prediction requirements of the 3-D tracking algorithm are the same as in the case of 2-D tracking (i.e. producing 2-D pixel locations of the desired points in the subsequent frame within 3 pixels), with the only difference being that it must be done for three images instead of one. The approach also makes use of the motion information recovered in previous frames, with the initial estimate (when no *a prior* motion information is available) being established by cross-correlation alone as was done in the 2-D problem. The difference between the 2-D and 3-D case is in how a point's location in the subsequent frame is predicted based on its velocity. The 3-D tracking algorithm apart from tracking the selected locations in the 2-D image sequences also generates a model which represents the motion in 3-D space. Associated with the 3-D motion of the model are the 3-D velocity vectors of each joint. The 3-D velocity is used to predict the model's posture in the subsequent frame, which is then projected into the left, middle and right image planes, providing estimates for the 2-D pixel location of all joints (Figure 5.19).

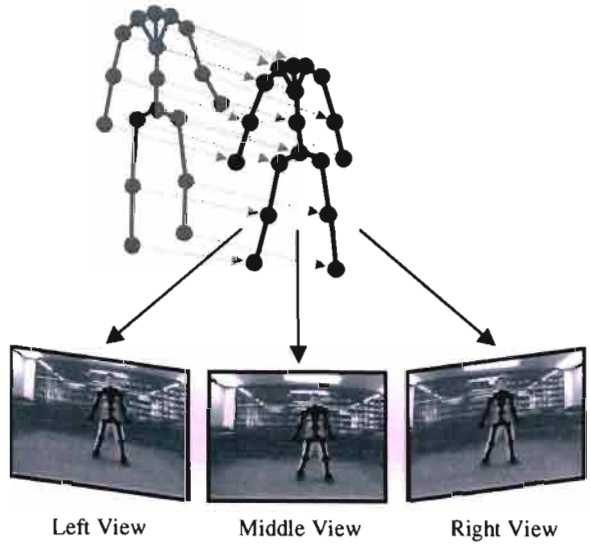


Figure 5.19: Predicting 2-D locations in multiple images based on the estimated motion of the 3-D model projected into the image planes.

Figure 5.20 shows the comparison between the 3-D velocity-based and the 2-D velocity-based predictions for the wrist point in Figure 5.12(b)). In most instances, the accuracy

of the predictions is similar and consequently, as a safeguard, the estimate improvement by cross-correlation is also performed. However, the 3-D based prediction produces better estimates at the extremes (where the direction of motion changes) which results in smaller search areas for the cross-correlation improvement step.

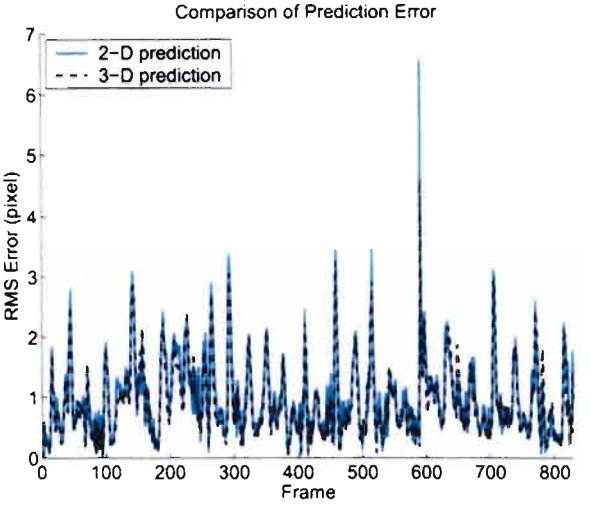


Figure 5.20: Pixel error of the 2-D velocity-based and 3-D velocity-based prediction.

5.3.4 Tracking in Multiple Views

The key aspect of the 3-D tracking algorithm is tracking in multi-image space. The information provided by multiple images allows the system to verify tracking results (discussed in Section 5.3.5) and most importantly compute the 3-D locations of the tracked data in each frame. The method of tracking corresponding points in multi-view sequences follows the strategy by D'Apuzzo [46] and is embodied in Figure 5.21. The process begins with known correspondences in *frame i* from the initialisation step or a previous tracking loop. The prediction scheme from Section 5.3.3 estimates the location of the point in the subsequent frame for each view and LSM is performed to refine the estimates. In *frame i+1*, the middle view is considered as the template and the left and right views as the search images. Spatial matching is performed to find the correspondences in the left and right views, which must produce comparable results to those of the temporal matching, in order for the point to be considered tracked in the two frames of the multi-view sequence.

The middle view is set as the *"template view"* because the resulting image pairs used in spatial matching (middle-left and middle-right) will contain the smallest level of geometric distortion out of the possible image combinations. Hypothetically this should yield the most reliable and accurate results from the given data. Accurate spatial matching results are important as the accuracy of the spatial correspondences directly affects the 3-D reconstruction i.e. the generation of the model.

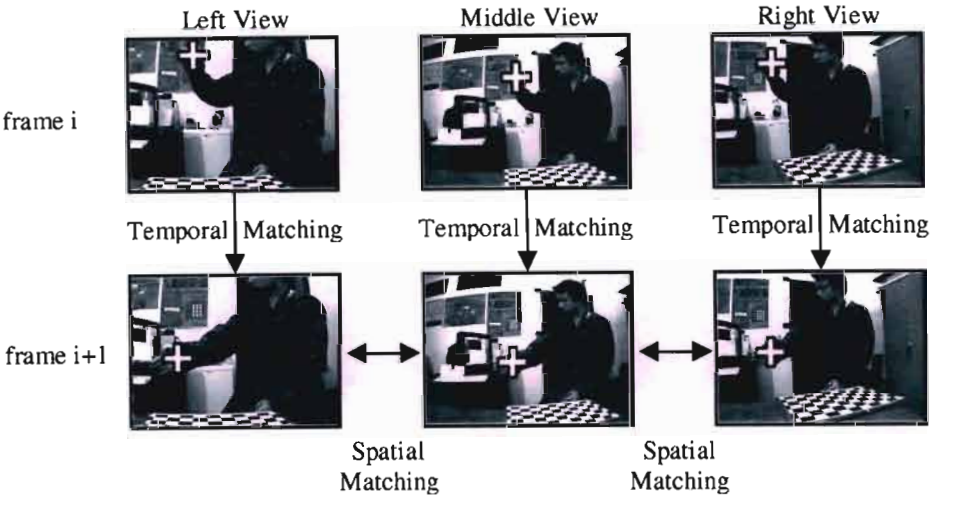


Figure 5.21: Strategy for point tracking in multiple views.

5.3.5 Tracking Verification

Tracking verification analyses the temporal and spatial matching results and exploits the epipolar geometry to establish whether the 2-D projections of the same 3-D point are being tracked in all views. An example of good tracking in one view is shown in Figure 5.22(a), where the epipolar line intersection and the matching locations closely match up. Without the spatial matching step the multi-view tracking problem essentially becomes a 2-D tracking problem performed in parallel for the three views. Spatial matching helps to determine how closely the point is tracked in the three views. A large difference between the temporal and spatial result is indicative of the 2-D tracking in the respective view diverging from the desired location. The possible outcomes of the spatio-temporal matching analysis are as follows:

1. Distances between temporal/spatial locations in both left and right views are within limits.

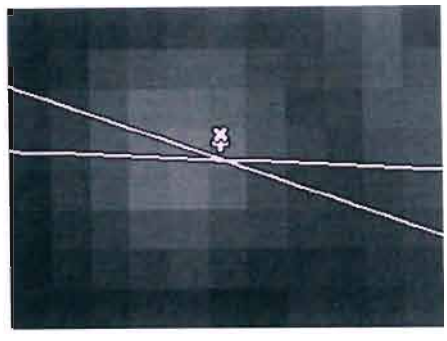
2. Distance between temporal/spatial locations in either the left or the right view is not within limits.

3. Distances between temporal/spatial locations in both left and right views are not within limits.

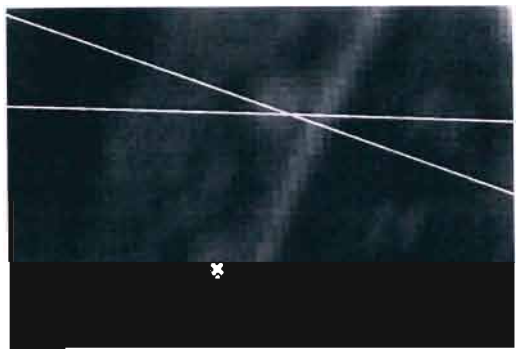
The limit of the distance between the two points is defined by the threshold value d_s which has been set to 0.15 pixels to allow for some disparity between the matching results. Case 1 represents the desired scenario where temporal tracking in all three views corresponds to the same point. Case 2 indicates a deviation of the tracked point in the left or right view respectively which needs to be adjusted before the following tracking loop. The adjustment involves using the spatial matching result to define the template patch for the subsequent temporal matching. In this way the point in the affected view is not allowed to diverge from the other two views. A divergence in the middle view is indicated by a spatio-temporal deviation in both the left and right views (case 3). The adjustment in this situation becomes more complex since no correctional data exists (according to the strategy depicted in Figure 5.14). Consequently the spatial matching process is repeated in reverse (i.e. the left and right views become the templates and the correspondence is found in the middle (search) view) and the better result (assessed by R_{ts}, σ_0, σ_x, σ_y) is used for the adjustment.

Consistency of the temporal and spatial matching results may not always guarantee successful tracking. When the image contains similar patterns, the matching process may converge onto the wrong solution and still display good matching properties. These cases are detected using epipolar geometry as introduced in Section 5.3.2. Figure 5.22(b) portrays the instance of incorrect matching convergence detected by epipolar geometry. When the presence of two correspondences can be established in the image triplet the erroneous location in the third view is corrected by repeating matching at the epipolar line intersection. On the other hand, if none of the tracked locations correspond to each other, the whole multi-image matching loop is repeated with improved estimates and modified

matching parameters.



(a)



(b)

Figure 5.22: Verification of tracking results, temporal matching location is indicated by a "×", spatial matching location is indicated by a "+". (a) successful tracking, (b) error detection by epipolar geometry.

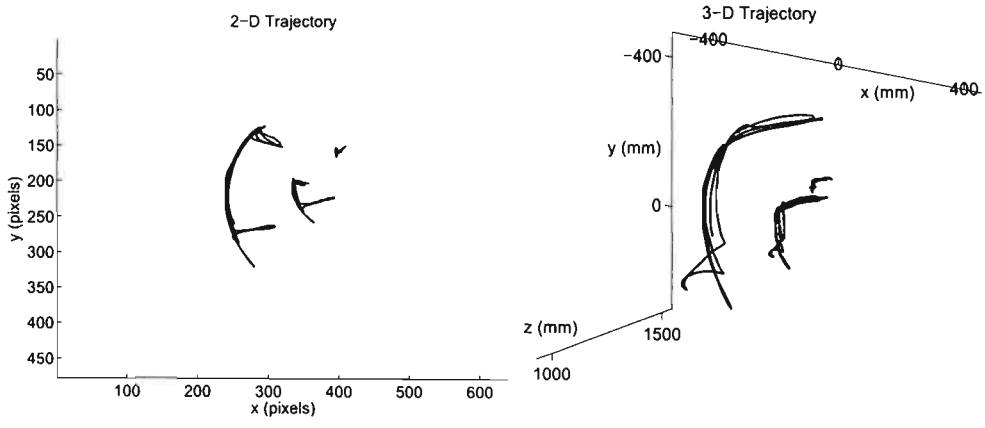
5.3.6 Advantages of Multiple Views

Tracking in multiple views has distinct advantages over tracking in monocular sequences. The benefits come in the form of increased robustness and the level of information associated with the output data. The 2-D tracking algorithm is solely dependant on temporal matching to track points throughout a sequence. As long as temporal matching is exhibiting good performance the tracking is considered successful and there is no additional data to support or contradict that assumption. By using multiple views, the tracking status may be assessed more thoroughly by combining the multi-view data as discussed in Sections 5.3.4 and 5.3.5. In addition to the data verification, lost tracking in one view

may be recovered through spatial matching from another view. Thus, if at least one view successfully tracks a point in the subsequent frame, tracking may not necessarily fail as would be the case in monocular sequences.

The second benefit of multiple views is particularly crucial to the application of the tracking algorithm to 3-D human pose recovery. The use of multiple calibrated views allows unambiguous 3-D reconstruction. Recovery of 3-D data is essential if the application requires motion information in 3-D space rather than just the 2-D image plane. The difference in output is illustrated by Figure 5.23 which displays several frames from the test sequence (a) and the locations of the selected points in each frame resulting from 2-D tracking (b) and 3-D tracking (c). Human motion capture systems that aim to extract

(a)



(b) (c)

Figure 5.23: Data output of the 2-D and 3-D tracking algorithm for an arm tracking sequence, (a) frames from the input sequence, (b) 2-D output trajectories, (c) 3-D output trajectories.

3-D pose from monocular sequences are faced with the effects of depth ambiguity, i.e. a segment of length l in a 2-D image has two possible orientations in 3-D space. This

becomes a trivial problem with the information recovered by the multi-view 3-D tracking algorithm.

Associated with the 3-D trajectories of each point is also the velocity and acceleration vector of the point at each time step which may be used in analysis of the particular motion. In this work, the recovered velocity was used to adjust the 3-D trajectories of the tracked point. The human motion capture system presented in this thesis samples the motion of the subject at 30 Hz. At this frequency, the motion changes smoothly between successive time steps and accordingly the velocity curve of each component (x, y, z) should also be smooth for the duration of the sequence. However, the recovered velocity exhibits some jitter, particularly in the z-component as depth is the most difficult dimension to recover accurately. Figure 5.24 displays the recovered z-component velocity of the wrist from Figure 5.23(a) (solid grey curve) and the smoothed result (black broken line). By smoothing the noisy velocity data and updating the 3-D trajectories, the characteristics of the captured human motion become more natural.

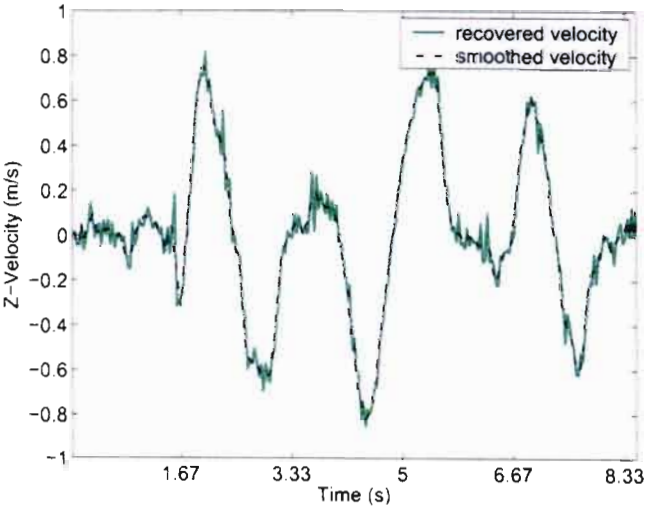


Figure 5.24: Smoothing of the z-velocity component.

5.4 Summary

The core idea of tracking points by finding correspondences in subsequent frames was adopted from D'Apuzzo [46]. This idea was then adapted to the presented application in order to track key location defining the complete human pose. Tests were incorporated into the tracking algorithm to monitor the behaviour of the affine update parameters in order to minimise user intervention. Furthermore, the implemented 3-D tracker performed prediction in 3-D space rather than 2-D image space, and included procedures to recover lost tracking in one or two views. Whereas D'Apuzzo [46] aimed to measure the subject's surface, the work presented in this thesis focused on MPEG-4 pose recovery and consequently tracking the complete surface was not necessary. The investigation of local surface tracking benefits is proposed as future work in Chapter 8.

This chapter has presented the necessary components required to utilise least squares matching for tracking purposes. The convergence radius of least squares matching is an important characteristic because it determines the necessary precision of the prediction scheme that generates estimates for the matching process. It has been shown that the convergence radius is neither set nor equal in all directions. A conservative estimate accuracy of 3 pixels was adopted for the prediction schemes of the implemented tracking algorithms which is a commonly suggested guideline and has proven sufficient through experimentation.

Least squares matching was first integrated into a 2-D tracking algorithm, which was then extended to a 3-D tracking algorithm by incorporating multiple views and a 3-D model. In both cases the prediction schemes utilised previously recovered motion information to predict the location of selected points in subsequent frames. For 2-D tracking the point's velocity in the image plane was used. In the 3-D case, 3-D velocity of the joints predicted the posture of the skeleton model, which was then projected into the three image planes to generate 2-D estimates of future states. In most cases both prediction schemes provided estimates within the 3 pixel error margin, however in instances where the direction of motion changed for fast moving points the error margin was exceeded. For this reason the velocity-based prediction was improved by normalised cross-correlation in a reduced

search space. Prediction based on 3-D velocity displayed smaller errors at the extremities, resulting in smaller cross-correlation search space.

The 2-D tracking algorithm tracks user selected points throughout the monocular sequence to produce the pixel trajectories of the tracked points. The 3-D tracking algorithm uses multiple views to track points both temporally and spatially. Temporal and spatial matching together with epipolar geometry is used to verify tracking results in each frame as well as recover lost tracks in one or two views thereby adding a degree of robustness when compared to the 2-D tracking algorithm. When tracking data is verified, the 3-D locations of the points are triangulated and the skeleton model is updated. Upon completion of the tracking process the user is presented with 3-D trajectories of the tracked points. The subject's motion is being sampled at 30 Hz and consequently the recovered motion should be smooth. By this assumption, the recovered velocity of the tracked points is smoothed in order to adjust the output 3-D data.

This chapter has presented the part of the model-based coding system that extracts pertinent information from the input images. Separating the problem into modules, the input to the human motion capture module is a multi-view sequence of images captured by calibrated cameras and the output consists of the 3-D locations of the joints in each frame. Chapter 6 discuses the following module which extracts pose parameters from the 3-D data. These parameters constitute the information sent over the communication channel which is used by the decoder at the receiver end to rebuild the skeleton model in each frame and consequently reconstruct the subject's motion.

Chapter 6

Virtual Humanoid Animation with MPEG-4

This chapter represents the crossover from the computer vision aspect of the research to the computer animation domain. At this point in the model-based coding scheme the 3-D human pose has been generated from the input video sequence by the human motion capture component. To complete the analysis task performed at the encoder, the model is parameterised, with the extracted parameters comprising the information sent over the communication link. At the receiver the decoder synthesises the output by transforming the model into the pose described by the incoming parameters. By performing this process at the frequency defined by the image sequence frame rate, the output model is animated and human motion reconstruction is achieved.

This chapter introduces the pose parameters, the extraction process from the tracked 3-D data and the model used for the synthesis of the output. The pose parameters have been defined with compliance to the ISO/IEC 14496 standard developed by the Moving Pictures Experts Group (MPEG) which is better known as MPEG-4. MPEG-4 has been designed with low bit-rate applications in mind and provides specific tools for model-based coding and animation of humans, as well as efficient coding schemes for the compression of the extracted parameters. The scope of this work deals with the animation aspect of

the MPEG-4 framework, which describes the orientation of individual limbs with respect to each other and does not deal with the modelling of the human body or compression of the transmitted parameters.

6.1 MPEG-4 Overview

The MPEG-4 standard is not a direct improvement of the MPEG-1 or MPEG-2 coding standards. MPEG-2 generates the content from various sources (video, text etc.) and composes it into a plane of pixels which are then encoded and transmitted. MPEG-2 is a static presentation engine; graphics and text may be added to the content, but cannot be deleted. MPEG-4 on the other hand is dynamic. Different objects are encoded and transmitted separately and on the other side decoded into their own elementary streams. The scene composition is performed after decoding instead of before encoding as done in MPEG-2. In this way the interactivity of MPEG-4 is introduced, as well as the coding efficiency because each object is encoded with its own optimal coding scheme.

MPEG-4 adopts an object orientated approach where different media (e.g. audio, video, text, animation, pictures, 2-D and 3-D) are treated as "media objects" that can be of natural or synthetic origin. This allows new functionality such as video manipulation, scalable video encoding, synthetic and natural video coding, 3-D object compression or face and body animation coding, while improving already available functionalities like coding efficiency and error resilience by using the most adequate coding technology for each type of data [141]. With the new aspects of MPEG-4 comes a diverse range of applications of which video telephony, video conferencing, remote classroom and web based virtual assistants are of particular interest to the concept of model-based video coding of humans.

The object-based scene generation constructs rich audiovisual scenes from the media objects, allowing more efficient and higher quality representation of media content [142]. Figure 6.1 displays an example of an MPEG-4 scene made up of various media objects that are coded, transmitted and manipulated independently. Objects within the scene can be grouped to form "complex media objects" which becomes beneficial when the

user wants to manipulate related objects (e.g. grouping of the visual and audio aspects associated with the dog).

