Human Modelling from Multiple Views

J.R. Starck

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Centre for Vision, Speech and Signal Processing
School of Electronics and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

July 2003

Summary

A long standing problem in computer graphics and animation is the production of synthetic computer graphics models whose appearance, movement and behaviour are visually indistinguishable from the real world. This thesis addresses the problem of reconstructing visually realistic computer graphics models using multiple camera views of real people. A model-based computer vision algorithm is introduced to reconstruct the shape and appearance of a person in an arbitrary pose viewed in a multiple camera studio.

Current techniques for multiple view reconstruction address the problem of general scene recovery. These non model-based approaches can fail to accurately reconstruct shape and appearance in the presence of visual ambiguities. The techniques also provide no structure to edit or reuse the captured content in computer animation. The primary novel contributions in this research work are 1) a shape constrained deformable model formulation to match a generic model to shape information in multiple view silhouettes in the presence of visual ambiguities; and 2) a model-based multiple view reconstruction algorithm to recover a model that matches appearance across multiple views to sub-pixel accuracy.

Model-based multiple view reconstruction of people is evaluated and results are presented for the reconstruction of shape and appearance of people in an arbitrary pose. The recovered models provide a accurate shape representation for a person and a visual appearance approaching the quality of the original camera images. The models also provide a consistent structured representation for the editing, synthesis and transmission of 3D content in computer graphics and animation.

Key words:
Visual Scene Reconstruction, Three Dimensional Graphics, Human Models.

Email: j.starck@eim.surrey.ac.uk

WWW: http://www.eim.surrey.ac.uk/

Acknowledgements

I would like to thank my supervisor, Dr. Adrian Hilton, for the opportunity to undertake this research and for his thoughtful guidance throughout. My heartfelt thanks go to my wife Emma and my parents Beryl and Brian who have always given me considerable support and encouragement. Finally I should like to thank all my friends and colleagues at CVSSP and within the PROMETHEUS consortium for their help in this work. This research project was supported financially by PROMETHEUS, an EPSRC/DTI Broadcast Link project (EPSRC Grant GR/M88075).

Contents

List of Figures

Mathematical Notation

In this thesis scalar quantities, vectors, matrices and functions are written in italics. In general, vectors are denoted with lower case letters and underlined such as $\underline{x}$. Matrices are represented with capital letters in bold type, such as $\mathbf{P}$. Functions are indicated in calligraphic letters, such as $\mathcal{H}$. The symbols used in the thesis are set out as follows.

Indices

Symbol	Meaning
i	Vertex on a model mesh
i'	Vertex in 1-neighbourhood of vertex i
j	Voxel on surface of visual-hull
k	Reconstructed point from stereo correspondence
c	Constraint point
s	Sample point
f	Triangle facet on a model mesh
g	Gaussian image
l	Laplacian image
$N.$	Number of indices

Camera image data

Symbol	Meaning
$\mathbf{P}$	Projective transformation to an image plane
$\underline{I}$	RGB camera image
I	Camera intensity image
σ	Standard deviation of pixel intensity
$\underline{I}^{TEX}$	RGB texture image
M	Mask image
$\mathcal{C}$	Correlation score between two intensity images

Two-dimensional data

Symbol	Meaning
$\underline{u}$	Image plane location
$\underline{U}$	Image location with a height component
H	Height from an image plane
$\underline{u}^{TEX}$	Image location in model texture map
$\underline{p}$	Image pixel
$l_{ii'}$	Edge length connecting two vertices in an image
$k_{ii'}$	Spring stiffness for an edge length

Three-dimensional data

Symbol	Meaning
$\underline{x}$	Position on a model surface
$\underline{y}$	Position on visual-hull surface
$\underline{z}$	Reconstructed point from stereo correspondence
$\underline{o}$	Observed point
$\hat{\underline{n}}$	Normalised direction vector
(α, β, h)	Barycentric and height coordinates in a triangle-centred frame
v	Visibility
d	Distance across the surface of a model

Sparse data interpolation

Symbol	Meaning
$\mathcal{R}$	Radial basis function for sparse data interpolation
η	Parameter defining the influence of a radial basis function
$\mathbf{A}$	Three-dimensional affine transformation matrix
$\mathcal{H}$	Interpolation function

Model texture

Symbol	Meaning
$\widetilde{v}_f$	Target visible image assignment for a model triangle
ϕ	Angle subtended at a vertex with respect to camera viewpoints
b	Blend weight in view-dependent rendering

Deformable model

Symbol	Meaning
$\mathcal{E}$	Energy function for a deformable model
$\mathcal{E}_{\mathcal{D}}$	Data energy term
$\mathcal{E}_{\mathcal{R}}$	Regularisation energy term
λ	Influence of regularisation in $\mathcal{E}$
w_{ij}	Assignment of vertex i to voxel j
w_{imk}	Assignment of vertex i to stereo point k in image m
μ	Influence of stereo data
$\sigma_{0.5}$	Standard deviation of pixel intensity where equal weight is given to stereo and silhouette matching ($\mu = 0.5$)
ν	Influence of sparse feature data
$d_{0.5}$	Surface distance from a feature point where equal weight is given to feature matching and stereo/silhouette matching ($\nu = 0.5$)
τ	Threshold on peak correlation score for stereo correspondence
T	Control temperature in deterministic annealing
c	Annealing schedule
δ	Step length for steepest descent optimisation