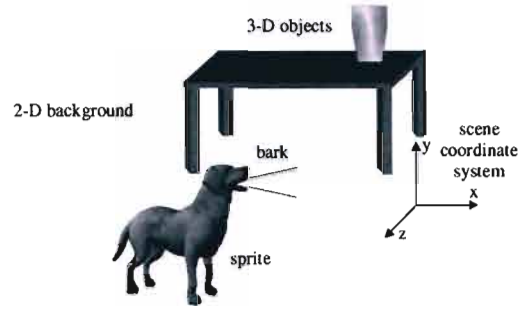


Figure 6.1: Example of an MPEG-4 scene.

The composition of the final scene is performed at the decoder using the scene description information. The dynamically changing scene description coding format within MPEG-4 is known as Binary Format for Scenes or BIFS. The description of the scene follows a hierarchical structure represented by the scene graph, where each node is a scene object. An example of a scene graph corresponding to Figure 6.1 is shown in Figure 6.2. The BIFS commands are not only able to add or delete objects from scenes, but also to change the object's visual or acoustic properties without changing the object itself. Thus it is possible to interact with objects within the scene and users may modify the scenes or even alter the behaviour of the separate objects.

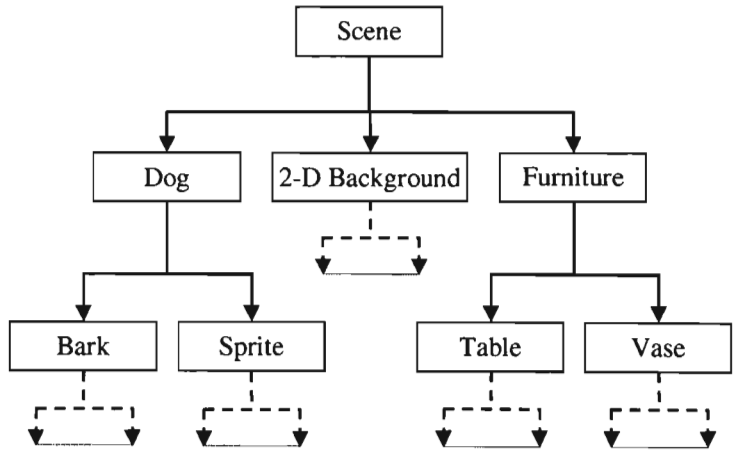


Figure 6.2: Example of a scene graph corresponding to Figure 6.1.

6.2 MPEG-4 Human Body Animation Tools

The topic of animating virtual characters has been addressed by the Face and Body Animation (FBA) subgroup of the Synthetic and Natural Coding Group (SNHC) [143] of MPEG. The MPEG-4 standard provides a complete solution by specifying the character modelling representation, the animation parameter representation and compression. Version 1 of the standard only addressed the animation of the face, with body specifications being added in the second version. The face and body objects are incorporated into the standard by defining specific nodes in the scene graph, the face and body nodes, and a unique dedicated stream, the Face and Body Animation (FBA) stream. The FBA object is defined by parameters capable of animating both realistic and cartoon-like humanoid models, providing tools for model-based coding of video sequences containing human faces and bodies. Since the FBA framework is limited to human-like characters, the SNHC group has adopted the Bone-based Animation (BBA) framework [144] which allows the realistic animation of any articulated virtual character whilst still achieving low bit-rate transmission [145].

If a virtual humanoid (e.g. the one synthesised by the presented model-based coding system) is added into the scene in Figure 6.1, the Body Object is represented by the FBA scene graph [146]. The FBA scene graph (Figure 6.3) has a Body node as its root with three children, the Body Definition Parameters (BDP) node, the Body Animation Parameters (BAP) node and the renderedBody node. These nodes are associated with anatomical segments and edges defining the relationship between them.

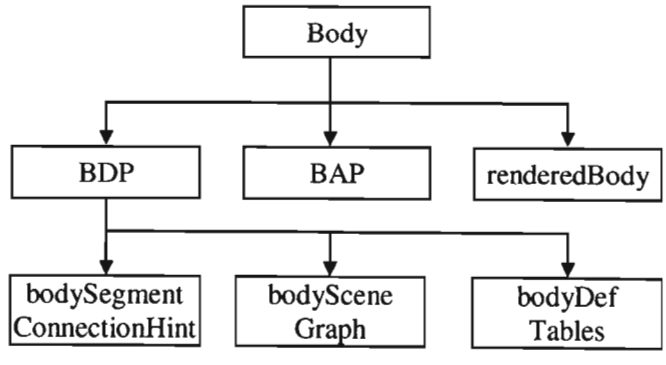


Figure 6.3: FBA scene graph.

BDPs control the intrinsic properties of the segment, its geometry, surface local topology and texture. These parameters are actor[1] specific, and allow the decoder to create an FBA model with specified shape and texture. BAPs define the extrinsic properties of a segment, i.e. its 3-D pose with respect to a reference frame attached to its parent. Unlike BDPs, BAPs are meant to be generic, so when correctly interpreted, a particular set of BAPs should produce similar motion/pose results when applied to different avatar models. Typically, the BDPs need to be transmitted only once, thereafter only BAPs are required to animate the model in each frame. Modelling and animating a model using only BDPs and BAPs may produce undesirable effects in the form of broken mesh at the joint level between two segments [147]. To deal with the realistic 3-D deformations induced by animation, Body Animation Tables (BATs) are used. BATs specify vertices of the 3-D model that undergo non-rigid motion, and displacement for each vertex is defined as a function of BAPs. MPEG-4 does not limit the range of motion of the segments defined by their BAPs. As a result, unrealistic motion is possible which may be an advantage in gaming applications, but may require extra effort when realistic simulations are desired. The MPEG-4 standard deals with the efficient coding of animation parameters, and does not standardise a specific model or a method of extracting the animation parameters. The standard does provide two methods for the encoding of the animation parameters; a prediction-based method and a DCT-based method. A comparison between these two methods was done by Pretaux *et al.* [148] [149]. The DCT-based method is more appropriate when dealing with a wide range of bit-rates, while the predictive method is recommended for applications requiring nearly lossless compression [150]. Typical bit-rates for body animation are 10 to 30 kbps [151], although Capin *et al.* [36] report that good quality animations of an FBA object were achieved at 4.4 kbps, with 1 kbps at a lower quality.

6.3 MPEG-4 Body Animation Parameters

The MPEG-4 standard consists of several parts. The FBA object is dealt with in Part 1: Systems [152] and Part 2: Visual [42]. Part 1: Systems specifies the representation and

[1] In computer graphics the subject that is being modelled is often referred to as the actor i.e. in this work "actor" refers to the subject from the input video sequence.

coding of the geometry of the body (BDPs). The animation parameters that have been utilised in the presented system are specified in Part 2: Visual.

The MPEG-4 avatar is defined as a segmented virtual character using the H-Anim [39] nodes and hierarchy. The BAP parameters comprise joint angles connecting the different body parts, with joints including toe, ankle, knee, hip, spine (C1-C7, T1-T12, L1-L5), shoulder, clavicle, elbow, wrist, and the hand fingers. Each movement may be described by up to 3 separate rotations which together with 3 translation BAPs (tr_vertical, tr_lateral, tr_frontal), and 110 BAPs reserved for extensions bring the total to 296 BAPs defined for the animation of the virtual character. The translation parameters are defined in millimetres and the rotation unit (BAPU) as $\pi/10^{-5}$ radians. The angular values are specified with respect to their local 3-D coordinate system, whose origin lies at the gravity centre of the joint common to the segment and its parent. The coordinate system is fixed to and moves with the parent body part i.e. the rotation normals are not fixed with respect to the body or world coordinate system. The joint rotations are restricted to the anatomical planes of the human body defined in Figure 6.4(a) with the exception of twisting and torsion movements. The two most common joint movements, flexion and abduction, are illustrated in Figures 6.4(b) and 6.4(c) respectively for the left shoulder joint.

The independence of the BAP description and the generic 3-D model means that a model does not need to be transmitted in order to animate the avatar. If no BDPs are transmitted, the decoder uses the default body model for subsequent animation. Animation is performed by accessing the H-Anim joints and altering their angle values to match those of the BAP stream. Any BAP value can be null, in which case the null component is replaced by the corresponding default value when the body is rendered at the decoder. In order to further decrease the bandwidth requirements, the joints comprising the body are grouped into 24 different categories with respect to their interrelationships and importance. These groups describe a section of the body in various degrees of model complexity (e.g. simple spine to complex spine). In this way, if the subject is only waving his right arm, only the BAPs corresponding to the right arm group need to be transmitted.

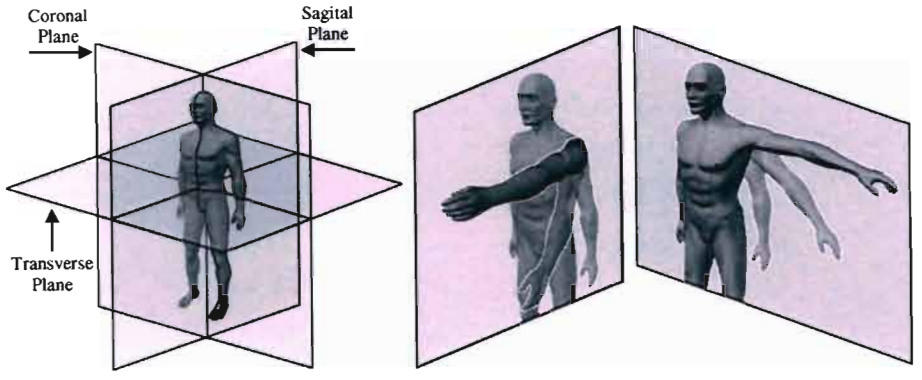


Figure 6.4: (a) Anatomical rotation planes, (b) shoulder rotation in the sagital plane (flexion), (c) shoulder rotation in the coronal plane (abduction).

The default joint centre locations suggested by MPEG-4 produce a default skeleton as depicted in Figure 6.5. The default posture is defined as:

- Standing posture
- Feet pointing in the front direction
- The two arms placed on the side of the body
- Palms facing inwards
- Hands pointing down, with thumb at 45 degrees inclination

The default posture implies that all BAPs have default values 0. Any subsequent rotation is assumed positive in the counter-clockwise direction with respect to the rotation normal. The default MPEG-4 skeleton in Figure 6.5 defined by the suggested joint centres represents an elaborate model. MPEG-4 however does not force applications to use such complex models and typically simpler variations are implemented to suit the purpose or to balance out computational speed with realism. In the scope of this work, the function of the model is to represent the extracted pose of a subject in each frame of the video sequence. Realistic modelling of the subject is not necessary as the skeleton figure adequately characterises the performed motion. Although the model used in tracking and

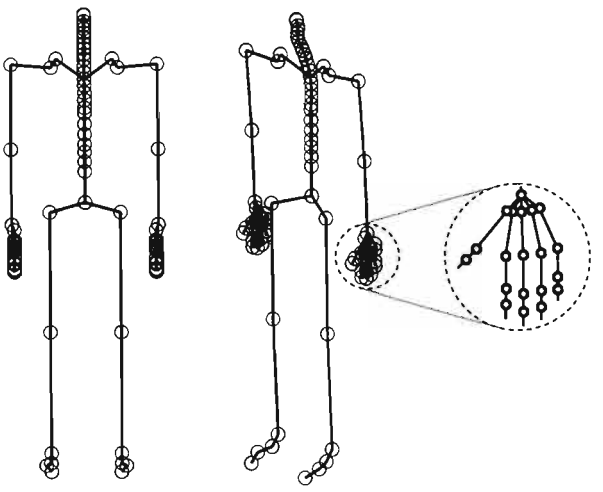


Figure 6.5: Suggested joint centre locations of the complete MPEG-4 skeleton showing front view, rotated view and close-up of the hand.

the model used in the synthesis process represent two separate problems, in the presented system they are very closely related. Since the tracking model needs to predict the locations of the joints that define the pose of the synthesised model, both the model in the human motion capture component and the model in the model-based coding system are described by the same joints which have been established in Chapter 5.

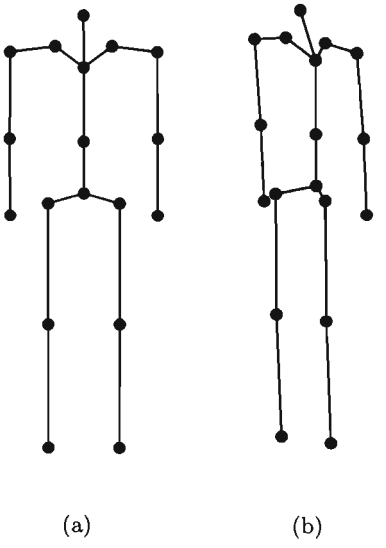


Figure 6.6: Model-based coding human model, a) front view, b) rotated view.

The default model present at the encoder and decoder of the model-based coding scheme is a simplified 31 DOF version of the MPEG-4 skeleton, depicted in Figure 6.6. Each joint has up to three BAPs associated with it representing the corresponding DOF. At the top of the hierarchy lies the HumanoidRoot. The HumanoidRoot is located near the sacroiliac, however it is not a joint; it is the link between the scene coordinate system and the local coordinate systems of skeleton i.e. it defines the location and orientation of the virtual character in the MPEG-4 scene. The 31 BAPs used to describe the pose of the skeleton model in Figure 6.6 are summarised in Table 5. Each BAP has a unique ID defined by the MPEG-4 standard that identifies the respective parameter. The angular values of the animation parameters indicate the rotation about a specified normal with respect to the default position of the affected anatomical segment. The rotation normals are standardised by [42] and an example pertaining to the right arm is given in Figure 6.7.

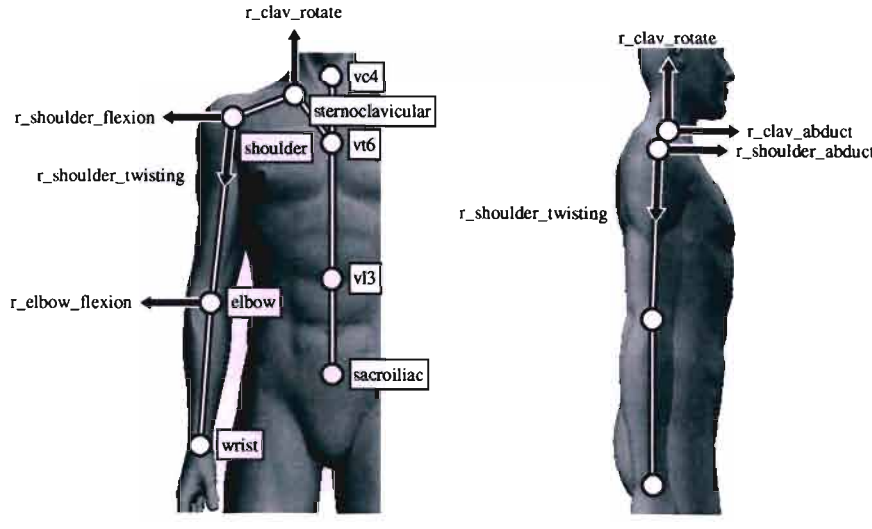


Figure 6.7: Rotation normals of the right arm, a) front view, b) side view.

Table 6.1: BAPs associated with the joints of the skeleton model

joint name	DOF	BAP ID	BAP name	movement restriction
HumanoidRoot	3	181	HumanoidRoot_tr_vertical	along the y-axis
		182	HumanoidRoot_tr_lateral	along the x-axis
		183	HumanoidRoot_tr_frontal	along the z-axis
sacroiliac	3	1	sacroiliac_tilt	sagital plane
		2	sacroiliac_torsion	body vertical axis
		3	sacroiliac_roll	coronal plane
l_hip	3	4	l_hip_flexion	sagital plane
		6	l_hip_abduct	coronal plane
		8	l_hip_twisting	along thigh axis
l_knee	1	10	l_knee_flexion	sagital plane
l_ankle	0	n/a	n/a	n/a
r_hip	3	5	r_hip_flexion	sagital plane
		7	r_hip_abduct	coronal plane
		9	r_hip_twisting	along thigh axis
r_knee	1	11	r_knee_flexion	sagital plane
r_ankle	0	n/a	n/a	n/a
vl3	2	114	vl3_roll	coronal plane
		116	vl3_tilt	sagital plane
vt6	3	87	vt6_roll	coronal plane
		88	vt6_torsion	body vertical axis
		89	vt6_tilt	sagital plane
vc4	0	n/a	n/a	n/a
l_sternoclavicular	2	24	l_sternoclavicular_abduct	coronal plane
		26	l_sternoclavicular_rotate	transverse plane
l_shoulder	3	32	l_shoulder_flexion	sagital plane
		34	l_shoulder_abduct	coronal plane
		36	l_shoulder_twisting	along forearm axis
l_elbow	1	38	l_elbow_flexion	sagital plane
l_wrist	0	n/a	n/a	n/a
r_sternoclavicular	2	25	r_sternoclavicular_abduct	coronal plane
		27	r_sternoclavicular_rotate	transverse plane
r_shoulder	3	33	r_shoulder_flexion	sagital plane
		35	r_shoulder_abduct	coronal plane
		37	r_shoulder_twisting	along forearm axis
r_elbow	1	39	r_elbow_flexion	sagital plane
r_wrist	0	n/a	n/a	n/a

6.4 Pose Parameter Extraction

The 31 BAPs from Table 6.1 are extracted from the tracked 3-D data as the final step of the analysis process performed by the encoder. The tracked 3-D model provides the 3-D locations of the joints, inherently defining the model segments and the segment lengths. Following the H-Anim specification [39], the virtual body consists of segments connected by joints arranged in a hierarchical fashion, where each joint node may contain other joint nodes. The human motion capture data does not provide any information regarding the rotation normals of each joint. Since the normals are attached to and move with the body parts, their directions are unknown. As a result, BAP extraction is performed one joint at a time starting at the top of the joint hierarchy shown in Figure 6.8.

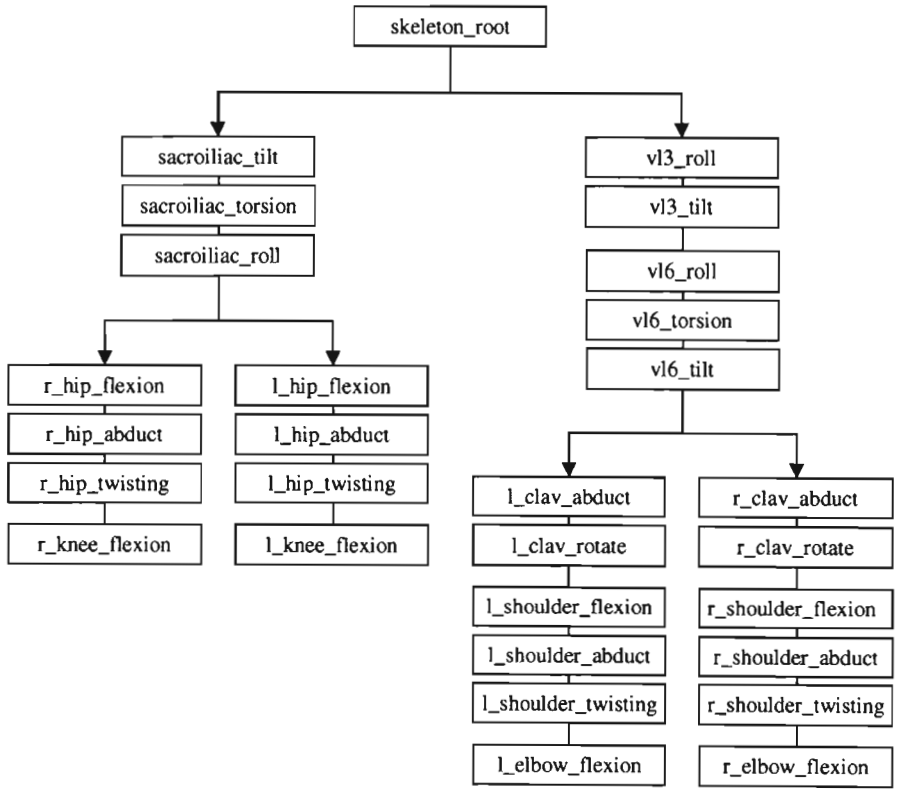


Figure 6.8: Skeleton model topology.

The pose parameter extraction process assumes that the HumanoidRoot (skeleton_root)

undergoes only translation within the scene. This is a reasonable assumption considering the human motion capture system cannot handle occlusion and therefore situations when the body rotates significantly in the scene causing joints to disappear are eluded. Thus the rotation normals of the sacroiliac and lumbar vertebrae (3) are aligned with those of the default posture and the respective BAPs can be recovered by finding the angle between the current segment and its default orientation. Once known, the inverse equivalent compound rotation is applied to all the dependant joints, essentially transforming the children joint(s) to their default locations. In this way, the default rotation normals of the children joints may be used, and the BAP extraction process may propagate down the articulated chain. The flow diagram of the pose extraction process is shown in Figure 6.9. The required data comes from the 3-D model of the subject and the skeleton model in the default posture scaled to fit the segment dimensions of the actor. Ideally the scaled default model would remain the same throughout the sequence, however due to errors introduced by the calibration, matching and reconstruction processes the segment lengths do vary within the sequence and consequently the default model must be generated in each frame.

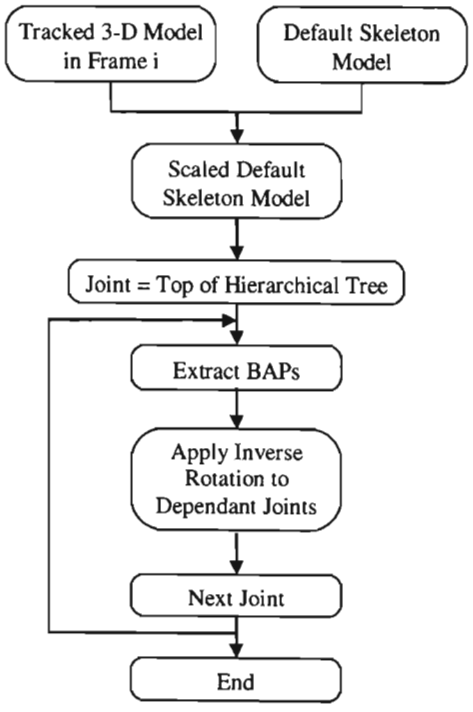


Figure 6.9: Pose parameter extraction flow diagram.

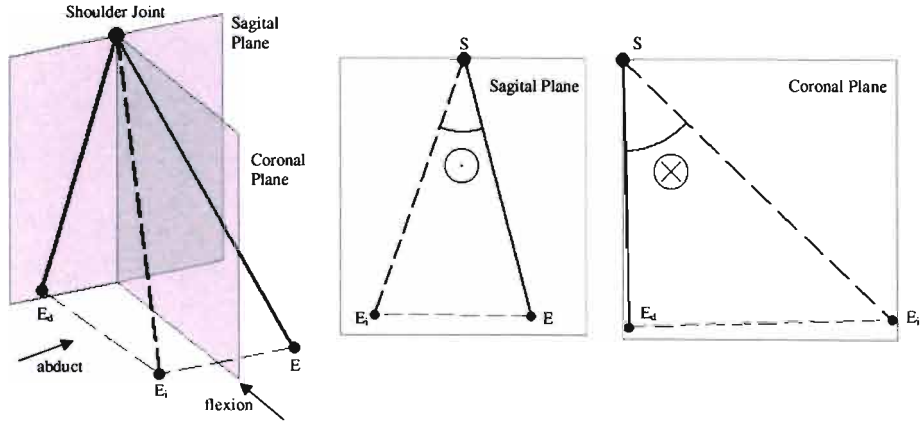
Recovering the rotation of joint's with only one DOF is a straight forward task which entails finding the angle between the default position of the segment represented by the vector $\mathbf{s}_d$ and the rotated segment in the current frame $\mathbf{s}_c$. The angle is found by utilising the dot product between the two vectors:

$$|\mathbf{s}_d|\,|\mathbf{s}_c| \cos\alpha = \mathbf{s}_d \bullet \mathbf{s}_c \tag{6.1}$$

To give the angle α by:

$$\alpha = \cos^{-1}\left(\frac{\mathbf{s}_d \bullet \mathbf{s}_c}{|\mathbf{s}_d|\,|\mathbf{s}_c|}\right) \tag{6.2}$$

Joints with multiple DOFs that are free to move both in the sagital and the coronal planes require an intermediate step which recovers the vector between the successive rotations before the equation (6.2) can be used to calculate the respective BAPs. This is illustrated in Figure 6.10, which shows the rotation of the right shoulder. The segment is defined by the shoulder and the elbow joints. In order to move the segment to its default location, it must first rotated from the elbow position E, to the intermediate position E_i, and then from E_i to the default location E_d. The resultant BAPs are the shoulder flexion and shoulder abduct. The flexion is the angle between the vectors SE and SE_i projected into the sagital plane, while the abduction is represented by the angle between SE_d and SE_i projected into the coronal plane.



(a) (b) (c)

Figure 6.10: BAP extraction for joints with multiple DOF, (a) compound rotation of the right shoulder, (b) recovery of r_shoulder_flexion, (c) recovery of r_shoulder_abduct. Rotation normals are indicated as either going into, or out of the page.

The BAP extraction process is illustrated in Figure 6.11 and Figure 6.12 using the right arm to represent a simple articulated chain, with the right sternoclavicular being the root at the top of the joint hierarchy. The pose data is generated by transforming the default pose of the right arm by some known BAPs (Figure 6.11). The process begins at the sternoclavicular joint which together with the shoulder joint define the segment that has been rotated by the r_clav_abduct and r_clav_rotate BAPs. A close-up of the two joints together with their rotation normals is shown in Figure 6.12(a) left. Once the two pose parameters are recovered, the inverse rotation is applied at the sternoclavicular which rotates the segment (corresponding to the collar bone) into its default position Figure 6.12(a) right. The inverse rotation propagates down the articulated chain, rotating the connected segments as well as the normals of rotation attached to each joint. The same process is then applied at the shoulder joint Figure 6.12(b) and the elbow joint Figure 6.12(c). The shoulder BAP r_shoulder_twisting can only be recovered once the flexion and abduct are known and the elbow has been rotated into its default position. Its rotation normal is along the upper arm and consequently is unknown at the time the shoulder is being processed. As the motion of the hands is not tracked, there are no parameters associated with the wrist which merely acts as an end-point that defines the forearm segment.

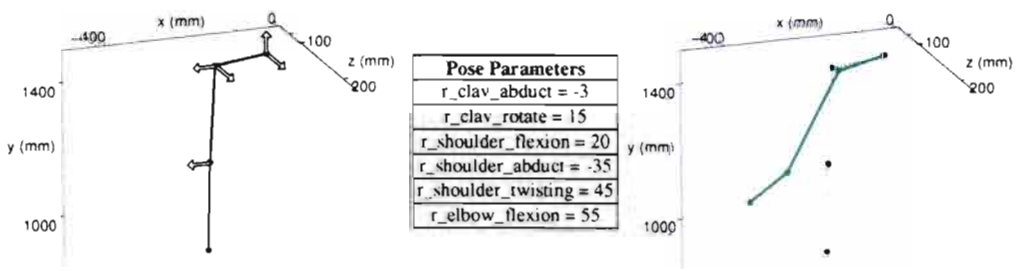
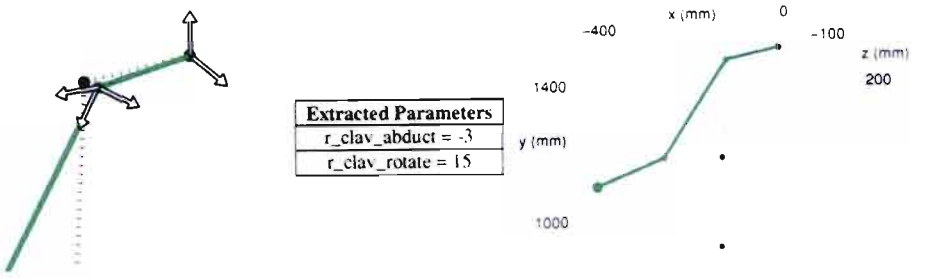
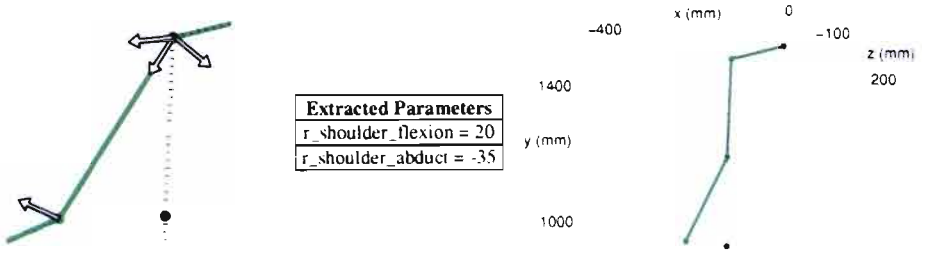


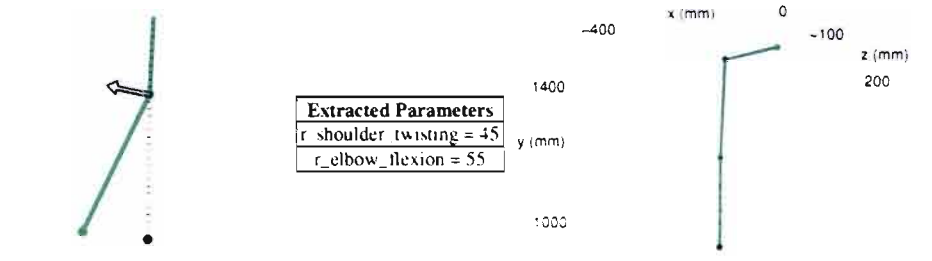
Figure 6.11: Generation of a synthetic pose of the right arm. On the left is the right arm in the default position, in the middle are the applied pose parameters, on the right is the transformed arm (solid line) as well as the default pose (dotted line).



(a)



(b)



(c)

Figure 6.12: Steps 1 (a), 2 (b) and 3 (c) of the BAP extraction process for the right arm. A close-up of the current joint and segment under examination is shown on the left, the recovered BAPs are given in the middle, and the effect of the inverse rotation is shown on the right. Default posture of the arm is indicated by dotted lines.

6.5 Summary

This chapter has discussed the final aspect of the model-based coding scheme presented in this thesis, the definition and extraction of the parameters describing the pose of the 3-D skeleton model. The pose parameters have been defined such that they comply with the MPEG-4 standard. The MPEG-4 standard is a natural choice for this research as it has been designed with low bit-rate applications in mind and it defines specific tools for the modelling and animation of virtual humanoids. The virtual character is represented in the MPEG-4 scene by the FBA scene graph which contains nodes that are able to change the physical characteristics of the model (i.e. customise it to a particular actor) and to animate it. When the same animation parameters are applied to different MPEG-4 compliant models they should generate reasonably similar motion regardless of whether they are applied to photorealistic or cartoon like models. The model used in this work is a 31 DOF simplification of the MPEG-4 skeleton. These 31 parameters are extracted from the 3-D data coming from the human motion capture system. The BAP extraction is performed sequentially for each joint starting at the root of the skeleton. The final result is a set of pose parameters that describe the posture of a subject in the image sequence at a given frame. Low bit rates are achieved because only 31 parameters need to be sent over a channel in order to reconstruct the motion at the receiver.

Chapter 7

Results and Discussion

In this chapter the performance of the motion reconstruction system is evaluated through a series of experiments. The objective of the system is to recreate human motion with enough precision to be convincing to most human observers. A quantitative evaluation of a motion reconstruction system requires human motion ground truth data. The ground truth data is typically acquired by either a commercial system [153] [154] [155] [156] such as the ones described in Chapter 2, or by generating synthetic sequences using a human model rendering package [80] [157] such as Curious Labs Poser [158], that allows the user to model and animate a virtual character. Often neither of these options is available to researchers who typically employ *"heuristic visual inspection to judge their results"* [157]. There was no ground truth available to test the presented system. To assess and compare the performance of the system to previous research, the various components were considered separately. Their accuracy evaluated and the effects of the errors on the overall system examined.

Figure 7.1 reveals the error propagation within the complete system. Starting at the output of the system and working backwards, the accuracy of the reconstructed motion is dependant on the precision of the extracted pose parameters. The pose parameters are extracted from the 3-D motion data and consequently any errors in the tracked data are translated to the output. The ability of the pose parameters to convey the motion

information is also dependant on the modelling of the human body and how the model fits into the data. However the focus of attention is on the 3-D tracking system and the precision with which points are tracked in 3-D space. The success of the overall system rests on the ability to track points in video sequences and the tracking precision offers a quantitative comparison to previous research. The tracking process relies on accurate image matching to establish temporal and spatial correspondences. It is affected both directly and indirectly by calibration errors through imprecise epipolar geometry and inaccurate 3-D reconstruction of the model respectively.

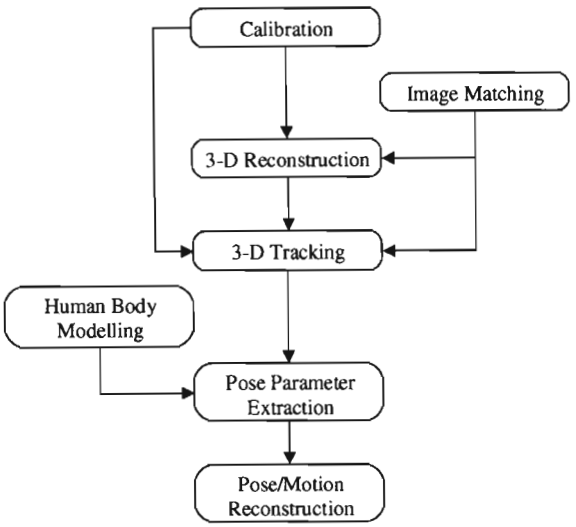


Figure 7.1: Error propagation in the complete system.

Section 7.1 provides a brief discussion of the calibration results which are given for the sequence that was used to test the system's motion reconstruction performance. Section 7.2 evaluates the performance of least squares matching under various conditions and compares the results to normalised cross-correlation. Section 7.3 demonstrates the system's ability to accurately reconstruct 3-D measurements and highlights the necessity of sub-pixel accuracy in 2-D correspondence generation for the triangulation process. Section 7.4 evaluates the performance of the 3-D tracking algorithm. Once again sub-pixel matching accuracy is shown to be necessary for successful tracking, and tracking failure of a coarse matching technique is illustrated using normalised cross-correlation. The limits of the tracking algorithm are discussed and results for full body tracking are given. Section 7.5

presents a visual comparison between the tracked and reconstructed model. Extracted parameters for the whole test sequence are shown for a selected joint and a complete set of animation parameters describing a specific pose is given. Performance results of the complete system are shown in Section 7.6.

7.1 Camera Calibration

The calibration results presented in Table 7.1 and Table 7.2 are for the "walking" sequence which was used to demonstrate the performance of the complete system. Both the intrinsic (Table 7.1) and extrinsic (Table 7.2) parameters show typical calibration values acquired for the testing sequences throughout the course of the research. The calibration was performed in two steps, first each camera was calibrated separately to generate the intrinsic parameters, and then the extrinsic parameters of the stereo setup were computed. The focal length is expressed in units of horizontal and vertical pixels f_x, f_y which is computed by f/s_x and f/s_y respectively. The relative rotation between two views is given in terms of a three parameter vector. The rotation matrix is obtained by the Rodrigues' rotation formula [159] described in Appendix B. Although CCD cameras are usually capable of spatial accuracy greater than 1/50 pixel, due to various error sources affecting image formation, this accuracy is not easily attained [160]. The accuracy in camera calibration is often quantified by how well it can measure the 3-D world. There are two basic approaches to quantifying the calibration error, either the 3-D points are projected onto the image plane and the pixel distance to the corresponding 2-D point is measured (forward-projection), or the 2-D points are back-projected into space to measure the distance between the 3-D points. Zhang's calibration method [112] [114] minimises the pixel error criterion to estimate the camera parameters. Although this does not guarantee the accuracy of the physical camera parameters, the composite effect provides satisfactorily accurate 3-D measurements, which is the main requirement of most computer vision applications. The reader is referred to [161] for an analysis and discussion of physical camera parameter accuracies. The planar pattern calibration process compared to more elaborate methods such as Tsai [162] produces relatively high accuracy with trivial effort [163]. The

calibration results for the setup used to capture test sequences in this evaluation generated acceptable accuracy, with the pixel error below the typical precision of 0.2 pixels [164].

Table 7.1: Intrinsic camera parameters of the multi-view setup

	Left Camera		Middle Camera		Right Camera	
parameter	value	uncertainty	value	uncertainty	value	uncertainty
f_x	545.306	1.580	542.878	1.339	542.417	1.336
f_y	545.093	1.557	542.518	1.387	541.438	1.329
o_x	321.036	2.578	355.193	2.415	305.984	2.347
o_y	202.697	1.803	247.899	1.805	226.293	1.664
pixel error	[0.1390 0.1265]		[0.1330 0.1254]		[0.1184 0.1271]	

Table 7.2: Extrinsic camera parameters of the multi-view setup

	Left Camera with respect to Middle Camera	
parameter	measurement	uncertainty
translation (mm)	[-98.5670 -35.9327 10.1271]	[0.1762 0.1461 0.1122]
rotation	[-0.0457 0.0635 -0.0025]	[0.0003 0.0003 0.0002]
	Right Camera with respect to Middle Camera	
parameter	measurement	uncertainty
translation (mm)	[103.8563 -27.4050 3.7135]	[0.1570 0.1319 0.1021]
rotation	[-0.0293 -0.0400 -0.0212]	[0.0002 0.0003 0.0002]
	Right Camera with respect to Left Camera	
parameter	measurement	uncertainty
translation (mm)	[203.6011 6.9546 4.279]	[0.2409 0.1687 0.1509]
rotation	[0.0157 -0.1040 -0.0168]	[0.0003 0.0004 0.0002]

7.2 Image Matching

Image matching is the core of the presented human motion capture system. The tracking algorithm relies on LSM to find temporal and spatial correspondences necessary for tracking points in multi-view image sequences. The following experiments demonstrate the ability of LSM to find accurate correspondences despite large image differences caused by geometric distortion, radiometric distortion and noise.

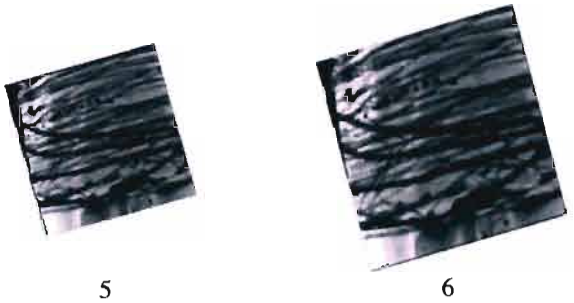
The test images used in the experiments are displayed in Figure 7.2. The template image (Figure 7.2(a)) is a large patch taken from a frame of the "walking" sequence in order for the image data to be representative of the sequences used in motion tracking. The search images were generated from the template by known transforms, thus providing ground truth data for the assessment of matching results. Search images 1-4 (Figure 7.2(b)) have been rotated by 2.5, 5, 15 and 20 degrees respectively. Images 5 and 6 (Figure 7.2(c)) were generated by scaling search image 3 with scaling factors of 0.9 and 1.25 respectively. Zero mean Gaussian noise with variances of 0.00125, 0.0025 and 0.005 was added to image 2, to produce images 7, 8 and 9 respectively (Figure 7.2(d)). Image 10 has been sheared horizontally and image 11 has been rotated, corrupted by noise (zero mean Gaussian, variance = 0.002) and its brightness and contrast were also modified (Figure 7.2(e)). This set of test images encompasses the various image distortions caused by inter-frame motion, different viewpoint and random noise encountered in the video sequences. The only deviation from real data is that the images represent perfectly planar surfaces, although this aspect is offset to a large degree by exaggerating the image differences in comparison to those found between the three views or in subsequent frames throughout this work.

The matching tests were performed using both least squares matching and normalised cross-correlation for comparison purposes. To produce more accurate correspondence estimates with NCC, equations 4.4 and 4.5 were used to generate sub-pixel locations. The summarised results in Table 7.3 emphasize the need for true sub-pixel accuracy when exact correspondences are required. The term "true sub-pixel accuracy" refers to a method that is able to account for the various causes of differences between two images rather than acquiring sub-pixel location through interpolation. NCC is only able to produce good re-

(a)

1 2 3 4

(b)



5 6

(c)

7 8 9

(d)

10 11

(e)

Figure 7.2: Images used in matching experiments. (a) Template Image, (b) Search Images 1, 2, 3, 4 (rotated), (c) Search Images 5, 6 (rotated, scaled), (d) Search Images 7, 8, 9 (rotated, scaled, zero mean Gaussian noise), (e) Search Images 10 (horizontal shear), 11 (rotated, brightness & contrast modified, zero mean Gaussian noise).

sults when the difference between the conjugate images is small as was the case for search images 1, 2 and 10. Under these conditions, the resulting surface of the cross-correlation coefficient over the search space is characterised by a dominant peak (Figure 7.3(a)). The remaining search images are too distorted to be able to locate a correspondence purely based on a similarity measure, because the cross-correlation coefficient surface becomes flattened disguising the correct location among a number of possible candidate peaks (Figure 7.3(b)).