Operators

Symbol	Meaning
$\nabla\cdot$	Gradient of a function
$\mathbf{J}\cdot$	Jacobian matrix of function derivatives
$\triangle\cdot$	Change in value

Chapter 1

Introduction

Computer generated three-dimensional (3D) graphics has found widespread use throughout many forms of visual media. One of the key challenges for 3D content production is the creation of visually realistic models of people. Model production is currently a high-cost and labour-intensive task, and the results often appear synthetic when compared to the world we see around us. Computer vision offers the opportunity to capture these graphical models directly from the real-world with the visual realism of conventional video images.

In this thesis a framework is introduced to reconstruct a 3D computer graphics model of a person from multiple camera views. Previous multiple view reconstruction techniques concentrate on capturing the 3D geometry of a general scene and provide no structure to edit or reuse the 3D content. The objective of this work is to derive a controllable animated 3D model of people from images. Such a model provides the potential to reproduce the complex shape and appearance we are accustomed to in conventional video images and to allow the manipulation of model dynamics and viewpoint for 3D content production.

1.1 Motivation

The challenge of creating a realistic computer generated human has always been a central goal in computer graphics. The film industry, for example, has seen an explosion

in the use of computer graphics over the past two decades and realistic synthetic actors or "synthespians" have always been just around the corner. In 1986, the Pixar group at Lucasfilm produced the first 3D character, a knight who leaps from a stained glass window, in the film Young Sherlock Holmes. In 1991 Industrial Light and Magic produced the first high profile digital character, a liquid-metal humanoid robot, that performs key sequences throughout the film Terminator II. In 2001 we saw the first film distributed by a major Hollywood studio to feature a cast made up entirely of "hyper-realistic" computer generated humans, in the film Final Fantasy.

Computer graphics technology has an obvious advantage in the film industry where computer generated imagery enables a director to create and edit any situation without the dependence on live action shots. The greatest success for realistic human models has been as stunt doubles for "mixed reality" clips where seamless transitions between live action and computer graphics are used to enable special effects and camera shots that would not otherwise be possible. However, the use of realistic models as principal characters in films has received a mixed response. Production of a human model is a challenging task and computer generated realism can appear synthetic and impersonal compared to the live action that we are accustomed to.

Currently the production of computer generated human characters is a high cost and labour intensive task, limiting the application to the big-budget film, advertising and game industries. In low budget applications such as television broadcast, use of computer graphics has instead concentrated on the "Virtual Studio". In the virtual studio, a camera films live action against a constant background such as a blue-screen. The key colour in the background is removed and the scene is combined with a virtual set or normal video. Camera tracking information can also be incorporated to allow the camera to move, pan, tilt or zoom within the virtual set or within the video to provide matched virtual camera movements [67]. Virtual studio technology has now developed to the point where action can be overlaid live with the real or synthetic video.

Virtual studio production is principally used to composite shots of actors with real or virtual backgrounds. Everyday examples include news and weather reports or natural history and scientific documentaries where a presenter is composited with a video of a

remote location or computer graphics. Recently the concept of using multiple cameras in a virtual studio to capture a scene in 3D has been introduced [65]. Three-dimensional production or 3D video was first popularised by Kanade et al. in 1997 [87] who coined the term "Virtualized Reality". Conventional video provides only a two-dimensional (2D) view of a scene in a linear form where the director defines every moment to be viewed. Presenting an event in 3D allows visualisation in the same way as virtual reality to give an immersive 3D experience in which a viewer has the freedom to control and interact with the scene in three dimensions.

In this thesis, the convergence of computer graphics modelling and 3D virtual studio production is explored. Computer graphics models of people provide the freedom to manipulate characters to create and edit a desired event. However, this can come at a prohibitive cost due to the skill and time required to generate the models and does not meet the realism we are accustomed to in conventional video. The 3D virtual studio on the other hand provides the potential to capture real dynamic scenes in 3D with the realism of video. This work investigates the reconstruction of computer graphics models of real people from multiple view video. Such models can capture the complex shape and appearance of a person available in video images and allow the manipulation of the model dynamics and viewpoint in visualisation as a computer graphics model. The ultimate goal is to capture 3D models of people with the same visual quality as the original video to give a "video-realistic" model.

The potential application of a "video-realistic" human computer graphics model extends beyond improving the realism of current models in games and films, or providing the means to generate 3D content in television and multi-media. A system for capturing controllable models of people could be used to simulate any sort of human interaction, for example in virtual rehearsals for broadcasting, simulations for medical or safety training, and interactive educational services. Models captured for individuals could be used to personalise automated or electronic communications such as video call centres, e-mails, or teleconferencing. Such models could also be used for the personal design of equipment or clothing. This technology has the potential for a widespread impact across the broadcast, entertainment and communication industries.