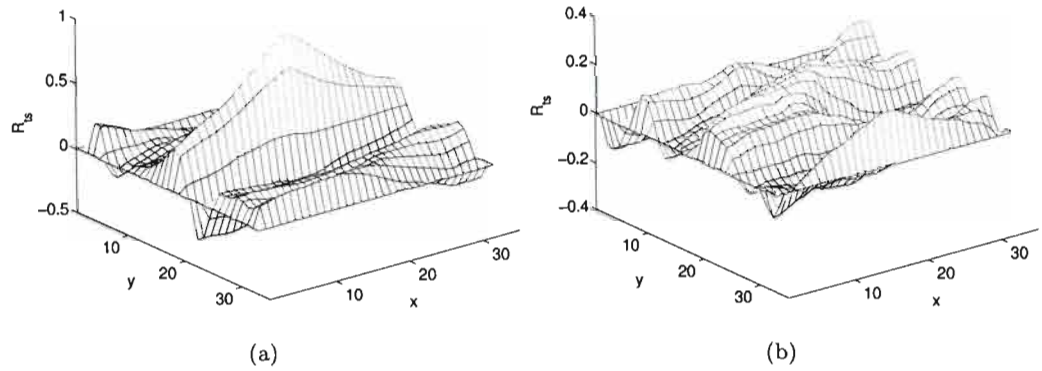


Figure 7.3: Surface of the cross-correlation coefficient over the search space for (a) good correlation (search image 1), (b) bad correlation (search image 11).

Least squares matching on the other hand produced accurate results in all situations. The affine transform (4.9) accounts for the geometric differences between the template and search image. The results for the rotated set of images (images 1-4) display a consistent pixel error, with the increasing rotation largely affecting the number of iterations required to satisfy the convergence criteria (which has been set in these tests at 0.01 for shifts and 0.02 for scales). Images 5 and 6 tested the ability of LSM to recover scaling differences. The outcome interestingly indicates that LSM is able to recover a scale increase more easily than a scale reduction. This could be attributed to the fact that in scale reduction image data is lost and conversely when the patch is enlarged, additional data is created. This phenomenon has a significant effect on human motion tracking in situations when the subject moves away from the cameras and is revisited in Section 7.4. The addition of a significant level of noise to the search image intuitively has negative effects on the precision

of the result. However, because the **A** matrix (4.15) is calculated from the average of the search and template patch (as described in Section 4.3.2) the effects of noise are reduced, and accurate results are obtained even with a high degree of image corruption experienced by search image 9. In addition to rotation and noise, the radiometric property of image 11 was also modified. By using r_0 and r_1, the additive and multiplicative coefficients of radiometric correction introduced in 4.10, LSM is able to correct the brightness and contrast differences and still produce a location within 1/20 of a pixel.

Table 7.3: Comparison of least squares matching and normalised cross-correlation matching results

	LSM			NCC	
search image	pixel error	R_{ts}	iterations	pixel error	R_{ts}
1	0.012	0.991	5	0.057	0.947
2	0.018	0.994	5	0.178	0.812
3	0.007	0.993	8	0.467	0.273
4	0.016	0.993	10	6.187	0.274
5	0.033	0.986	10	17.403	0.268
6	0.001	0.993	8	16.689	0.252
7	0.018	0.981	7	0.700	0.276
8	0.023	0.971	7	0.453	0.258
9	0.050	0.946	7	2.221	0.259
10	0.001	0.999	5	0.160	0.905
11	0.035	0.965	7	1.837	0.262

The results presented in this section establish the sub-pixel accuracy of LSM under various situations. The ability of LSM to account for geometric and radiometric distortion as well as to reduce the effects of noise makes it a suitable method for generating accurate correspondences. The significance of accurate correspondences is demonstrated in Section 7.3

and Section 7.4.

7.3 3-D Reconstruction

3-D reconstruction is the process of estimating the shape and position of 3-D objects from their images. As described in Chapter 3, a point's location in 3-D space is established at the intersection of the rays originating at the COP of the two cameras and passing through the 2-D correspondence in the respective image plane. By performing camera calibration the intersection of the two rays in 3-D space may be found by simple triangulation using the recovered intrinsic and extrinsic camera parameters. However, due to incomplete modelling of the image formation process, calibration and 2-D correspondence inaccuracies, an uncertainty is introduced into the 3-D reconstruction process. According to Hartley and Zissermann [165] the uncertainty of reconstruction is dependant on the angle between the two rays. As the rays become more parallel the 3-D points are less precisely localised. The shape of the uncertainty region is shown in Figure 7.4. The centralised camera setup that aids the matching process by minimising spatial distortion between the respective views, in this case hinders the reconstruction process by introducing a greater uncertainty.

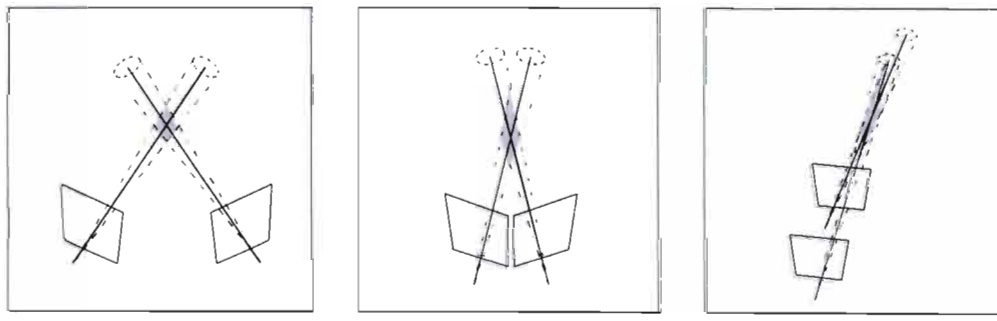


Figure 7.4: Regions of reconstruction uncertainty depending on angle between rays.

The analysis of the reconstruction uncertainty focused on the effects of 2-D correspondence error since Zhang's calibration method [114] has been shown to produce results with high accuracy [163]. Observing the regions of reconstruction uncertainty in Figure 7.4, it is evident that the greatest challenge is to recover depth (z-component). Depth estimation

is a difficult task because the images on their own provide no information regarding depth, which needs to be inferred from the 2-D data. Since the recovery of the "lost" dimension in the data capture process is a vital aspect of 3-D human motion capture, the effects of correspondence errors was investigated to determine the necessary accuracy of the matching process. In the experiment, an arbitrary location on a subject was tracked by LSM as he moved away from the cameras. The point's 3-D location was triangulated in every frame and the resulting 3-D trajectory was considered as ground truth. A matching error of 0.5, 1 and 1.5 pixels was introduced to the tracked 2-D locations of one of the views, producing three additional 3-D trajectories. The distance between each of the three tra-

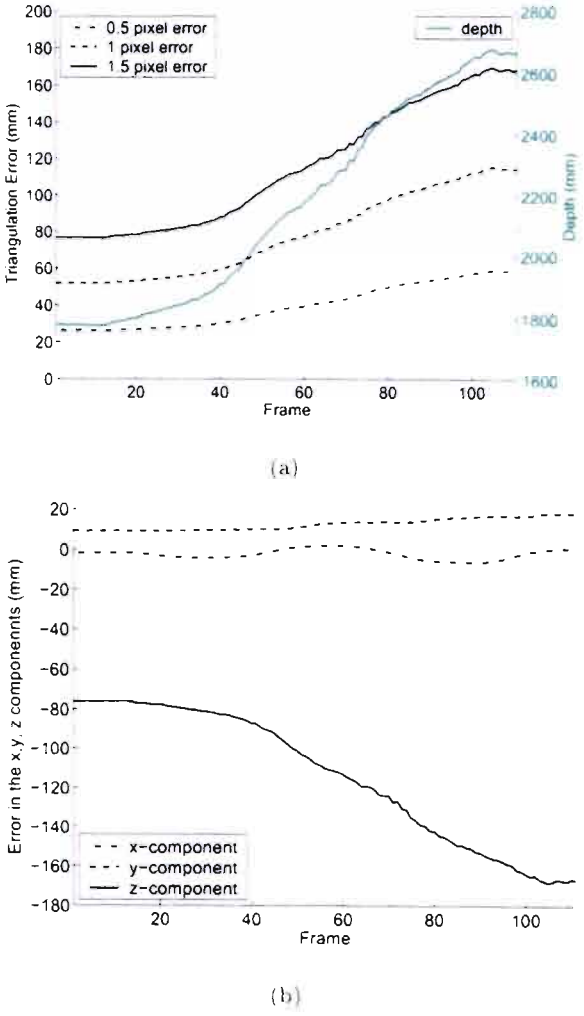


Figure 7.5: Effects of correspondence error on the 3-D reconstruction process.

jectories and the "ground truth" trajectory represents the reconstruction error displayed in Figure 7.5(a). The data corresponding to the right hand y-axis is the distance of the tracked point from the camera at each frame of the sequence. The relationship between the increasing 3-D error and depth confirms Hartley and Zissermann's hypothesis that reconstruction uncertainty increases as the rays become more parallel. Figure 7.5(b) indicates that the main contribution to the reconstruction error comes from the z-component estimate as was implied by Figure 7.4. A more important conclusion arising from the experiment is that a small pixel error (0.5 pixels) in the 2-D correspondence can cause a 60 mm error in the corresponding 3-D location. In terms of 3-D HMC, a 60 mm joint location discrepancy is detrimental to the motion reconstruction process. The 3-D errors introduced by the 1 and 1.5 pixel inaccuracies are clearly unacceptable in a 3-D motion reconstruction application.

To test the ability of the implemented system to recover 3-D information, two points defining a specific length on a ruler were matched in the right and middle views by LSM at various depths and orientations. Selected images from this experiment are displayed in Figure 7.6. The distance between the two points was 348 mm and details of the recovered 3-D data are summarised in Table 7.4. These results confirm that the accuracy of the

Table 7.4: Results of the 3-D reconstruction experiment

	measurement	absolute error
min (mm)	347.04	0.01
max (mm)	349.06	1.06
mean (mm)	348.27	0.52
σ (mm)	0.52	0.27

calibration results from Section 7.1 and the precision of LSM can accurately recover 3-D information (on average within almost 1/2 mm). Aside from the high accuracy of the recovered 3-D data an important outcome of the results in terms of human motion capture is also the consistency of the recovered lengths. This aspect of the 3-D reconstruction process pertains to the recovery of the segment lengths of the skeleton model and is

elaborated upon in Section 7.4.

Figure 7.6: Selected images from the 3-D reconstruction accuracy test (for middle camera).

In summary of the 3-D reconstruction experiments, it is clear the LSM is necessary for accurate triangulation of points in 3-D space. This statement is amplified when the desired point lies further away from the cameras, an inevitable situation when tracking human motion. The low precision of a matching technique such as cross-correlation established in Section 7.2 may lead to large 3-D errors that render the motion data ineffective. Although the centralised camera setup used to minimise image differences within the three views leads to a larger reconstruction uncertainty region, the accurate LSM results ensure reliable recovery of 3-D data.

7.4 3-D Tracking

The 3-D tracking process employs the previously discussed components (viz. camera calibration, image matching, 3-D reconstruction) to generated 3-D locations of desired points in each frame. Consequently the tracking process is susceptible to errors arising in any one of the three components. Thus far it has been shown in Section 7.2 that LSM can provide accurate correspondences in two images and in Section 7.3 that the correspondences together with calibration results are able to generate reliable 3-D data. Instinctively this leads to the assumption that by repeating the process of matching and triangulation, the tracking algorithm will always produce reliable motion data. However in real world situations systems always reveal hidden limitations that cause a deviation from theoretical predictions and the presented work is no exception. In this section the 3-D tracking performance is evaluated. First, similarly to Section 7.3 where the necessity of LSM for accurate reconstruction was established, LSM is shown as a crucial component for reliable temporal tracking. Next, the precision of 3-D tracking is demonstrated followed by a discussion of the tracking algorithm limitations and recovered human motion data.

7.4.1 Sub-Pixel Accuracy for Tracking

Throughout this thesis it has been argued that the precision of LSM is vital for the success of the 3-D human motion data recovery and ultimately motion reconstruction. Section 7.3

has already proven that accurate spatial correspondences are necessary for reliable 3-D data generation. This is only one part of the HMC system, as the points also have to be tracked throughout the video sequence, necessitating accurate temporal correspondences as well. To demonstrate the requirement of sub-pixel accuracy in temporal matching, two experiments were performed to show tracking divergence and tracking failure caused by inaccurate matching. Once again the comparison was made using NCC as the "less accurate" matching technique. In the two test sequences the tracking strategy was the same for both NCC and LSM, hence the failure/success can be attributed entirely to the matching technique.

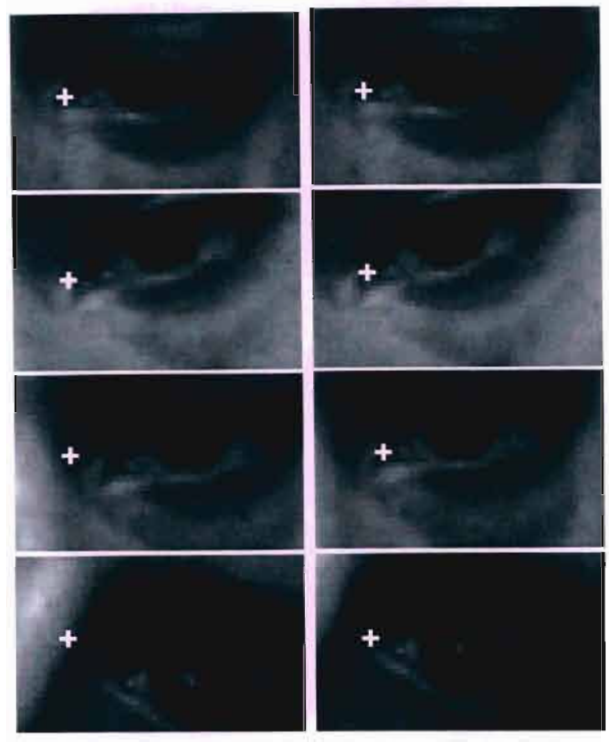
Figure 7.7 displays five selected frames from the first test sequence involving a moving arm. The top image represents frame 1, with the left column showing tracking with NCC and the right column with LSM. Although multiple views were used to allow spatio-temporal verification, only the middle view images are shown. The wrist point in the NCC tracking sequence starts to diverge from its original location early in the sequence and on the subsequent downward motion of the arm becomes "stuck" on the background. When later the arm moves past that point once more, it latches onto the hand, eventually becoming "stuck" again in the background where it remains for the rest of the sequence. LSM on the other hand is able to maintain track for the duration of the test sequence. The failure is most evident at the wrist because out of the three tracked joints the wrist moves the fastest. The fast motion translates to large differences between subsequent frames, which as shown in Section 7.2, NCC is unable to cope with. The remaining two points have also drifted slightly from their original locations, however this is not evident from the displayed images. This example demonstrates that it is necessary for the matching process to account for the geometric differences arising from motion between subsequent frames in order to successfully track points throughout the sequence.

The second test sequence tracked the corner of an eye as shown in Figure 7.8, where NCC tracking is once again displayed in the left and LSM tracking in the right column. The rigid motion of the head results in smaller inter-frame differences opposed to the motion of the wrist. In this experiment NCC does not fail completely, but diverges from the original location. This occurs because even the small errors (in cross-correlation terms),

(a) (b)

Figure 7.7: (a) Tracking failure due to inaccurate matching, (b) successful tracking with LSM.

such as the 0.178 pixel error in the matching example with search image 2 in Section 7.2, accumulate during the sequence causing a drifting effect. Since LSM matches points with much higher precision it is able to remain on track throughout sequence.



(a) (b)

Figure 7.8: (a) Tracking divergence due to inaccurate matching, (b) successful tracking with LSM.

7.4.2 3-D Tracking Precision

In the absence of ground truth motion data, the precision of the 3-D tracking algorithm was examined by a crude method of moving a target over a ruler (Figure 7.9). The target was a dark dot on top of a permanent marker pen. It was tracked as it moved up and down the ruler and every time it crossed a marked position (every 50 mm) a measurement was taken. The measurement was taken with respect to the target's initial position over the 0 mm mark. The measurement coming from the tracking system was compared with the actual displacement to determine the tracking error.

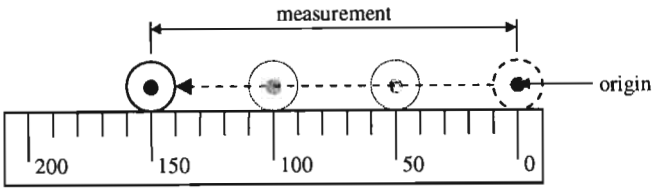


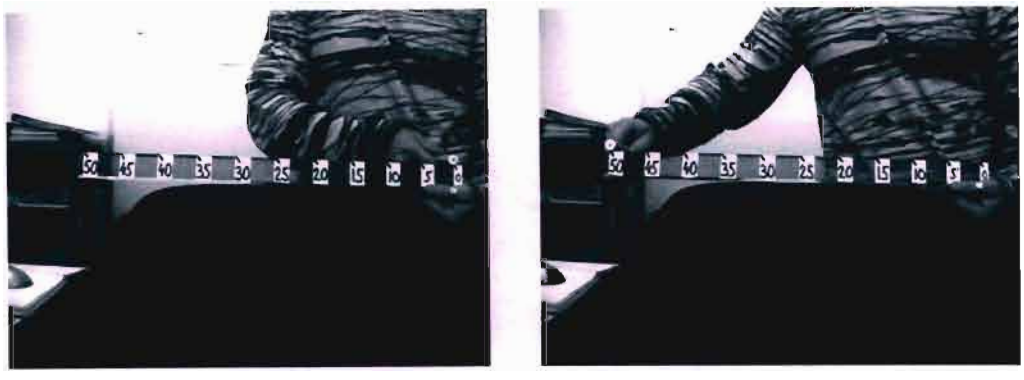
Figure 7.9: Evaluation of tracking accuracy by tracking a target over known distances.

This process was performed on three test sequences with motion in the x-direction, y-direction and z-direction (Figure 7.10). To compensate for the motion of the reference point, in addition to tracking the target, a point on the ruler at 0 mm was also tracked. Table 7.5 contains the tracking error for all three sequences.

Table 7.5: 3-D tracking error

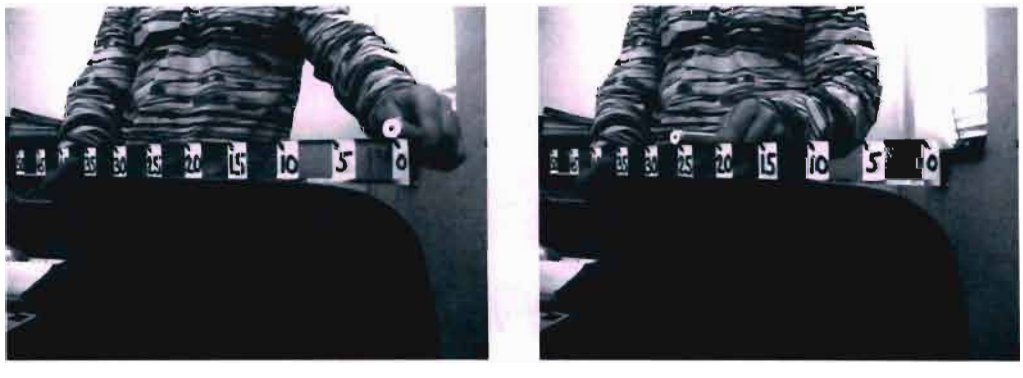
	Absolute Error		
sequence	max(mm)	mean (mm)	σ (mm)
x	2.53	1.60	0.60
y	3.14	1.55	0.98
xz	2.00	0.83	0.64

A surprising outcome is the accuracy achieved for the sequence with motion in the xz-direction, because from the discussion in Section 7.3 one would expect the least precise results in that particular experiment. It must be noted that there are several sources of error in these tests including the visual estimation of the point where the marker crosses a 50 mm boundary, the determination of the reference point, insufficient compensation for the motion of the reference point etc. Although these variables may have some affect on the exact accuracy of the results, the conclusion that the tracking algorithm is able to track points in 3-D space with millimetre accuracy is still valid.



(a)

(b)



(c)

Figure 7.10: Test sequences used for tracking accuracy evaluation, (a) motion in the x-direction, (b) motion in the y-direction, (c) motion in the xz-direction.

7.4.3 3-D Tracking Limitations

When considering tracking algorithms, it is important to consider both the tracking accuracy and limiting factors. In this instance, the limiting factors refer to influences that cause tracking failure or introduce an error that has not yet been discussed under the various sections. The tracking process relies on LSM and consequently a matching failure across multiple views translates to tracking failure of the HMC system. Thus to examine the limitations of the tracking process one must look at the limitations of LSM. There are three factors that may cause LSM failure; occlusion, bad initial estimate and insufficient signal in the matching patches. It has been established at the beginning of the thesis that occlusion is not dealt with in the current implementation and accordingly a constraint was placed on the tracked motion eliminating occlusion from the problem. The 3-D prediction scheme introduced in Chapter 5 together with a NCC refinement produced reliable estimates therefore the focus of this section is on the signal content (matching data) factor. Three aspects were examined; the violation of the assumption that matching patches represent locally planar surfaces, blurring of image data due to fast object motion and image data loss caused by excessive scale reduction.

Figure 7.11 illustrates the scenario where the image patches do not represent locally planar surfaces. The images are taken from the left and right views of the sequence used to determine the accuracy of 3-D tracking. As the origins of the image patches are defined at the edge of the ruler, the signal content making up the top half of each patch belongs to the image background which is different in each view. Only the warp between the portions belonging to the ruler can be described by the affine transform and subsequently matching fails in an extreme situation such as the one depicted in Figure 7.11. This scenario occurs to a lesser degree at the boundary of clothing items (e.g. sleeves) making tracking of hands and feet more challenging. Reducing the patch size decreases the undesirable effects, although very small patches may lead to less accurate results or matching failure as well.

The blurring effect caused by fast motion is demonstrated with three successive frames in Figure 7.12. Blurring is associated with the imaging hardware, namely the shutter speed and light sensitivity of the sensor. The cameras used in this work are capable of high

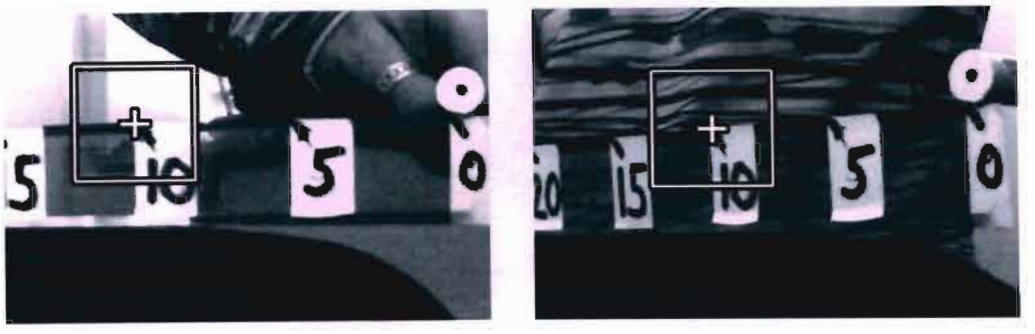


Figure 7.11: Image patches in a stereo pair violating the assumption of locally planar surfaces.

shutter speeds, up to 1/8000s [166]. However, when a high shutter speed is used the object must be well illuminated in order to produce usable image data. Although additional lighting was employed (in the form of three halogen lamps), in some instances blurring was evident at the extremities of limbs (forearms and calves). Motion blur degrades the gradient information within the image patch necessary for LSM to recover the affine parameters. Depending on the intensity of the motion blur the matching accuracy will decrease and in severe cases fail.

Figure 7.12: Blurring effect of fast human motion.

A subject's relative motion away from the camera results in the scaling down of his/her image in the video sequence. The scaling is associated with loss of resolution of the subject's image and ultimately is responsible for setting the maximum depth of the workspace, because each HMC system requires some minimum resolution to correctly interpret the image data (for this reason hands can not be tracked in full body motion capture systems). Systems with lower resolution imaging devices experience more pronounced effects of scale reduction at a given depth, and consequently are constrained to applications in smaller

workspaces. The typical size of a workspace in commercial systems ranges from 1-4 metres in radius [53]. The maximum depth displacement in the experiments performed during this research was 2.8 metres and due to low image resolution (640x480 opposed to resolutions of up to 2352x1728 of Vicon systems) some tracking divergence was experienced. The

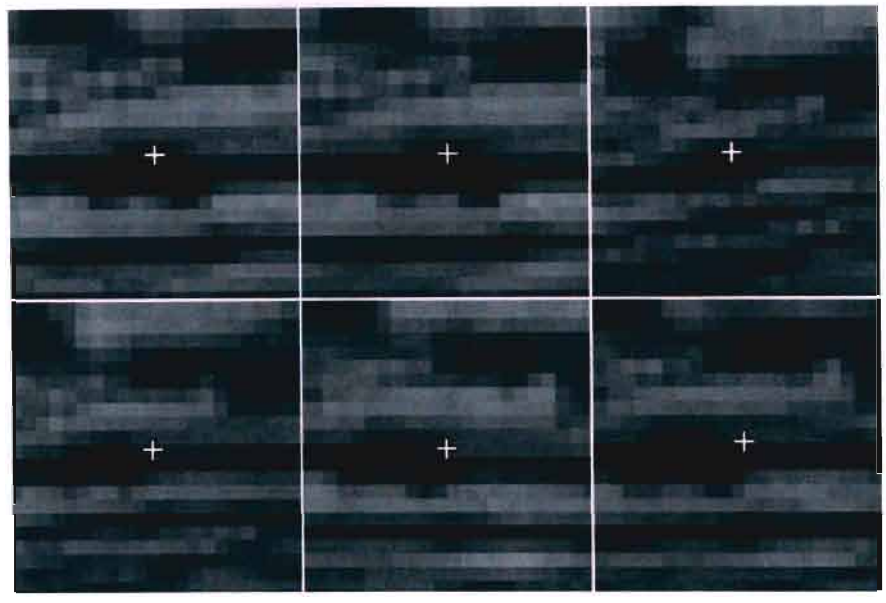


Figure 7.13: Tracking divergence due to large scale reduction.

scale reduction has an unfavourable effect on the matching process because image data of the tracked feature is lost. Despite good matching results, a point tracked in an image patch that has undergone a cycle of scale reduction and subsequent scale enlargement will drift from its original location. This drifting effect (tracking divergence) is illustrated in Figure 7.13. The top three images display the scale reduction of an image patch as the subject moves away from the camera and the bottom images show the subsequent enlargement as the subject returns to his original location. The image data loss is so severe that despite good LSM indicators (high R_{ts}, low σ_0) the matched result contains an error that accumulates during the sequence causing a 4 pixel drift by the end. Unfortunately this error is difficult to identify because the divergence occurs in all three views, thus is not discovered by the verification process discussed in Section 5.3.5. The end result is an uncertainty in the tracked 3-D data despite good matching and reconstruction. The magnitude of this error is small in comparison to that caused by 2-D correspondence error

examined in Section 7.3. In the example shown in Figure 7.13 the discrepancy between the tracked location at the beginning and end of the sequence is 13.5mm.

7.4.4 Human Motion Data

The accuracy and limitations of the tracking algorithm have now been examined and it remains to evaluate the system's performance on the application to full body motion tracking. Figure 7.14 displays four frames from the "walking" test sequence. Only the middle view is shown with the corresponding tracked 2-D skeleton overlaid and the reconstructed 3-D skeleton model. All 18 joints were successfully tracked in the 255 frame sequence for motion performed at a depth of 1.8 to 2.7 metres.

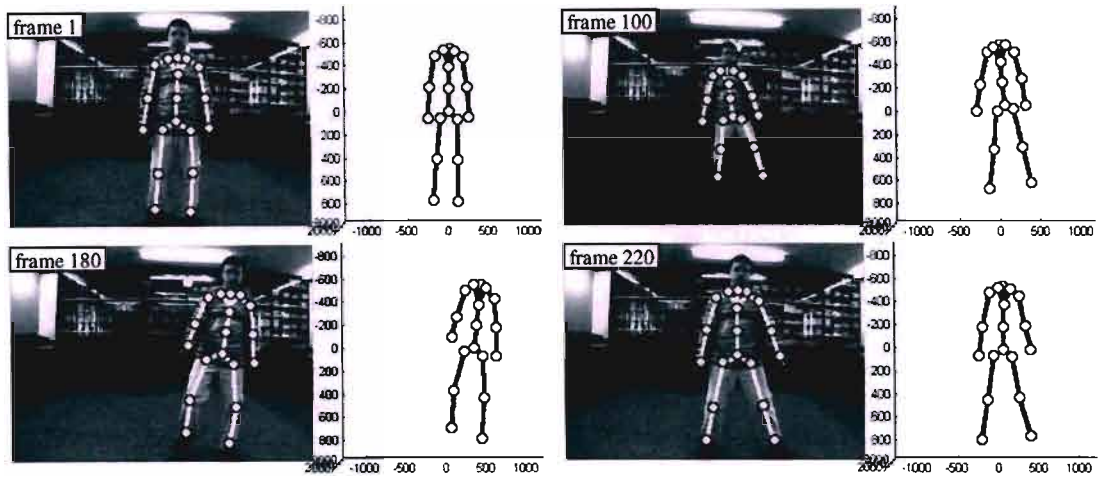


Figure 7.14: 3-D tracking of human motion. Only the middle view of the multi-view video sequence is shown, with the corresponding 3-D model fit to the tracked 3-D points.

There is no ground truth to which the tracked 3-D skeleton can be compared to. In this situation, the best way to evaluate the motion data is to examine the consistency of the separate segments. The experiment in Section 7.3 that reconstructed a defined length at various locations in 3-D space displayed very consistent results. However, those matching patches consisted of well defined artificial planar targets, did not experience motion blur or large depth displacement as did patches in the "walking" sequence. Thus the assessment of segments recovered from the motion data (17 segments in total) reveals the effect of the

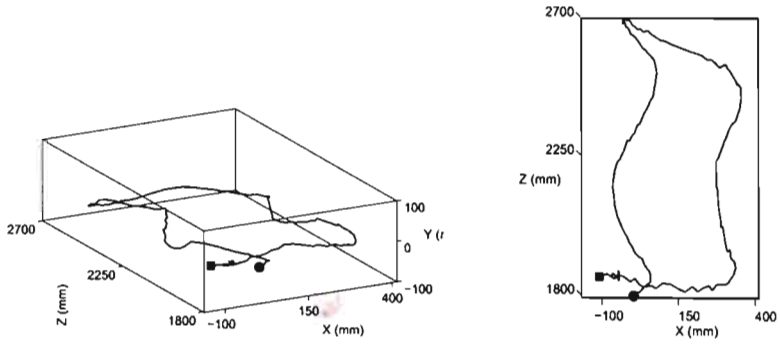
factors described in Section 7.4.3 on the precision of the HMC system.

Table 7.6: Statistics of segment length recovery in the "walking" sequence (segment names are defined in Appendix C)

	Measurement				% Deviation	
segment	min (mm)	max (mm)	mean (mm)	σ (mm)	max	mean
r_farm	213.53	260.58	244.47	9.46	12.66	2.97
r_uarm	232.73	265.06	251.54	5.26	7.48	1.50
r_col	99.02	115.52	107.66	2.64	8.02	1.75
r_s2v	144.25	164.66	153.09	3.61	7.56	1.85
l_s2v	138.14	159.59	148.87	4.21	7.21	2.30
l_col	112.10	127.88	117.76	3.74	8.59	2.80
l_uarm	226.70	261.26	234.55	5.96	11.38	1.97
l_farm	189.67	250.75	229.76	10.25	17.45	3.35
u_spine	132.94	171.92	150.73	7.17	14.06	3.93
m_spine	172.84	181.79	176.66	1.84	2.90	0.87
l_spine	164.48	178.64	171.03	3.21	4.45	1.51
r_shin	285.06	383.72	343.77	18.31	17.08	4.48
r_thigh	328.45	353.54	341.55	6.05	3.84	1.46
r_pel	110.01	132.00	121.17	3.84	9.20	2.48
l_pel	102.59	130.18	119.45	5.88	14.11	3.97
l_thigh	309.67	342.53	324.55	6.39	5.54	1.62
l_shin	305.19	363.81	336.65	9.66	9.34	2.25

Table 7.6 presents the statistics of each segment throughout the "walking" sequence. The % deviation provides the most informative manner of evaluating the results. The necessary

segment consistency for visually accurate animation was studied by J. Harrison *et al.* [38]. After a series of experiments they conclude that human observers are unlikely to perceive segment changes of 2.7% even when the change of length is expected and the observe is paying full attention to the segment. Segment changes of up to 20% may go unnoticed if the observer is not focusing on the particular part. Furthermore, the study shows that changes are more difficult to perceive for fast motions and out-of-plane rotations. 11 out of the 17 recovered segments from the "walking" sequence on average experienced a segment variation of less than the perceivable amount. Segments such as the forearms and shins undergo fast out-of-plane motion, concealing the small length variations. The maximum percentage deviation of all segments is below the 20% level that signifies the maximum unnoticeable change. The results from Table 7.6 validate the suitability of the implemented HMC system for visually accurate motion data.



(a) (b)

Figure 7.15: 3-D path of the sacroiliac in the "walking" sequence, (a) rotated view, (b) top view (x-z plane).

The tracked 3-D data may be used apart from pose recovery for the analysis of the movement. Figure 7.15 displays the subject's 3-D trajectory of the HumanoidRoot i.e. his motion in the scene. Each tracked point also has velocity associated with its trajectory. The three components of the HumanoidRoot's velocity vector in each frame are shown in Figure 7.16(a)-(c). The z-component is considerably noisier than the x- and y-components which can be attributed to the greater uncertainty associated with depth recovery. The motivation for smoothing the noisy velocity component is to produce smoother motion.

Although this step does not have a significant effect on the motion data in terms of the consistency of segment lengths, it should provide for a more fluent output animation.

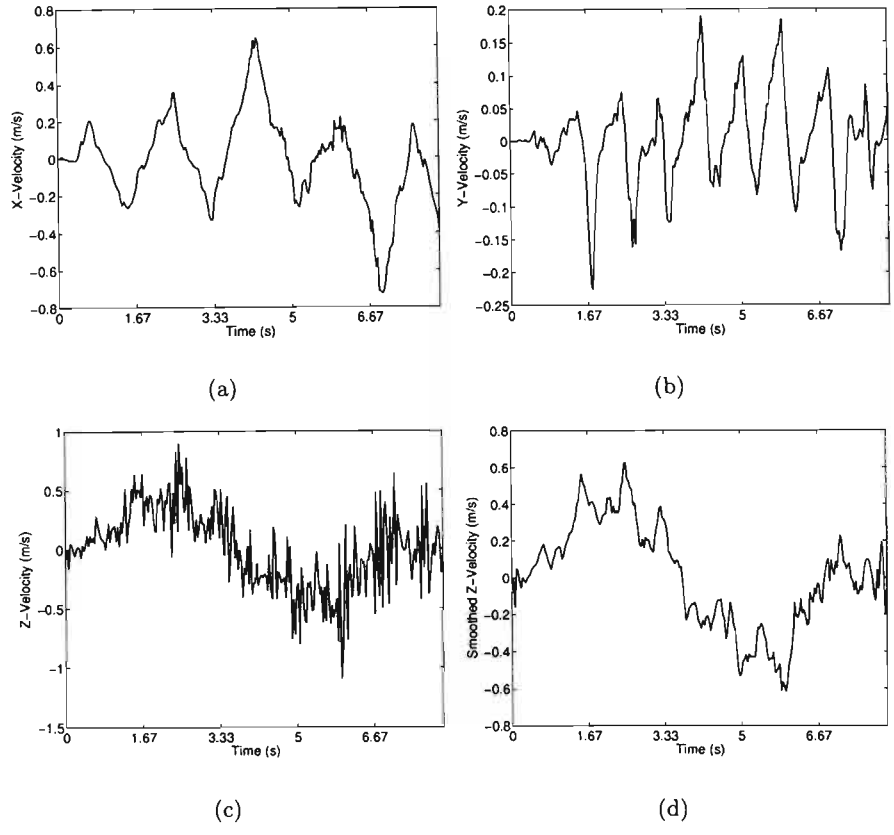
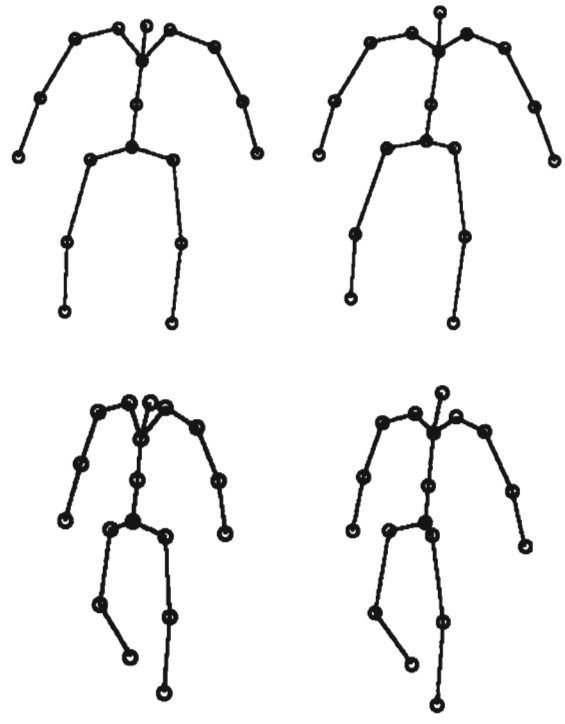


Figure 7.16: Velocity of the HumanoidRoot, (a) x-component, (b) y-component, (c) z-component, (d) smoothed z-component (using the Savitzky-Golay smoothing filter).

7.5 Pose Reconstruction

The human motion capture system tracks joint locations in 3-D space and fits a skeleton model to the data in every frame of the multi-view video sequence. The tracked skeleton is parameterised and 28 rotations together with 3 translations are extracted. These 31 pose parameters must adequately describe the pose so that when they transform the output model (at the decoder) the reconstructed posture must be visually similar to that of the subject in the corresponding frame of the input video sequence. Figure 7.17 shows two views of the tracked skeleton and the reconstructed model side by side. In the frontal

views the two models seem almost identical. However the rotated views reveal a difference in the torso area of the two models. This discrepancy is a result of the modelling stage of the virtual character.

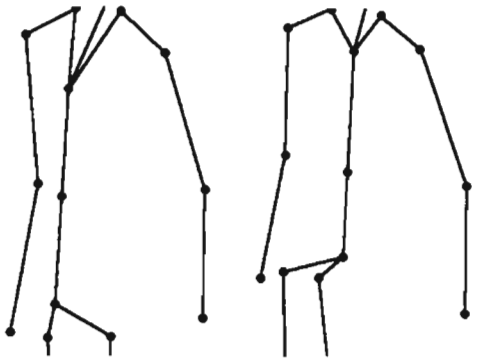


(a) (b)

Figure 7.17: Front and rotated views of (a) the tracked 3-D skeleton and (b) the reconstructed model at the decoder.

Although both the tracked and the output skeletons are defined by the same joints, the fundamental difference between them is that the former is essentially planar, whilst the latter accounts for the volumetric property of the human body. While the MPEG-4 standard defines the skeleton joints at the anatomically correct locations, the HMC system merely estimates the joint locations on the surface of the subject as it has no access to the actual joints (Figure 7.18). As a result, the tracked data does not account for the volumetric structure of the human body. The tracked spine is the main area affected by the estimation of joint locations as it is defined in front rather than at back of the torso. In terms of pose reconstruction however, the limbs are visually well represented with the

offset of the spine not impairing the reconstructed posture to a degree where the resulting motion would not be adequately conveyed.



(a) (b)

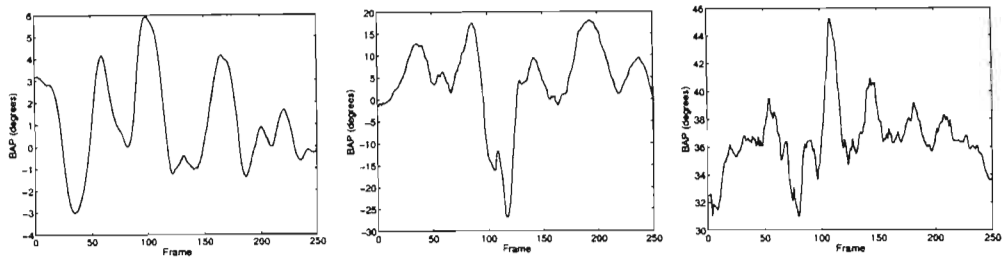
Figure 7.18: Difference between tracked model (a) and H-Anim based model (b) caused by lack of volumetric data.

Traditional coding schemes achieve compression by exploiting the stochastic properties of each frame. The model-based coding approach on the other hand only extracts and transmits useful information that is necessary to reconstruct the scene at the receiving end. In the case of human motion capture, the 31 pose parameters constitute the useful information necessary to transform the humanoid model in order to reconstruct the performed motion. The values of the parameters pertaining to the pose of the models shown in Figure 7.17 are presented in Table 7.7.