1.2 The PROMETHEUS project

This research forms part of the PROMETHEUS project: PROduction of multi-MEdia content for THree dimensional Environments distribUted over networkS. The PROMETHEUS project funded by EPSRC/DTI Link Broadcast program (1999-2003), coordinated by the BBC, was set-up in anticipation of the future of television as a fully interactive 3D medium or "Virtualized Reality". The objective was to develop production tools for a 3D virtual studio and demonstrate content creation, transmission, and display as a full 3D programme chain. This was show-cased at the International Broadcasting Convention, IBC 2002 [132].

PROMETHEUS was a collaborative project with the BBC, AvatarMe, BTexact Technologies, De Montfort University, Queen Mary University of London, Snell and Wilcox, University College London, and the University of Surrey. The work in this thesis formed the basis for creating realistic whole-body models of actors in a broadcast studio. Other work at the University of Surrey explored the estimation of human motion from multiple views to define the animation parameters for an actor model. BTexact Technologies addressed the problem of separately acquiring a model for the human face and tracking facial motion in a camera image to derive facial animation parameters. University College London's Virtual Environments and Computer Graphics group developed a real-time cloth simulation system to integrate virtual clothing with an animated actor model. The delivery of this 3D content was addressed using the MPEG-4 model-based coding standard by Queen Mary, University of London. De Montfort University explored techniques to render the content to a 3D display. Finally the BBC coordinated and integrated this work to demonstrate the potential for a complete 3D programme chain.

1.3 Overview

The overall objective of the research presented in this thesis is to recover visually realistic models of people from multiple view images. Video capture is based on the concept of the 3D virtual studio in which dynamic scenes of actors are captured against

Figure 1.1: Camera images in a 3D Virtual Studio, (courtesy of BBC R & D).

a controlled background setting with multiple camera views. The problem is to derive a 3D representation of a person from these images as a controllable computer graphics model. Figure 1.1 shows a typical set of multiple view camera images recorded in a broadcast studio.

The reconstruction of 3D shape from camera images forms a central problem in Computer Vision and techniques have been developed previously to construct models from multiple views [87, 113, 175]. In recent years the problem of generating visually realistic models from images has been addressed. Image-based representations for appearance have been combined with geometric scene reconstruction to give highly realistic 3D models with the visual quality of camera images. These techniques have been applied to capture the shape and appearance of people from multiple views in a studio. Kanade et al. [87] first demonstrated the ability to recover dynamic scenes of people from multiple video images as an off-line process. Matusik et al. [113] presented a system to generate a virtual camera image in real-time from multiple views. Vedula [175] introduced a technique to generate virtual views and interpolate the visual appearance of a person across time as well as space to create re-timed special effects.

The goal here is to capture controllable 3D models of people. Previous techniques for multiple view reconstruction [87, 113, 175] have been based on recovering the geometry of a dynamic scene containing a person. These techniques make no assumption on the structure of the scene and generate a new scene model for each time frame from a multiple view video sequence. The advantage of this is that there are no restrictions

on the content of the scene. However, this approach has a number of limitations: (i) The scene models do not necessarily have a consistent structure over time and inconsistencies in the models at different time frames will become apparent when viewed as a sequence; (ii) Without any prior knowledge of the scene geometry, reconstruction can be sensitive to visual ambiguities leading to incorrect shape and appearance in generating novel views; (iii) There is no temporally consistent structure to encode a scene for transmission; and most importantly (iii) There is no consistent structure to edit or reuse the dynamic content in a scene, limiting the techniques to replaying a capture event.

The objective of the work in this thesis is to overcome the limitations of current techniques for multiple view reconstruction and capture a scene model with the necessary structure for model animation. The approach is based on the functional modelling paradigm introduced by Terzopoulos et al. [162], which has been proposed for whole-body modelling of people from a single camera by Hilton et al. [79]. Reconstruction is based on a prior humanoid model which is adapted to match the shape and appearance of a person across multiple views. This has a number of important advantages: (i) The prior model can be designed for rendering and manipulation as a standard computer graphics model; (ii) The model can be instrumented with a kinematic structure for model animation; (iii) A model provides prior shape information to influence multiple view reconstruction in the presence of visual ambiguities; and (iv) Model-based reconstruction provides a consistent structure either for a temporal sequence of a person or for the analysis of the shape and appearance of different people.