Figure 7.19 presents the BAP values (in degrees) of the 6th thoraic vertebrae throughout the "walking" sequence. These values represent the rotation in the coronal plane (a), about the vertical body axis (b) and in the sagital plane (c). The collection of the angular motion data of the tracked joints may be used in motion recognition applications [167].

Table 7.7: Extracted BAPs describing the subject's pose in one frame

BAP ID	BAP name	BAP value
1	sacroiliac_tilt	0.847
2	sacroiliac_torsion	13.484
3	sacroiliac_roll	-4.021
4	l_hip_flexion	-2.585
5	r_hip_flexion	-7.347
6	l_hip_abduct	3.450
7	r_hip_abduct	-11.767
8	l_hip_twisting	-28.357
9	r_hip_twisting	5.947
10	l_knee_flexion	-7.911
11	r_knee_flexion	-33.101
24	l_sternoclavicular_abduct	-13.223
25	r_sternoclavicular_abduct	15.428
26	l_sternoclavicular_rotate	21.266
27	r_sternoclavicular_rotate	-8.462
32	l_shoulder_flexion	-27.023
33	r_shoulder_flexion	-42.915
34	l_shoulder_abduct	41.472
35	r_shoulder_abduct	-32.776
36	l_shoulder_twisting	18.615
37	r_shoulder_twisting	-31.102
38	l_elbow_flexion	7.452
39	r_elbow_flexion	10.702
87	vt6_roll	2.321
88	vt6_torsion	-10.119
89	vt6_tilt	34.206
114	vl3_roll	-0.303
116	vl3_tilt	3.450
181	HumanoidRoot_tr_vertical	319.680
182	HumanoidRoot_tr_lateral	15.461
183	HumanoidRoot_tr_frontal	1931.290



(a) (b) (c)

Figure 7.19: Recovered BAPs (in degrees) of the 6th thoraic vertebrae, (a) vt6_ roll (rotation in the coronal plane), (b) vt6_torsion (rotation about the vertical body axis), (c) vt6_tilt (rotation in the sagital plane).

7.6 Complete System

The process of reconstructing human motion from multi-view video sequences with MPEG-4 compliant pose parameters is illustrated in Figures 7.20 to 7.24. The input to the encoder of the model-based coding system is the "walking" sequence recorded by three synchronised cameras. The analysis of the input sequence involves tracking manually defined joint locations in multi-image space using least squares matching. The tracked points in each view define a 2-D skeleton that represents the subject's 2-D pose in the image space. By triangulating corresponding points using camera parameters recovered through calibration, the joint locations in 3-D space are computed. A 3-D skeleton model is fitted to the 3-D motion data and subsequently 31 BAPs describing the model's pose are extracted. By applying the MPEG-4 pose parameters to an H-Anim compliant humanoid model, the model is transformed into the posture specified by the BAPs. A stream of BAPs animates the model, thus the subject's motion is reconstructed.

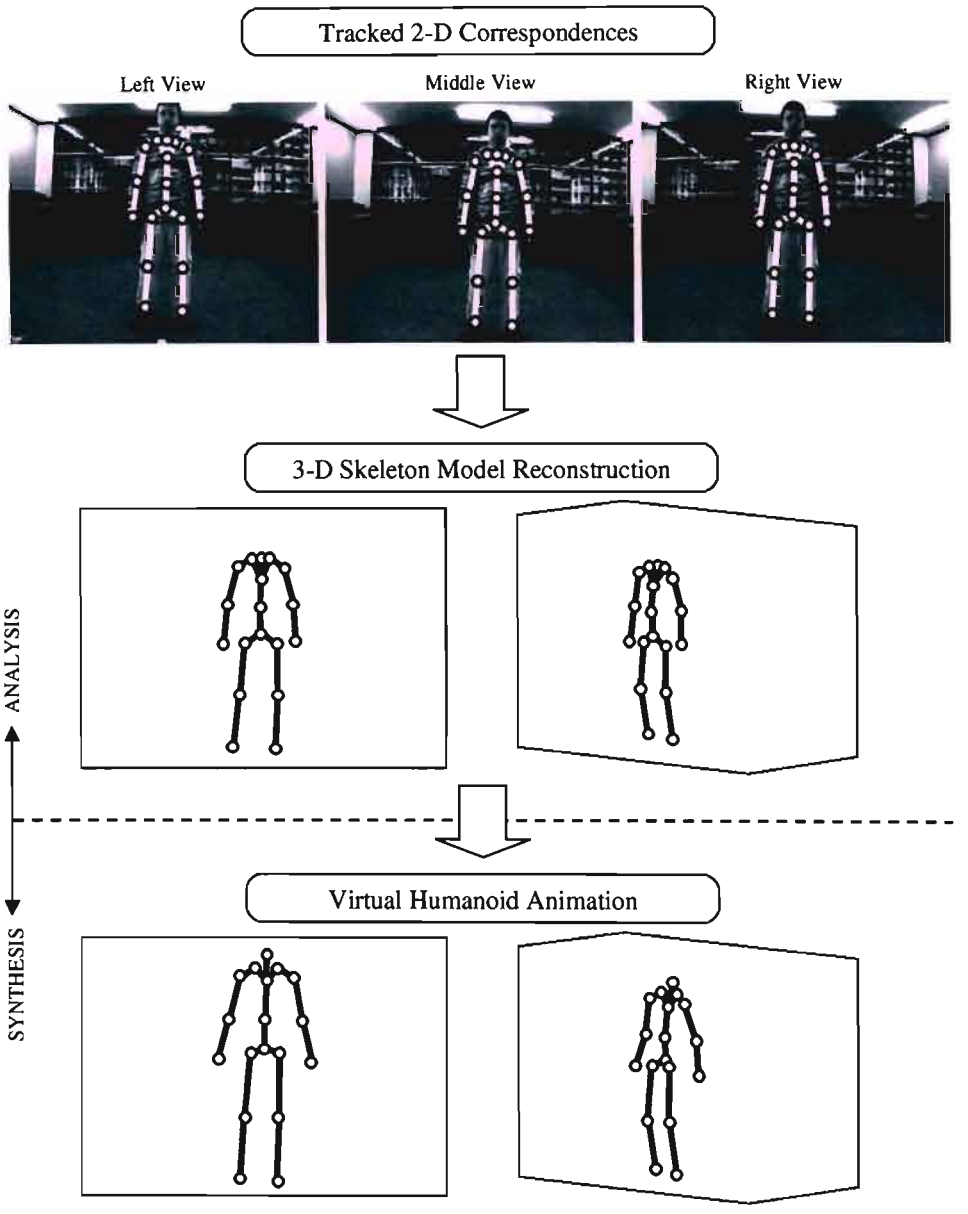


Figure 7.20: Frame 1 of the "walking" sequence.

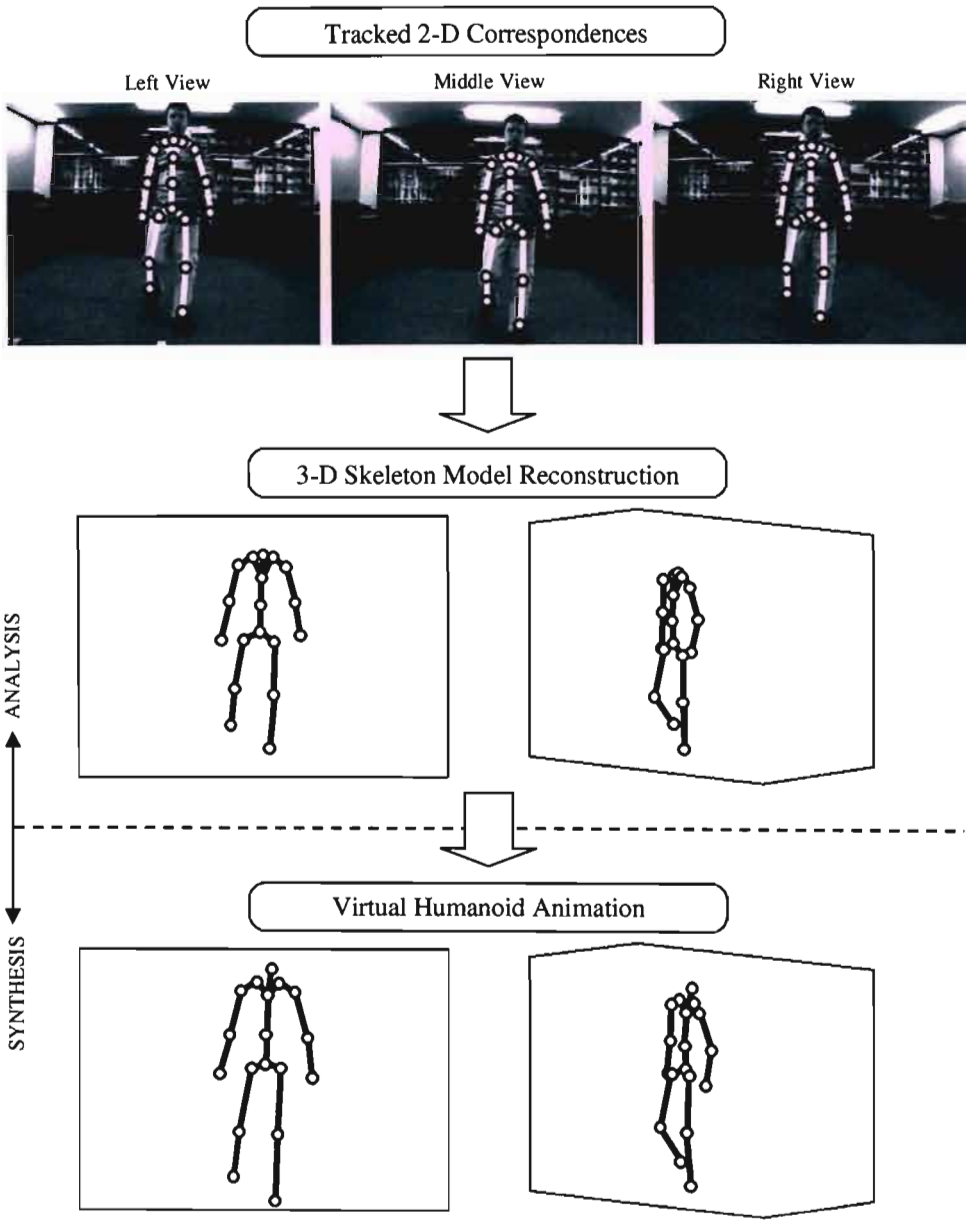


Figure 7.21: Frame 35 of the "walking" sequence.

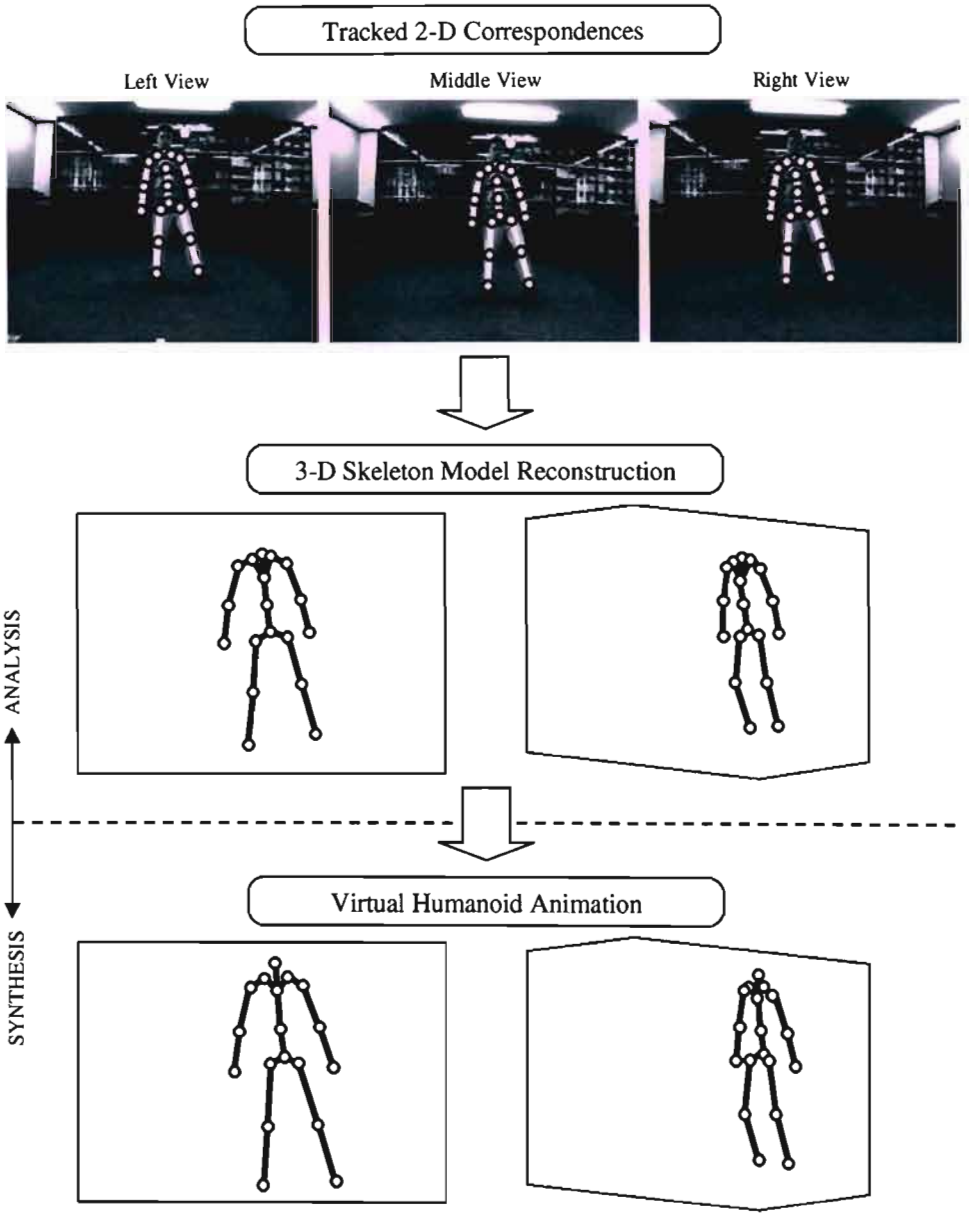


Figure 7.22: Frame 115 of the "walking" sequence.

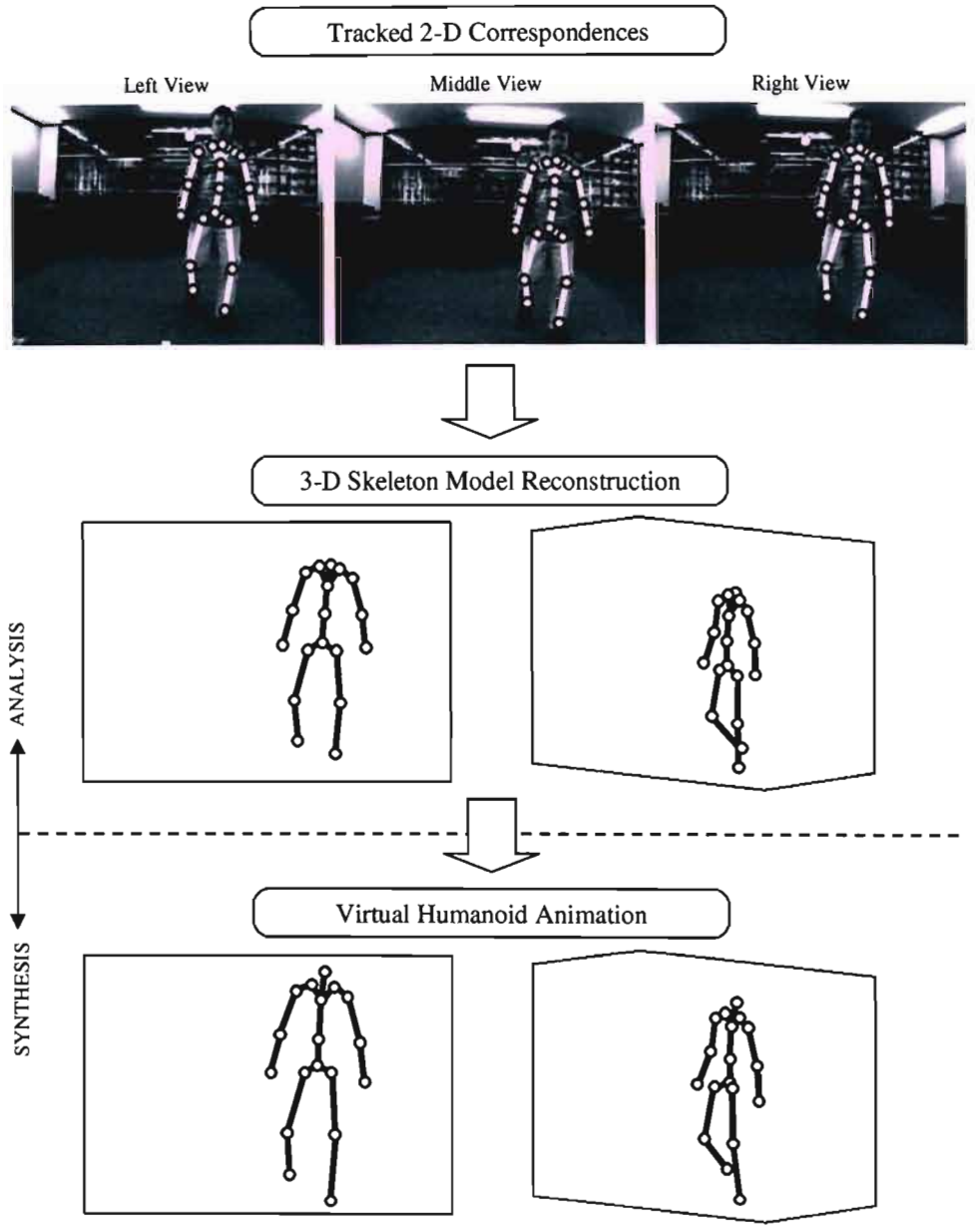


Figure 7.23: Frame 185 of the "walking" sequence.

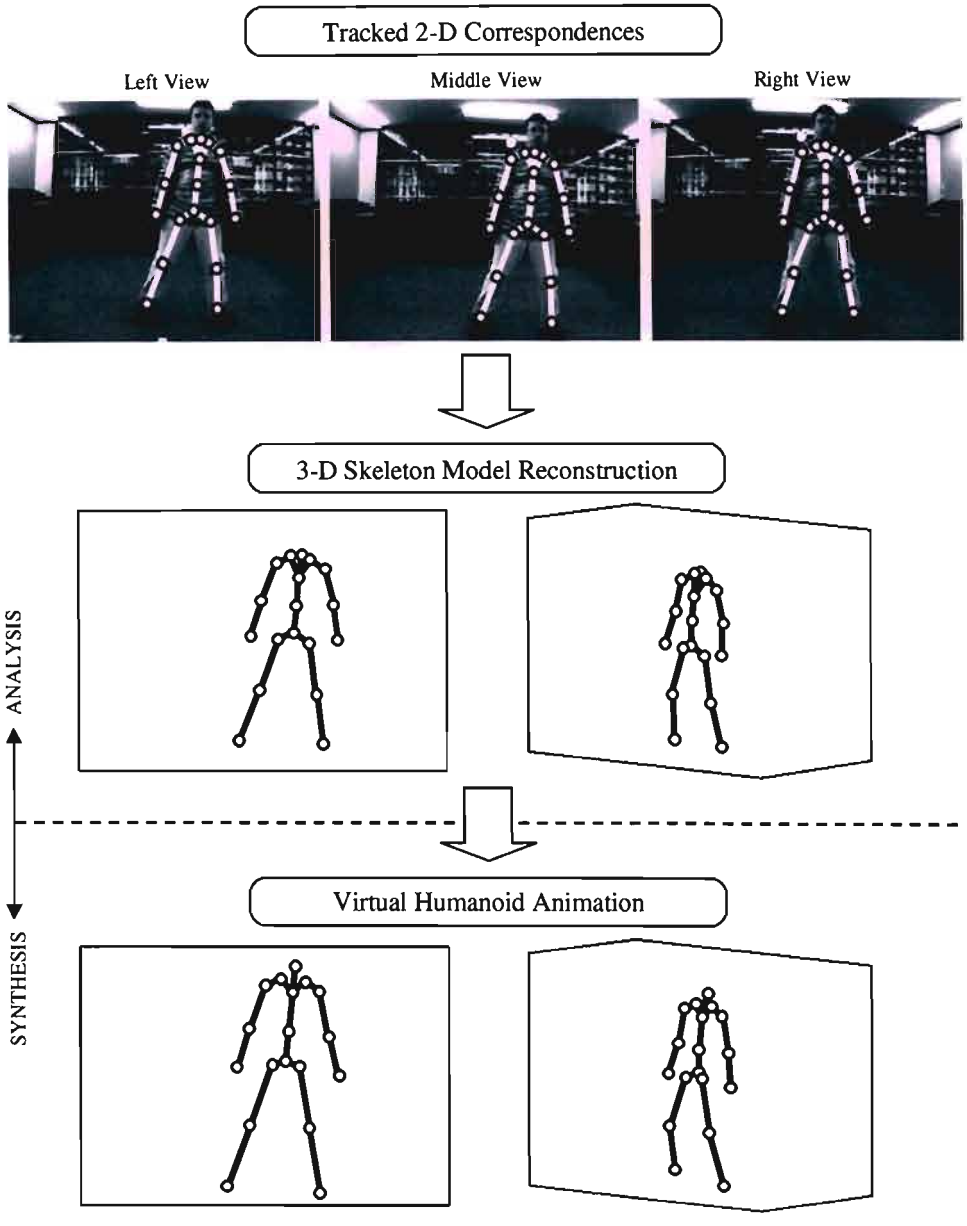


Figure 7.24: Frame 205 of the "walking" sequence.