1.4 Outline of thesis

In this thesis a model-based framework is introduced to reconstruct a computer graphics model of a person from images captured in a multiple camera 3D virtual studio. The thesis is organised as follows. Chapter 2 presents a literature survey of research on human modelling and multiple view reconstruction in Computer Vision and Computer Graphics. In Chapter 3 previous work on reconstructing whole-body models of people from a single camera presented by Hilton et al. [79] is considered. The model-based

technique using shape from silhouette is extended to recover the shape of a person in an arbitrary pose viewed from an arbitrary camera position in a multiple camera studio. This approach is found to be limited by the independent treatment of shape in each 2D image. In Chapter 4 a model-based technique to simultaneously recover the shape of a person from multiple image silhouettes is presented. The technique provides an approximate model for a person that satisfies the 3D constraint imposed by image silhouettes from multiple views. Chapter 5 then presents a model-based approach to match a prior model to refine the shape to match the appearance in the colour images to derive a more accurate shape model for the recovery of surface appearance. A technique is presented to update a prior model to match silhouette, appearance and feature data across multiple views. The appearance of the model is then derived in Chapter 6 from the camera images. Appearance is defined through either a view-independent surface colour for the model or an image-based representation allowing for a view-dependent appearance that reproduces the captured images. The techniques presented are evaluated in each chapter and Chapter 8.1 presents final results model-based reconstruction of people in a multiple camera studio. Finally, the thesis is concluded with a summary of the achievements and suggestions for future work.

Through-out this work two assumptions are made. Firstly it is assumed that the cameras in the studio are calibrated in order to perform a metric 3D reconstruction from the 2D images. Appendix A outlines the pinhole camera model adopted, the calibration of multiple cameras in a studio and the reconstruction of 3D position from multiple views. The second assumption is that the prior humanoid model for model-based reconstruction is registered to match the pose of a person in multiple images. In Appendix B a technique is described to register an animated model with a person through the use of a manual user interface to define feature points for the model in multiple images.

1.5 List of publications

A number of publications have resulted directly from this work.

- J.Starck, A.Hilton, and J.Illingworth. Human shape estimation in a multi-camera studio. In *British Machine Vision Conference* volume 2, pages 573–582, 2001.
- A.Hilton, J.Starck, and G.Collins. From 3d shape capture to animated models. In *1st International Symposium on 3D Data Processing Visualization and Transmission*, pages 246–255, June 2002.
- J.Starck, G.Collins, R.Smith, A.Hilton, and J.Illingworth. Animated statues. In *Machine Vision and Applications, Special Issue on Human Modeling, Analysis, and Synthesis*, 2002.
- J.Starck, A.Hilton, and J.Illingworth. Reconstruction of animated models from images using constrained deformable surfaces. In *10th International Conference on Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science*, volume 2301, pages 382–391, 2002.
- J.Starck and A.Hilton. Towards a 3D virtual studio for human appearance capture. In *Vision, Video, and Graphics*, 2003.
- J.Starck and A.Hilton. Model-based multiple view reconstruction of people. *IEEE International Conference on Computer Vision*, 2003.

In addition the complete work in the PROMETHEUS project has been published in.

- M.Price, J.Chandaria, O.Grau, G.A.Thomas, D.Chatting, J.Thorne, G.Milnthorpe, P.Woodward, L.Bull, E-J.Ong, A.Hilton, J.Mitchelson, and J.Starck. Real-time production and delivery of 3D media. In *Proceedings of the International Broadcasting Convention*, 2002.

Chapter 2

Literature Review

The challenge of achieving computer generated realism is leading to a convergence of Computer Vision and Computer Graphics techniques. This is demonstrated by the increasing use of vision in the graphics community [49, 114, 182], and work on graphics applications in Computer Vision [123, 141, 79]. Computer Graphics deals with the problem of constructing models and rendering images from them. Realistic image synthesis is a central goal and research has focused on modelling the complex geometry and material properties of real world objects, and simulating the complex path and surface interaction of light in order to generate realistic renderings. While this can provide highly realistic results, the problem of creating models that are indistinguishable from reality remains unsolved. The alternative approach pioneered in Computer Vision is to create new images from photographs to produce "photo-realistic" computer generated scenes.

Computer Vision concerns the problem of modelling and interpreting the world through the analysis of images. This in part addresses the problem of constructing geometric models from images, termed image-based modelling. The use of photographs or video offers a simple and attractive method to capture the geometry and appearance of real objects. In the simplest form a photograph can be incorporated as an image on a computer graphics model, termed a texture map, providing visual detail without the need for complex physical simulation of geometry and light transport in a scene [77]. At its most complex, a vision system with accurate camera models and estimates of

camera position and orientation can be used to capture models of geometry or scene appearance in order to synthesise realistic images of the scene from new view-points [45].

In this chapter current approaches are reviewed for image-based modelling and rendering from images in Computer Vision and Computer Graphics. Current methods for constructing computer graphics models of humans from images are then presented. Finally the literature is summarised and conclusions drawn on appropriate techniques for constructing human models in the target application area of the multiple camera studio.

2.1 Image based modelling and rendering

There are two contrasting approaches to the problem of synthesising new views from photographs: image-based modelling; and image-based rendering. In modelling from images, a 3D surface model is constructed for a scene and texture maps are extracted from the images. The advantage of this approach is that it allows the model to be manipulated both in visualisation through a conventional computer graphics rendering pipeline, and in modification of shape and appearance through 3D modelling software. The disadvantage lies in the quality of the geometric reconstruction that can be achieved from images and the fixed appearance given by the model texture.

Image-based rendering on the other hand represents a scene by the original photographs rather than explicit reconstruction of scene geometry. Novel views are synthesised by resampling the images. This provides the visual fidelity of the original data and reproduces the complex view-dependent lighting effects captured in the images. The drawback of the approach is that a scene must be captured sufficiently densely in order to synthesise new views, requiring restricted viewpoints to avoid a prohibitive number of sample images. This representation also provides no facility to manipulate the data in order to edit or generate new scenes.

While image-based rendering appears removed from the goal of reconstructing controllable models of humans from images, hybrid approaches have been presented com-

bining the key advantages of each representation. Hybrid techniques make use of an approximate geometric representation of a scene to provide the correspondence between views to synthesise new images from a set of sample images. This provides a conventional representation for 3D manipulation and simulates view-dependent appearance with a reduced set of images compared to image-based rendering. In this section both geometry-based, image-based and hybrid methods of scene representation are reviewed.

2.1.1 Geometric reconstruction

Reconstruction of 3D data from 2D images forms a central problem in Computer Vision from applications such as robot navigation or object recognition to scene capture for virtual environments. Techniques for reconstruction originated in the field of photogrammetry where it was recognised that photographs could be used to generate topographic maps, a technique that gained widespread used in World War I [48]. Reconstruction is based on the triangulation of 3D position from the 2D location in two or more images. A photograph provides a projection of the 3D scene onto a plane and a feature on the plane corresponds to a ray that extends from the centre of projection of the camera into the scene. If the feature is located in another image then the 3D location is given by the intersection, or triangulation, of the corresponding rays [56]. Triangulation can be used to interactively construct 3D models and several commercial software packages such as PhotoModeler [7] and ImageModeler [11] are available to do this. Manual construction of models is however a laborious task in which every feature that defines the 3D shape of the scene has to be marked in the captured images.

Computer vision research has focussed on the problem of automating scene reconstruction from camera images [56, 75, 173]. The basis of multiple view geometry has been developed in computer vision over the last two decades to define the parameters involved, the constraints between features in images, the estimation of camera parameters and the reconstruction of 3D position from image correspondences [75]. The classical approach to 3D reconstruction developed first in photogrammetry attempts to jointly estimate 3D structure and camera viewing parameters through a process termed *bundle-adjustment* [171]. In computer vision this problem is addressed

in *structure-from-motion* [56] where the task is to derive the 3D position and camera motion for an image sequence. In visual scene reconstruction this problem is simplified somewhat by calibrating the viewing parameters of the cameras. The problem is then to solve for the 3D shape of the scene that reproduces the images. Techniques for visual scene reconstruction can be broadly divided into passive methods that rely on matching visual cues between images, and active methods in which light patterns are projected into a scene to provide visual features for matching.

Active systems for reconstruction called range scanners employ either a time-of-flight approach, in which a focussed laser pulse is emitted and the time to return to a sensor gives the distance travelled, or use triangulation to locate the 3D position of light projected onto an object [43]. A range scanner provides 3D depth data within the field of view, termed a range map or 2.5D image. Commercial scanners are available to capture the shape of a person such as the WB4 from Cyberware [5], TriForm from Wicks and Wilson [14], Vitrus pro from Vitronic [13], and the Body Line Scanner from Hamamatsu [8]. These systems produce high accuracy and high resolution 3D computer graphics models. The Cyberware WB4 reconstructs human models to an accuracy of 0.5mm with a typical polygon count in the order of 300000. Such systems are however limited to the acquisition of static scenes in a restricted environment and often provide only limited colour information for the captured models.

Compared to active sensing techniques, passive scene reconstruction from images enables greater flexibility in scene capture, provides all the colour information inherent in the camera images and enables the capture of dynamic events. Passive techniques rely on matching visual cues such as features, surface appearance, shading and silhouette contours. This has been applied to capture 3D data from monocular images, stereo pairs, multiple camera views and image sequences from moving cameras. Visual scene reconstruction from images suffers from inherent ambiguities in deriving 3D information from 2D appearance and the robust reconstruction of accurate scene models remains an open area of research. Commercial applications include Boujou from 2D3 [1], an automated camera tracking tool for the composition of computer graphics with real video and the 3D Software Object Modeller from Canon [4] which uses shape from silhouette to reconstruct small objects from a single camera.

In this review multiple view reconstruction techniques are considered for the problem of recovering geometry from images in a multiple camera studio. With multiple fixed cameras, full calibration data can be derived by observing a known calibration object [56]. A camera is said to be "fully" calibrated if the extrinsic camera parameters, the position and orientation in space, together with intrinsic parameters that define the 3D to 2D projection onto a camera image plane are known. There are two different approaches to the problem of recovering shape from multiple calibrated cameras. In Section 2.1.1.1 the stereo vision approach is described in which a search is performed to locate matches between images and 3D position is reconstructed by triangulation. In Section 2.1.1.2 the approach used in volumetric scene reconstruction is described in which the 3D surface that is consistent with the 2D images is derived instead.