7.7 Conclusion

In this section, the various components presented in this thesis were evaluated. Zhang's calibration method [112] [114] provides good accuracy with little effort, making the cal-

ibration step worthwhile. Parameters recovered through calibration greatly simplify the computation of epipolar geometry and the process of 3-D reconstruction. Two image matching techniques have been presented in this work; cross-correlation matching and least squares matching, with the former being described as a coarse matching technique and the latter as a sub-pixel matching technique. Section 7.2 investigated the performance of least squares matching, and compares it to the performance of cross-correlation. The experimental results demonstrated the ability of least squares matching to cope with various levels of distortion between images. Conversely, a coarse matching technique such as cross-correlation was able to perform only in situations with very small geometric distortion.

In Section 7.3, accurate matching was shown to be a prerequisite for reliable 3-D reconstruction, particularly if the points lie further away from the cameras. The millimetre accuracy of the reconstruction results is indicative of high accuracy of both the calibration and least squares matching. Consequently, the least squares matching based tracking algorithm was able to track points in 3-D space within a few millimetres, provided the tracking targets were planar and contained well defined edges. Application of the tracking algorithm to markerless motion capture is faced with adverse influences such as deformation of clothes indescribable by the affine transform, motion blur, image resolution loss caused by scale reduction and in general less defined targets in comparison to synthetic markers. All these factors affect the accuracy of the 3-D tracking and in some cases a 1.5 cm divergence was experienced. Errors of this magnitude are acceptable for visually accurate animation as they do not cause easily perceivable change in segment length.

To complete the implementation of the model-based coding scheme, the MPEG-4 pose parameters (BAPs) describing the tracked skeleton's pose must be recovered and applied to some default model. Although the 31 utilised parameters adequately describe the given pose, the reconstructed model exhibits noticeable differences. The discrepancy is attributed to the loss of volumetric information regarding the structure of the human body by tracking points on the subject's surface. To guarantee better similarity between the tracked and reconstructed model, either the default model of the model-based coding system must be redefined, or the 3-D motion data must be better interpreted. The former

option is not favoured as it affects compliance with other MPEG-4 based systems. The latter alternative could make use of the two rigid regions (pelvis and upper chest where the thoraic vertebrae 6 connects to the left and right sternoclaviculars) to offset the spine region in order to account for the volume of the torso.

Despite the imprecision of the reconstructed model it is still able to convey the motion information as the posture is mainly defined by the limbs that are not susceptible to the volumetric aspect to such a large degree. Section 7.6 presents a number of sample frames from the "walking" sequence together with the reconstructed pose which by visual inspection believably depicts the subject's posture.

The tracking process using least squares matching extended the work by D'Apuzzo [46] [138] by incorporating tests for the detection of undeterminable affine parameters (Section 5.1.1) and allowing the recovery of lost tracking in one view by spatial matching from another. D'Apuzzo's approach to human motion capture tracks the subject's surface in the multi-view video sequence. Presented results show human motion representation by 11 points defined at locations unable to describe full body pose. The accuracy of tracking compares well to other work even with the slight divergence caused by factors discussed in Section 7.4.3. Azarbayejani and Pentland [66] reported accuracy of their 3-D tracking to be within 2 cm. Although the system presented in [66] is very robust, it only tracks the hands and head making it unsuitable for full body pose estimation. Full body pose was estimated in [80] by 16 BAPs and by 22 BAPs in [88]. Poppe *et al.* [80] recovered the BAPs with an average error below 15 degrees. Huang and Lin [88] do not provide quantitative results however report difficulties with fast motions.

The target precision of markerless human motion capture systems is currently the performance of the commercial optical systems that use controlled environments and retroreflective markers to achieve accuracies exceeding 0.1mm [9]. Markerless systems must first achieve the same level of precision and robustness to occlusion provided by the use of artificial markers before being able to compete with today's optical solution to motion capture.

Chapter 8

Conclusions

8.1 Conclusions

The analysis of human motion, whether applied to reconstruction, recognition or tracking is a problem that always reduces down to human motion capture, i.e. the generation of necessary motion data. Since the late 1980's, the optical approach became a feasible solution to human motion capture and particularly in the past decade has received a dramatic increase in attention from researchers. Capturing human motion is not a trivial task because the human body is a complex articulated object whose motion demonstrates extensive self-occlusion. Despite the research effort the problem of tracking unconstrained, markerless human motion in everyday environments has not yet been solved. Researchers adopt approaches depending on the intended application, introducing constraints and assumptions in order to simplify the task. The commercially available state of the art optical tracking systems make use of expensive sophisticated hardware and use artificial markers placed on the subject along with controlled environments to accurately track performed motion.

The presented system for tracking human motion made use of multiple cameras. Multiple views considerably simplify the problem of depth estimation and allow the exploitation of epipolar geometry inherent in a stereo setup. Straightforward and accurate camera

calibration is possible with the Matlab Calibration Toolbox [111] and the effort is justified by the resulting simplification of epipolar geometry computation and 3-D reconstruction.

The tracking algorithm incorporates two image matching techniques to track points in multi-image space throughout the input video sequence. The first image matching technique introduced was cross-correlation. Although cross-correlation is unable to cope with large geometric differences between images, it was found useful as a similarity measure and for improving estimates by searching a reduced image space. Least squares matching describes the geometric difference between the conjugate images by an affine transform and illumination differences by additive and multiplicative coefficients of radiometric correction. Image content encapsulated by the matching patches strongly influences the quality of the result with the presence of strong gradients aiding the recovery of the affine parameters. In order for the affine transform to adequately describe the geometric relationship between the two patches, the matching patches must represent planar surfaces. Effects of noise are reduced by calculating the $\mathbf{A}$ matrix from the average of the template and search patch. Experimental results presented in Section 7.2 show the ability of least squares matching to recover correspondences with high accuracy even in cases of large image differences. The sub-pixel accuracy of least squares matching is vital for both 3-D reconstruction and successful tracking as demonstrated in Section 7.3 and Section 7.4.

The accuracy of the tracking algorithm is defined by the accuracy of the matching and 3-D reconstruction processes. Test sequences with artificial matching targets demonstrated tracking accuracy within a few millimetres. Tests on real data revealed the significance of hardware in human motion capture. Motion blur impairs the image gradients necessary for accurate determination of the affine parameters, consequently affecting the accuracy of the tracked points or causing tracking failure altogether. Image resolution was found to affect tracking when motion path varied in depth. The loss of resolution of the tracked feature causes a drift from the original location. Nonetheless tracking results for full body motion with a depth displacement of up to 2.8m yielded accurate motion data in reference to the perceivable error by the human observer. It is postulated that more sophisticated hardware would result in more accurate tracking results without any change to the existing algorithm. Commercial systems overcome many of the discussed problems by using high

resolution cameras, high speed cameras that decrease the difference in successive frames (typical cameras with 30 to 60 fps are considered insufficient [53]) and retroreflective markers that appear very bright in an image when illuminated by IR light. The imaging hardware as well as the capture environments make optical motion capture considerably more expensive than other methods.

The process of motion reconstruction in this thesis was presented in the context of a model-based coding scheme. Model-based coding regards an image as a 2-D projection of 3-D objects. With default 3-D models present at both the encoder and decoder only the parameters describing an object's inter-frame change are transmitted. The tracked 3-D skeleton was parameterized by a selection of 31 MPEG-4 animation parameters. Compliance with the MPEG-4 standard ensures that the recovered data may be used to animate a variety of humanoid models.

In conclusion, the presented markerless human motion capture system once manually initialised was able to track points in 3-D space. Tracking success was strongly dependant on the sub-pixel accuracy of least squares matching. Inadequate hardware in human motion capture terms at times degraded image data and introduced inaccuracy into the tracking process. Resulting 3-D motion data however was still good enough for visual assessment. 31 MPEG-4 animation parameters (BAPs) were extracted from the skeleton model which fits into the tracked data. Motion reconstruction was achieved by applying the BAPs to a default model, demonstrating the application of model-based coding to low bit-rate video.

8.2 Future Work

There are a number of avenues for future research considering the final scenario of a fully automatic system, whose only input is the video sequence to the encoder and whose output is synthesised by a photorealistic model, scaled and texture mapped to resemble the subject(s) from the input sequence. This would require additions both in the computer vision and the computer graphics parts of the problem. Throughout the thesis, the focus

has been on the computer vision aspect of the system and the same applies to the suggested directions of future work. The main areas of improvement within the human motion capture system are the automation and the robustness of the tracking process.

At this stage, the user is required to calibrate the cameras as well as to manually select the key points in the first frame of one of the views. Auto-calibration is possible by Kruppa's equation [116] which links the fundamental matrix to the image of the absolute conic. The fundamental matrix itself can be calculated from 7 or more correspondences [118] eliminating the need for a special calibration setup. The calibration process may also be greatly simplified by using special hardware such as the Digiclops camera from Point Grey Research [45], specifically designed for 3-D computer vision applications. Digiclops consists of three progressive scan CCD cameras housed in one package, calibrated for lens distortion and internal alignment.

The elimination of the manual initialisation step is perhaps the greatest contribution towards fully automating the system. To accurately estimate the location of the desired joints, a 3-D visual hull representation of the subject as used in [18] would provide valuable information. Anthropometric information was exploited in [95] for pose estimation in single monocular images and could no doubt be exploited in a process of fitting a 3-D skeleton into the visual hull.

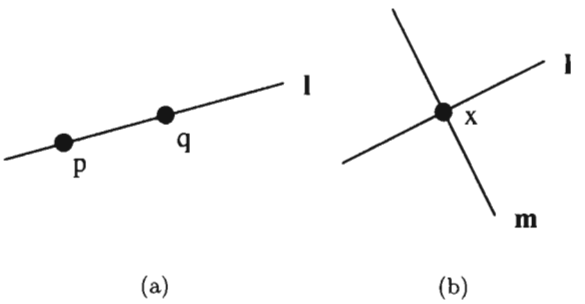
The main and most challenging aspect of human motion tracking is handling occlusion. Past research has handled occlusion by actively selecting the view in which a body part is not occluded [102], by reincorporating the occluded feature into the tracking process once it reappears [11] or by employing more sophisticated human models that can predict occlusion and collision more reliably [17]. In commercial systems, occlusion is handled by employing a large number of cameras to ensure that tracked features are visible in at least one view. Using greater number of cameras is an attractive solution. Distributing the cameras around the workplace would allow for most of the subject to be visible at all times, as well as for a more accurate visual hull representation. By tracking several points distributed around a given joint, the joint's actual location within the body could be estimated and thus the volumetric structure of the body would be recovered.

To achieve realistic modelling of the subjects from the video sequences, their BDPs in addition to the BAPs would also have to be extracted and transmitted. The BDPs are used by the decoder to customise the default model in order to resemble the subject, which entails scaling the default segment lengths, defining the size of modelled body parts and even texture mapping the model.

Appendix A

Homogenous (Projective) Representation of Lines

- A line $ax + by + c = 0$ is represented by the homogenous vector $\mathbf{l}$ where: $\mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$
- Any vector $k \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ represents the same line
- Only the ratio of the homogenous line coordinates is significant. A line can be specified by just two parameters, $y = mx + c$, so $ax + by + c = 0$ can be rewritten as: $y = -\frac{a}{b}x - \frac{c}{b}$
- Homogenisation rule for lines, $\begin{bmatrix} a \\ b \\ c \end{bmatrix} : \begin{bmatrix} -a/b \\ -c/b \end{bmatrix}$
- Properties involving lines and points
  - the point x lies on the line $\mathbf{l}$ *iff*: $x^T\mathbf{l} = 0$
  - two points define a line: $\mathbf{l} = p \times q$ (Figure A.1(a))
  - two lines define a point: $x = \mathbf{l} \times \mathbf{m}$ (Figure A.1(b))



(a) (b)

Figure A.1: (a) Two points define a line, (b) two lines define a point.

Appendix B

Rodrigues' Rotation Formula

This formulation of the Rodrigues' rotation formula directly follows that of [159]. It is an efficient method of computing the rotation matrix $\mathbf{R} \in SO(3)$ corresponding to a rotation by an angle $\theta \in \mathbb{R}$ about a fixed axis specified by the unit vector $\hat{\omega} = (\omega_x, \omega_y, \omega_z) \in \mathbb{R}^3$. $\mathbf{R}$ is then given by:

$$e^{\hat{\omega}\theta} = I + \hat{\omega}\sin\theta + \hat{\omega}^2(1 - \cos\theta)$$

$$= \begin{bmatrix} \cos\theta + \omega_x^2(1-\cos\theta) & \omega_x\omega_y(1-\cos\theta) - \omega_z\sin\theta & \omega_y\sin\theta + \omega_x\omega_z(1-\cos\theta) \\ \omega_z\sin\theta + \omega_x\omega_y(1-\cos\theta) & \cos\theta + \omega_y^2(1-\cos\theta) & -\omega_x\sin\theta + \omega_y\omega_z(1-\cos\theta) \\ -\omega_y\sin\theta + \omega_x\omega_z(1-\cos\theta) & \omega_x\sin\theta + \omega_y\omega_z(1-\cos\theta) & \cos\theta + \omega_z^2(1-\cos\theta) \end{bmatrix}$$

where $\mathbf{I}$ is the 3×3 identity matrix and $\hat{\omega}$ denotes the antisymmetric matrix:

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

Appendix C

Definition of Skeleton Model Segments

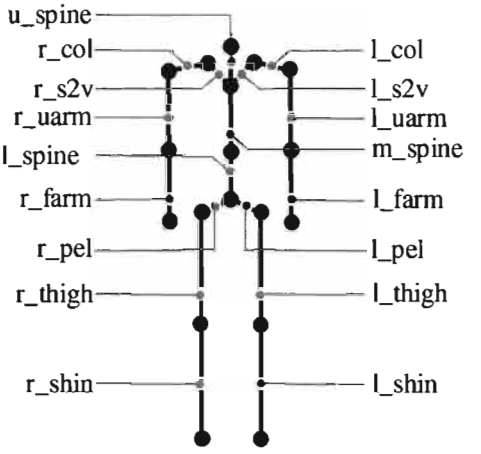


Figure C.1: Definition of segment names.

Bibliography

[1] D. Thalmann and F. Vexo, "MPEG-4 character animation.," *Kunstliche Intelligenz*, vol. 17, no. 4, pp. 39–, 2003.

[2] Wikipedia, the Free Encyclopedia, http://en.wikipedia.org.

[3] Australian Centre for the Moving Image, http://www.acmi.net.au.

[4] Polhemus, http://www.polhemus.com.

[5] Ascension Technology Corporation, http://www.ascension-tech.com.

[6] Intersense, http://www.intersense.com.

[7] Animazoo, http://www.animazoo.com.

[8] Leeds Metropolitan University, Faculty of Information and Technology, http://www.leedsmet.ac.uk.

[9] Vicon, http://www.vicon.com.

[10] A. Blake, R. Curwen, and A. Zisserman, "A framework for spatiotemporal control in the tracking of visual contours," *International Journal of Computer Vision*, vol. 11, pp. 127–145, October 1993.

[11] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780–785, July 1997.

[12] M. Leung and Y. Yang, "First sight: A human-body outline labeling system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 359–377, April 1995.

[13] S. Ju, M. Black, and Y. Yacoob, "Cardboard people: A parameterized model of articulated image motion," in *International Conference on Automatic Face and Gesture Recognition*, (Killington, VT), 1996.

[14] R. Holt, T. Huang, A. Netravali, and R. Qian, "Determining articulated motion from perspective views: A decomposition approach," in *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 126–137, 1994.

[15] C. Wren and A. Pentland, "Dynaman: Recursive modeling of human motion," Technical Report 451, MIT Media Lab, 1997.

[16] D. Hogg, "Model-based vision: A program to see a walking person," *Image and Vision Computing*, vol. 1, pp. 5–20, February 1983.

[17] D. Gavrila and L. Davis, "3-D model-based tracking of humans in action: A multi-view approach," in *Conference on Computer Vision and Pattern Recognition, San Francisco*, 1996.

[18] C. Theobalt, M. Magnor, P. Schueler, , and H. Seidel, "Combining 2-D feature tracking and volume reconstruction for online video-based human motion capture," in *Proceedings of Pacific Graphics*, (Beijing), pp. 96–103, 2002.

[19] R. Plankers, N. D'Apuzzo, and P. Fua, "Automated body modeling from video sequences," in *IEEE International Workshop on Modeling People*, (Corfu, Greece), p. 45, 1999.

[20] J. McIntosh, "Implementation of an application specific low bit rate video compression scheme," Master's thesis, University of Natal, February 2002.

[21] E. Jackson, "High ratio wavelet video compression through real time rate distortion estimation," Master's thesis, University of Natal, Durban, South Africa, July 2003.

[22] T. Murugas, "Video object tracking," Master's thesis, University of Natal, Durban, South Africa, 2004.

[23] K. Aizawa and T. Huang, "Model based image coding: Advanced video coing techniques for very low bit-rate applications," *Proceedings of the IEEE*, vol. 83, pp. 259–271, February 1995.

[24] D. Pearson, "Model-based image coding," in *Proceedings of GLOBECOM*, vol. 16, (Dallas, Texas), pp. 554–558, November 1989.

[25] R. Forchheimer, O. Fahlander, and T. Kronander, "Low bit-rate coding through animation," in *Proceedings of the International Picture Coding Symposium (PCS)*, (Davis, CA, USA), pp. 113–114, 1983.

[26] M. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," *Inernational Journal of Computer Vision*, vol. 25, 1997.

[27] C. Choi, K. Aizawa, H. Harashima, and T. Takebe, "Analysis and synthesis of facial image sequences in model-based image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, pp. 257–275, June 1994.

[28] T.Cootes, G. Edwards, and C.J.Taylor, "Active appearance models," in *European Conference on Computer Vision 1998* (H.Burkhardt and B. Neumann, eds.), pp. 484–498, Springer, 1998.

[29] P. Fitzpatrick, "Head pose estimation without manual initialization," term paper for course 6.892, MIT, Cambridge, MA, 2001.

[30] C. Nastar, B. Moghaddam, and A. Pentland, "Generalized image matching: Statistical learning of physically-based deformations," in *European Conference on Computer Vision*, pp. I:589–598, 1996.

[31] J. Strom, "Model-based real-time head tracking," *Journal of Applied Signal Processing*, vol. 10, pp. 1039–1052, October 2002.

[32] J. Strom, T.Jebara, and A. Pentland, "Model-based real-time face tracking with adaptive texture update," Technical Report LiTH-ISY-R-2342, Linkoping University, Sweden, March 2001.

[33] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–96, 1991.

[34] F. P. M. Preda, T. Zaharia, "3D body animation and coding within a MPEG-4 compliant framework," in *International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging*, (Fira, Santorini, Greece), pp. 74–78, September 1999.

[35] T. Capin, E. Petajan, and J. Ostermann, "Efficient modeling of virtual humans in MPEG-4," in *IEEE International Conference on Multimedia and Expo*, pp. 1103–1106, July 2000.

[36] T. Capin, E. Petajan, and J. Ostermann, "Very low bitrate coding of virtual human animation in MPEG-4," pp. 1107–1110, July 2000.

[37] P. Reitsma and N. Pollard, "Perceptual metrics for character animation: sensitivity to errors in ballistic motion," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 537–542, 2003.

[38] J. Harrison, R. Rensink, and M. van de Panne, "Obscuring length changes during animated motion," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 569–573, 2004.

[39] Specification for a Standard VRML Humanoid, (DRAFT) Version 1.1, www.h-anim.org.

[40] T. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Journal of Applied Signal Processing*, vol. 81, no. 3, pp. 231–268, 2001.

[41] D. Thalmann, J. Shen, and E. Chauvineau, "Fast realistic human body deformations for animation and VR applications," in *Proceedings of Computer Graphics International IEEE Computer Society Press*, pp. 166–174, June 1996.