2.1.1.1 Stereo vision

Stereo vision is the process of inferring 3D depth from a pair of camera images, a problem that has been extensively researched in Computer Vision. The stereo reconstruction problem is reviewed by Dhond and Aggarwal [51], and more recently by Szeliski [157]. The problem is to derive the correspondence between images such that matched image points correspond to the projection of the same points in the scene. The amount of displacement, or disparity, between matched points can then be related to depth to give a 2.5D scene representation as shown in Figure 2.1. Stereo vision is used later in Chapter 5 to derive the geometry of a model that matches the appearance of a person between camera images.

In general, stereo correspondence is solved through a search for matching points based on a measure of similarity between the images. A number of factors complicate this search for correspondence [63]: (i) Measuring the similarity between images requires a local variation in appearance to differentiate between correct and incorrect matches; (ii) Several similar points may then be found in the search; (iii) Occlusions in the scene will make the correspondence ambiguous; and (iv) Non-uniform surface reflectance and perspective projective to different view-points will distort the local appearance. All these factors are apparent in images of people where clothing can lack a local variation in

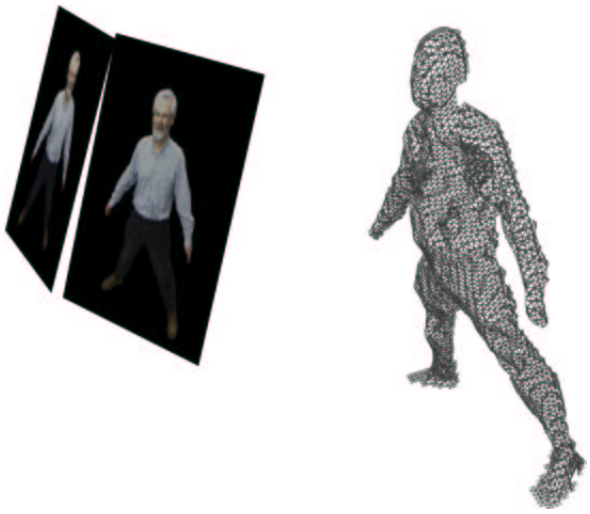


Figure 2.1: The 2.5D depth reconstructed for the left camera of a stereo pair using dense area-based matching with the right-hand camera image.

appearance or have a repeated pattern, articulation of the body leads to self-occlusions, and clothing or skin exhibits a non-uniform view-dependent appearance. The following constraints can be imposed in stereo reconstruction to reduce these ambiguities in the correspondence problem [56].

- **Epipolar constraint:** The epipolar constraint exploits the geometry of a camera system. A point in one image corresponds to a ray in space that in turn projects to a line in another image, the epipolar line. Corresponding image points are therefore constrained to lie on the respective epipolar lines reducing the 2D search for correspondence to a 1D line search.

- **Uniqueness constraint:** The uniqueness constraint requires that a point in one image can only match one point in another image for opaque objects. Cross-checking can then be used to verify correspondences and establish occluded regions where matching is inconsistent.

- **Continuity constraint:** The continuity constraint assumes that a scene consists of smooth surfaces. The disparity derived in stereo matching should therefore vary smoothly for a single surface. This constraint can be applied to reduce the range

of feasible disparities surrounding a given point.

- **Ordering constraint:** The ordering constraint defines that for a single surface, the sequence of matches between points that are visible in both images are ordered. The order of matches will only change for points that are occluded in one image or for points from a different surface.

These constraints reduce mismatches in establishing corresponding points between images. The epipolar geometry of a camera system provides an important constraint that also reduces the complexity in the search for stereo correspondence. Image rectification [64] can be used to transform camera images so that the epipolar lines become parallel to an image axis and the search for correspondence can be performed efficiently along an image scan-line. This constraint will fail where inaccuracies in camera calibration lead to an incorrect estimate of the epipolar geometry and the rectified images will not then match along the scan-lines. The uniqueness constraint allows erroneous matches to be removed after matching and occluded regions to be identified. In "left-right consistency" [59] uniqueness is verified by ensuring that the match found in the right camera image of a stereo pair, with respect to a left camera point, should return the left point as a match when searching for correspondence in the left camera image. The drawback of the uniqueness constraint is that establishing the similarity of points is inherently ambiguous and a consistent correspondence may only be found for points with a sufficiently distinct local appearance. The ordering and continuity constraints impose a constraint on the shape of an object to be recovered. These constraints have lead to the use of global optimisation schemes such as dynamic programming to extract continuous surfaces that preserve the order of matches and that are locally connected [105, 41, 155]. These techniques can remove the noisy matches obtained in independently matching points between images, but are restricted to the reconstruction of a single discontinuous visible surface.

Methods for establishing correspondence are broadly divided into area-based and feature-based matching. Image features are regions that exhibit a distinct appearance with a relatively large change in intensity and are inherently sparse. Area-based methods match all image pixels between views and hence provide dense 2.5D geometry as shown

in Figure 2.1. Area-based stereo matches small windows between images with a metric that measures the aggregated match of corresponding window pixels. In traditional area-based stereo, a fixed window size is used and pixel intensity is correlated or the sum of squared intensity differences (SSD) is measured [86]. The correspondence is established where the windows have either a maximum correlation or minimum SSD.

Area-based stereo has a number of limiting assumptions. Firstly it must be assumed that the local relative change in intensity within an image window is consistent between views so that pixel intensity can be correlated. Secondly it is assumed that there is sufficient intensity variation within a window to enable reliable matches between images. Finally it is assumed that disparity is constant within a window, equivalent to a fronto-parallel surface, so that there is no geometric distortion between views. The later assumptions lead to a tradeoff between the size of an image window needed to obtain a sufficient variation in intensity and the need to limit the window size to maintain a constant disparity. Adaptive window schemes have been proposed to change the window used to automatically compensate for a lack of intensity variation and minimise the error in regions of non-constant disparity [86, 63]. These techniques demonstrate improved performance compared to fixed window area-based stereo reducing the noise on the estimated disparity.

In order to minimise the geometric distortion between camera views for dense area-based stereo a short baseline is required between camera positions such that the scene appears similar in the images. The drawback of this is that the depth resolution decreases as the baseline is reduced and large errors in computed depth can result from small errors in disparity. This trade-off between reconstruction accuracy and camera baseline has been improved by combining more than two cameras in dense stereo reconstruction. Here a common 2.5D depth map is constructed with respect to a reference image from multiple camera pairs. In trinocular stereo [56] three cameras are used to form two stereo pairs. The correlation scores in the two pairs are then combined to establish correspondence. In multiple-baseline stereo, Okutomi and Kanade [126] arrange cameras in a linear parallel configuration so that the pair-wise correlation can be summed for more than two camera pairs with respect to a common underlying depth. The use of multiple camera pairs in trinocular stereo and multiple baseline stereo allows

the noisy stereo estimates from individual pairs to be integrated to allow a more precise estimate of depth.

In area-based stereo the baseline between cameras in a stereo pair is restricted. Feature-based stereo on the other hand allows for a much wider baseline between cameras by only matching distinct image features between views. Features are defined for matching in the images where there is a distinct appearance that is stable to changes in viewpoint. The Harris interest point detector [74] has been used to define point features [19], points have been linked to define line segments [160], line segments have been linked to form planar regions [133], and periodicity in appearance has been used to define "distinguished" image regions [33]. Matching is based on descriptors that define the appearance of these features with changes in position, scale, rotation and skew between viewpoints [19, 69]. Image features have a locally unique appearance and feature matching can provide a reliable correspondence between views. However, features are sparse and provide only limited shape information in a scene requiring interpolation to recover a complete geometric model. The reliability of feature based correspondence has been combined with the resolution of dense stereo matching. Konrad and Lan [89] proposed a technique to constrain the feasibility disparity ranges for area-based stereo using feature matches. However, recovering dense geometry with an area-based approach still requires a relatively short baseline between cameras to minimise the distortion between the appearance in the image windows for matching.

Area-based stereo can provide the dense geometric detail needed to model people from multiple camera views. Methods are limited however by the requirement for a short camera baseline for matching which leads to noisy 3D estimates. Area-based matching also requires a sufficient local variation in appearance to recover correspondence. Matching will fail where there is a uniform appearance, a repeated pattern of appearance or where there is occlusions between views. The correlation of image windows requires the assumption that the local relative change in intensity is consistent between images and that the depth is constant across the window size chosen.

Stereo vision has been used previously to reconstruct models of people from multiple cameras in a studio. Kanade et al. [87] and Narayanan et al. [123] describe a "Virtu-

alized Reality" system with 51 cameras located on a hemi-spherical dome. The system constructs a separate 2.5D depth map for each camera using the multiple-baseline stereo technique introduced by Okutomi and Kanade [126]. The depth maps are integrated into a single polygonal surface model and texture mapped by projecting the colour images back onto the model. A large number of algorithms have been developed to solve the stereo correspondence problem with area-based stereo. Scharstein and Szeliski [138] present a comparative evaluation of different techniques.

2.1.1.2 Volumetric reconstruction

Volumetric scene reconstruction operates in the 3D domain to recover the volume of the scene that is consistent with the images. The advantage of this approach is that it removes the need to search for image correspondence, the problem instead is to determine image consistency for a particular 3D point. Widely separated views can then be considered and the occlusion of scene points between views can be modelled. In Chapter 4 volumetric reconstruction is used to recover an approximate shape of a person from multiple view silhouettes and in Chapter 5 volumetric reconstruction from image colour is considered to refine the shape from silhouette.

Volumetric model reconstruction from photographs originated in recovering geometric models from image silhouettes. Martin and Aggarwal, 1983 [111], first described a method for building models from multiple images based on the "occluding contour" of the silhouette, the boundary constraint on the 2D image-plane area of a scene. Each silhouette contour, together with the center of projection for the camera, forms an infinite cone in space that places a constraint on the volume occupied by the observed scene. The intersection of the cones then defines a bounding volume that approximates the shape of the scene as illustrated in Figure 2.2. Laurentini [95] introduced the concept of the visual-hull as the closest 3D approximation that can be obtained from image silhouettes taken from all possible views. In this thesis the term visual-hull is used to refer to the volume that is reconstructed by the intersection of a number of silhouettes, rather than the volume at the limit from an infinite number of silhouettes.