[42] ISO/IEC JTC1/SC29, ISO/IEC 14496-2:2004, Information technology - Coding of audio-visual objects - Part 2: Visual.

[43] M. Shah and R. Jain, *Motion-Based Recognition.* Kluwer Academic Publishers, 1997.

[44] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 126–133, 2000.

[45] Point Grey Research, http://www.ptgrey.com.

[46] N. D'Apuzzo, "Surface measurement and tracking of human body parts from multi-image video sequences," *Journal of Photogrammetry and Remote Sensing*, vol. 56, pp. 360–375, August 2002.

[47] E. Muybridge, *Muybridge's Complete Human and Animal Locomotion*, vol. 3. Dover Publications, July 1979.

[48] G. Johansson, "Visual motion perception," *Scientific American*, pp. 76–88, November 1976.

[49] D. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[50] H. Feys, W. D. Weedt, B. Selz, G. Steck, R. Spichiger, L. Vereek, K. Putman, and G. V. Hoydonck, "Effect of therapeutic intervention for the hemiplegic arm in the acute phase after stroke: a single blinded, randomised controlled multi-centre trial," *Stroke*, vol. 29, no. 4, pp. 785–792, 1998.

[51] J. Eriksson and M. Mataric, "Hands-off robotics for post-stroke arm rehabilitation," Technical Report CRES-04-011, USC Center for Robotics and Embedded Systems, October 2004.

[52] M. Meredith and S. Maddock, "Motion capture file formats explained," technical report, University of Sheffield, 2000.

[53] A. Mulder, "Human movement tracking technology," Technical Report 94-1, School of Kinesiology, Simon Fraser University, July 1994.

[54] A. Mulder, "Human movement tracking technology: Resources," Addendum to Technical Report 94-1, School of Kinesiology, Simon Fraser University, July 1994.

[55] D. Bhatnagar, "Position trackers for head mounted display systems: A survey," Technical Report TR93-010, University of North Carolina, Chapel Hill, NC, USA, 1993.

[56] C. Cedras and M. Shah, "Motion-based recognition: A survey," *Image Vision Computing*, vol. 13, no. 2, pp. 129–155, 1995.

[57] J. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, 1999.

[58] H. Zhou and H. Hu, "A survey: human movement tracking and stroke rehabilitation," Department Report CSM-420, University of Essex, December 2004.

[59] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models: Their training and application," *Computer Vision and Image Understanding*, vol. 61, pp. 38–59, January 1995.

[60] A. Baumberg and D. Hogg, "An efficient method for contour tracking using active shape models," in *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 194–199, 1994.

[61] W. Freeman, K. Tanaka, J. Ohta, and K. Kyuma, "Computer vision for computer games," in *Proceddings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 100–105, 1996.

[62] F. Quek, "Eyes in the interface," *Image and Vision Computing*, vol. 13, pp. 511–525, August 1995.

[63] T. Darrell and A. Pentland, "Space-time gestures," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 335–340, 1993.

[64] R. Polana and R. Nelson, "Low level recognition of human motion," in *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 77–82, 1994.

[65] R. Fablet and M. Black, "Automatic detection and tracking of human motion with a view-based representation," in *European Conference on Computer Vision*, vol. 1, pp. 476–491, 2002.

[66] A. Azarbayejani and A. Pentland, "Real-time self-calibrating stereo person tracking using 3D shape estimation from blob features," in *International Conference on Pattern Recognition*, pp. III: 627–632, 1996.

[67] C. Wren and A. Pentland, "Dynamic models of human motion," in *International Conference on Automatic Face and Gesture Recognition*, (Nara, Japan), pp. 22–27, 1998.

[68] C. Wren, B. Clarkson, and A. Pentland, "Understanding purposeful human motion," in *IEEE International Workshop on Modelling People*, (Corfu, Greece), pp. 19–25, September 1999.

[69] K. Akita, "Image sequence analysis of real world human motion," *Pattern Recognition*, vol. 17, pp. 73–83, 1984.

[70] M. Leung and Y. Yang, "Human body motion segmentation in a complex scene," *Pattern Recognition*, vol. 20, no. 1, pp. 55–64, 1987.

[71] W. Long and Y. Yang, "Log-tracker: An attribute-based approach to tracking human body motion," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 5, pp. 439–458, 1991.

[72] S. Kurakake and R. Nevatia, "Description and tracking of moving articulated objects," in *International Conference on Pattern Recognition*, pp. 491–495, 1992.

[73] I. Chang and C. Huang, "Ribbon-based motion analysis of human body movements," in *International Conference on Pattern Recognition*, (Vienna), pp. 436–440, 1996.

[74] M. Black and Y. Yacoob, "Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion," in *International Conference on Computer Vision*, pp. 12–17, 1995.

[75] Q. Cai and J. Aggarwal, "Tracking human motion using multiple cameras," in *International Conference on Pattern Recognition*, (Vienna), pp. 68–72, 1996.

[76] Q. Cai, A. Mitiche, and J. Aggarwal, "Tracking human motion in an indoor environment," in *International Conference on Image Processing*, (Washington D.C.), pp. 215–218, October 1995.

[77] H. Lee and Z. Chen, "Determination of 3D human body postures from a single view," *Computer Vision, Graphics, and Image Processing*, vol. 30, pp. 148–168, May 1985.

[78] Z. Chen and H. Lee, "Knowledge-guided visual perception of 3D human gait from a single image sequence," *Systems, Man, and Cybernetics*, vol. 22, pp. 336–342, 1992.

[79] J. Zhao, L. Li, and K. Keong, "A model-based approach for human motion reconstruction from monocular images," in *Proceedings of the International Conference on Information Technology for Application*, (Harbin, China), pp. 94–99, 2004.

[80] R. Poppe, D. Heylen, A. Nijholt, and M. Poel, "Towards real-time body pose estimation for presenters in meeting environments," in *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, (Plzen, Czech Republic), pp. 41–44, 2005.

[81] C. Barron and I. Kakadiaris, "Monocular human motion tracking," *Multimedia Systems*, vol. 10, no. 2, pp. 118–130, 2004.

[82] J. Rehg and T. Kanade, "Visual tracking of high DOF articulated structures: An application to human hand tracking," in *European Conference on Computer Vision*, (Stockholm), pp. 35–46, 1994.

[83] J. Rehg and T. Kanade, "Model-based tracking of self-occluding articulated objects," in *International Conference on Computer Vision5*, pp. 612–617, 1995.

[84] K. Rohr, "Towards model-based recognition of human movements in image sequences," *Computer Vision Graphics and Image Processing*, vol. 59, pp. 94–115, January 1994.

[85] T. B. Moeslund and E. Granum, "3-D human pose estimation using 2-D data and an alternative phase space representation," in *Workshop on Human Modeling, Analysis and Synthesis at CVPR*, (Hilton Head Island, SC), pp. 26–33, June 2000.

[86] T. Moeslund and E. Granum, "Multiple cues for model-based human motion capture," in *International Conference on Automatic Face and Gesture Recognition*, (Grenoble, France), pp. 362–367, March 2000.

[87] K. Rohr, "Incremental recognition of pedestrians from image sequences," in *Conference on Computer Vision and Pattern Recognition*, (New York), pp. 8–13, June 1993.

[88] C.-L. Huang and C.-C. Lin, "Model-based human body motion analysis for MPEG IV video encoder," in *Proceedings of the International Conference on Information Technology: Coding and Computing*, pp. 435–439, 2001.

[89] J. O'Rourke and N. Badler, "Model-based image analysis of human motion using constraint propagation," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 2, pp. 522–536, November 1980.

[90] P. Fua, A. Gruen, R. Plankers, N. D'Apuzzo, and D. Thalmann, "Human body modeling and motion analysis from video sequences," in *International Symposium on Real Time Imaging and Dynamic Analysis*, (Hakodate, Japan), pp. 866–873, June 1998.

[91] P. Fua, R. Plankers, and D. Thalmann, "From synthesis to analysis: Fitting human animation models to image data," in *Proceedings of the International Conference on Computer Graphics Interface*, (Washington, DC, USA), p. 4, IEEE Computer Society, 1999.

[92] D. Gavrila and L. Davis, "3-D model-based tracking of human upper body movement: A multi-view approach," in *International Symposium on Computer Vision*, (Coral Gables, U.S.A.), pp. 253–258, 1995.

[93] D. Gavrila and L. Davis, "Towards 3-D model-based tracking and recognition of human movement: A multi-view approach," in *International Workshop on Gesture and Face Recognition*, (Zurich), pp. 272–277, IEEE Computer Society, 1995.

[94] A. Pentland, "Automatic extraction of deformable part models," *International Journal of Computer Vision*, vol. 4, pp. 107–126, March 1990.

[95] C. Barron and I. Kakadiaris, "On the improvement of anthropometry and pose estimation from a single uncalibrated image," *Machine Vision and Applications*, vol. 14, no. 4, pp. 229–236, 2003.

[96] The American Heritage, *Stedman's Medical Dictionary.* Houghton Mifflin Company, 2002.

[97] D. Marr and H. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proceedings of the Royal Society of London*, vol. B-200, pp. 269–294, 1978.

[98] L. Goncalves, E. di Bernardo, E. Ursella, and P. Perona, "Monocular tracking of the human arm in 3-D," in *International Conference on Computer Vision*, pp. 764–770, 1995.

[99] Q. Delamarre and O. Faugeras, "3D articulated models and multi-view tracking with silhouettes," in *International Conference on Computer Vision*, (Corfu, Greece), pp. 716–721, September 1999.

[100] Q. Delamarre and O. Faugeras, "3D articulated models and multiview tracking with physical forces," *Computer Vision and Image Understanding*, vol. 81, pp. 328–357, March 2001.

[101] I. Kakadiaris and D. Metaxas, "3-D human body model acquisition from multiple views," in *International Conference on Computer Vision*, (Cambridge), pp. 618–623, 1995.

[102] I. Kakadiaris and D. Metaxas, "Model-based estimation of 3-D human motion with occlusion based on active multi-viewpoint selection," in *Conference on Computer Vision and Pattern Recognition*, (San Francisco), pp. 81–87, 1996.

[103] I. Kakadiaris and D. Metaxas, "Vision based animation of digital humans," in *Computer Animation*, pp. 144–152, IEEE Computer Society, June 1998.

[104] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision.* Upper Saddle River, NJ: Prentice Hall, 1998.

[105] E. Ong and S. Gong, "Tracking hybrid 2D-3D human models from multiple views," in *International Workshop on Modeling People*, (Corfu, Greece), p. 11, September 1999.

[106] N. Grammalidis, G. Goussis, G. Troufakos, and M. Strintzis, "Estimating body animation parameters from depth images using analysis by synthesis," in *Second International Workshop on Digital and Computational Video*, pp. 93–100, 2001.

[107] V. Paramesvwaran, *View invariance in visual human motion analysis.* PhD thesis, University of Marylan, 2004.

[108] T. Shakunaga, "Pose estimation of jointed structures," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 566–572, 1991.

[109] Q. Cai and J. Aggarwal, "Tracking human motion in structured environments using a distributed-camera system," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 12, pp. 1241–1247, 1999.

[110] B. Dorner, "Hand shape identification and tracking for sign language interpretation," in *International Joint Conference on Artificial Intelligence Workshop on Looking at People*, (Chambery), 1993.

[111] J.-Y. Bouguet, "Camera Calibration Toolbox for Matlab," (http://www.vision.caltech.edu/bouguetj/calib_doc), April 2002.

[112] Z. Zhang, "A flexible new technique for camera calibration," Technical Report MSRTR-98-71, Microsoft Research, December 1998.

[113] Intel OpenCV Computer Vision Library (C++), http://www.intel.com/research/mrl/research/opencv/.

[114] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *International Conference on Computer Vision*, (Corfu, Greece), pp. 666–673, September 1999.

[115] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1106–1112, 1997.

[116] O. Faugeras, Q. Luong, and S. Maybank, "Camera self-calibration: Theory and experiments," in *European Conference on Computer Vision*, pp. 321–334, 1992.

[117] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, September 1981.

[118] Q. Luong and O. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *The International Journal of Computer Vision*, vol. 17, pp. 43–75, January 1996.

[119] R. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 580–593, June 1997.

[120] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *The International Journal of Computer Vision*, vol. 27, pp. 161–195, March 1998.

[121] G. Csurka, C. Zeller, Z. Zhang, and O. Faugeras, "Characterizing the uncertainty of the fundamental matrix," *Computer Vision and Image Understanding*, vol. 68, pp. 18–36, October 1997.

[122] E. Gulch, *Erzeugung digitaler Gelandemodelle durch automatische Bildzuordnung.* PhD thesis, Institute of Photogrammetry, University of Stuttgart, 1994.

[123] P. Burt, C. Yen, and X. Xu, "Local correlation measures for motion analysis: A comparative study," Image Processing Laboratory Technical Report IPL-TR-024, 1982.

[124] M. Lemmens, "A survey on stereo matching techniques," in *International Archives of Photogrammetry and Remote Sensing*, vol. 27, (Kyoto, Japan), pp. 11–23, 1988.

[125] W. Webber, "Techniques for image registration," in *Proceedings of IEEE Conference on Machine Processing of Remotely Sensed Data*, pp. 1–7, 1973.

[126] F. Ackerman, "High precision digital image correlation," in *Proceedingsof the 39th Photogrammetric Week, Institut fur Photogrammetrie, Universitat Stuttgart*, (Stuttgart, Germany), pp. 231–243, 1983.

[127] A. Gruen, "Adaptive least squares correlation: A powerful image matching technique," in *South African Journal of Photogrammetry, Remote Sensing and Cartography*, vol. 14, pp. 175–187, June 1985.

[128] A. Gruen, P. Agouris, and H. Li, "Linear feature extraction with dynamic programming and globally enforced least squares matching," *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pp. 83–94, January 2002.

[129] A. Gruen and E. Baltsavias, "Adaptive least squares correlation with geometrical constraints," in *Proceedings of Computer Vision for Robots*, vol. 595, (Cannes, France), pp. 72 – 82, Society of Photo-Optical Instrumentation Engineers, December 1985.

[130] A. Gruen and E. Baltsavias, "High precision image matching for digital terrain model generation," *Photogrammetria*, vol. 42, no. 3, pp. 97–112, 1987.

[131] A. Gruen and D. Stallmann, "High-accuracy matching of object edges," in *Videometrics* (El-Hakim and F. Sabry, eds.), vol. 1820, pp. 70–82, Society of Photo-Optical Instrumentation Engineers, February 1993.

[132] H. Maas, A. Stefanidis, and A. Gruen, "From pixels to voxels: Tracking volume elements in sequences of 3-D digital images," *International Archives of Photogrammetry and Remote Sensing*, vol. 30, pp. 539–546, September 1994.

[133] S. Zheltov and A. Sibiryakov, "Adaptive subpixel cross-correlation in a point correspondence problem," in *Optical-3D Measurement Techniques*, (Zurich), 1997.

[134] E. Baltsavias, *Multiphoto Geometrically Constrained Matching*. PhD thesis, Institute of Geodesy and Photogrammetry, Zurich, 1991.

[135] Y. Xin, "Automating procedures on a photogrammetric softcopy system," Master's thesis, University of Calgary, September 1995.

[136] D. Rosenholm, "Empirical investigation of optimal window size using least squares image matching method," *Photogrammetria*, vol. 42, pp. 113–125, 1987.

[137] P. Thevenaz, T. Blu, and M. Unser, "Image interpolation and resampling," in *Handbook of Medical Imaging, Processing and Analysis* (I. Bankman, ed.), (San Diego CA, USA), pp. 393–420, Academic Press, 2000.

[138] N. D'Apuzzo, *Surface Measurement and Tracking of Human Body Parts from Multi Station Video Sequences.* PhD thesis, Institute of Geodesy and Photogrammetry, ETH Zurich, Switzerland, November 2003.

[139] W. Forstner, "Matching strategies for point transfer," in *Photogrammetric Week 1995* (H. Fritsch, ed.), (Heidelberg), pp. 173–183, Herbert Wichmann Verlag, 1995.

[140] A. Pertl, "Empirical results of automatic parallax measurement," in *Proceedingsof the 40th Photogrammetry Week, Institute of Photogrammetry, Stuttgart University*, no. 11, (Stuttgart), pp. 109–125, 1986.

[141] R. Koenen, F. Pereira, and L. Chiariglione, "MPEG-4: context and objectives," *Image Communication Journal*, vol. 9, pp. 295–304, May 1997.

[142] R. Koenen, "MPEG-4, or why efficiency is much more than just a compression ratio," in *IBC*, (Amsterdam), September 2002.

[143] ISO/IEC JTC1/SC29/WG11, ISO/IEC 14496:1999, "Coding of audio, picture, multimedia and hypermedia information, n3056," December 1999.

[144] M. B. S. et al., "PDAM of ISO/IEC 14496-1 / AMD4, ISO/IEC JTC1/SC29/WG11, N4415," December 2001.

[145] M. Preda and F. Preteux, "Advanced animation framework for virtual character within the MPEG-4 standard," in *Proceedings of the IEEE International Conference on Image Processing*, (Rochester, NY), pp. 509–512, September 2002.

[146] "'SNHC Verification Model 9.0' ISO/IEC JTC1/SC29/WG11 W2301," July 1998.

[147] M. Preda and F. Preteux, "Advanced virtual humanoid animation framework based on the MPEG-4 SNHC standard," in *Proceedings EUROIMAGE International Conference on Augmented, Virtual Enviroments and Three-Dimensional Imaging*, (Mykonos, Greece), pp. 311–314, May 2001.

[148] F. Preteux, M. Preda, and T. Zaharia, "Results of core experiment on BAP coding, ISO/IEC JTC1/SC29/WG11, M4283," December 1998.

[149] F. Preteux, M. Preda, and T. Zaharia, "Predictive versus DCT-based BAP coding, ISO/IEC JTC1/SC29/WG11, M4254," March 1999.

[150] F. Preteux, M. Preda, and N. Rougon, "Advanced MPEG-4 animation system, milestones and deliverables," Report D4-4, June 2002.

[151] A. Sappa, N. Aifanti, N. Grammalidis, and S. Malassiotis, *3D Modeling and Animation: Synthesis and Analysis Techniques*, ch. Advances in Vision-Based Human Body Modeling, pp. 1–26. Idea Group Publishing, January 2004.

[152] ISO/IEC JTC1/SC29, ISO/IEC 14496-1:2004, Information technology - Coding of audio-visual objects - Part 1: Systems.

[153] S. Bhatia, L. Sigal, M. Isard, and M. J. Black, "3-D human limb detection using space carving and multi-view eigen models," in *Conference on Computer Vision and Pattern Recognition Workshop*, vol. 1, (Washington D.C., USA), p. 17, 2004.

[154] Y. Tao and H. Hu, "Building a visual tracking system for home-based rehabilitation," in *Proceedings of CACSCUK*, (Luton, England), pp. 343–348, September 2003.

[155] C. Curio and M. Giese, "Combining view-based and model-based tracking of articulated human movements," in *IEEE Computer Society Workshop on Motion and Vision Computing*, (Beckenridge, Colorado), January 2005.

[156] H. Sidenbladh, F. D. la Torre, and M. Black, "A framework for modelling the appearance of 3-D articulated figures," in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, p. 368, March 2000.

[157] A. Agarwal and B. Triggs, "Learning to track 3-D human motion from silhouettes," in *International Conference on Machine Learning*, (Banff, Canada), pp. 9–16, 2004.

[158] Curious Labs, http://www.curiouslabs.com.

[159] E. Weisstein, "Rodrigues' rotation formula," in *MathWorld - A Wolfram Web Resource*, (http://mathworld.wolfram.com/RodriguesRotationFormula.html).

[160] J. Heikkila, "Geometric camera calibration using circular control points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1066–1077, October 2000.

[161] S. Shih, Y. Hung, and W. Lin, "Accuracy analysis on the estimation of camera parameters for active vision systems," Technical Report TR-IIS-96-006, Inst. Inform. Sci., Acdemia Sinica, Taipei, Taiwan, 1996.

[162] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3-D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 322–344, August 1987.

[163] W. Sun and J. Cooperstock, "Requirements for camera calibration: Must accuracy come with a high price?," in *IEEE Workshop on Applications of Computer Vision*, (Breckenridge), pp. 356–361, January 2005.

[164] M. Armstrong, *Self-Calibration from Image Sequences.* PhD thesis, University of Oxford, England, 1996.

[165] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge Univeristy Press, second ed., 2003.

[166] Point Grey Research, Vancouver, B.C., Canada, *Dragonfly Specifications*, 2004.

[167] L. Campbell and A. Bobick, "Recognition of human body motion using phase space constraints.," in *International Conference on Computer Vision*, pp. 624–630, 1995.