Shape from silhouette has a number of limitations. The first is the requirement for

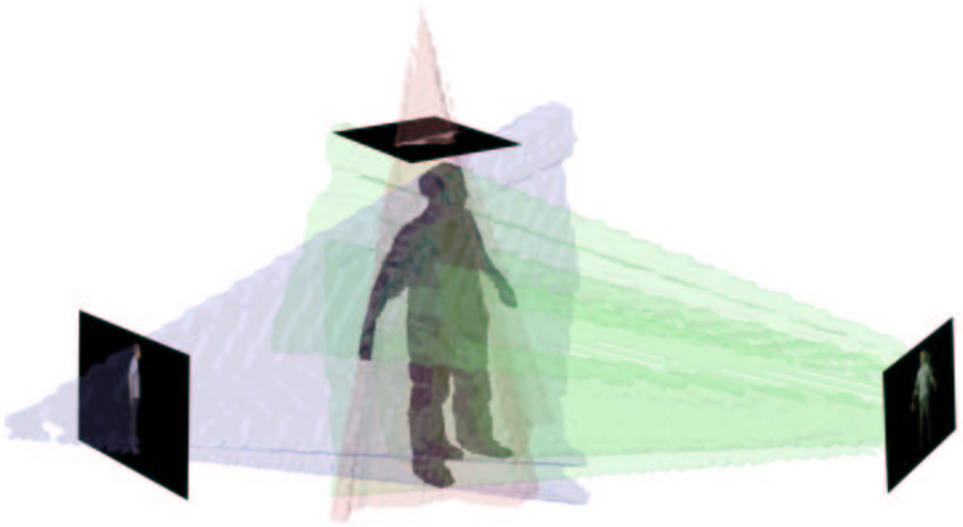
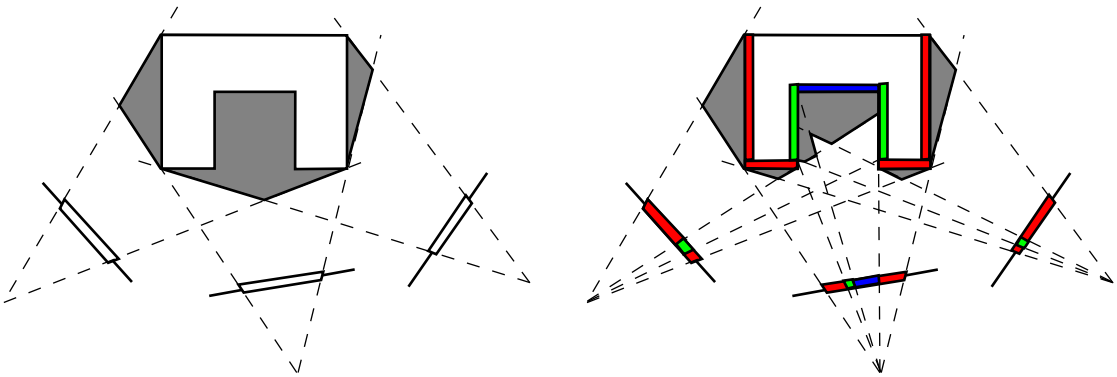


Figure 2.2: Reconstruction of the visual-hull from the volume intersection of the visual-cones from three orthogonal image silhouettes of a person.

controlled background conditions and illumination to enable foreground silhouette extraction without shadows in all images. Foreground extraction makes the assumption that the foreground scene has a sufficiently distinct colour appearance to the background for segmentation. Segmentation will fail where the scene has the same colour as the background or shadows lead to a change in appearance for the background. The second limitation is that the visual-hull cannot model concavities in the scene as concave regions will be self-occluded in the silhouette images as illustrated in Figure 2.3(a). Finally the shape information from multiple silhouettes is only guaranteed to provide an upper bound that will contain the true surface of a scene as shown in Figure 2.3(a).

Techniques for volumetric reconstruction of the visual-hull in general use a discrete representation of space as a set of volume elements or voxels [53, 148]. The voxels corresponding to the visual-hull are extracted by intersecting the visual cones for the silhouettes. This intersection test, also called the voxel occupancy problem, is performed by projecting voxels to each image in turn and testing the overlap with the silhouettes [156]. A discrete volumetric representation can be memory intensive and the occupancy test must be performed for a large number of voxels. With a volumetric



(a) Visual-hull from silhouettes (b) Photo-hull from image colour

Figure 2.3: Volumetric reconstruction with concavities in a scene. The volume is shown shaded for (a) the visual-hull from image silhouettes and (b) the photo-hull from image colour.

grid of N elements on each side, the number of voxels to be stored and tested is $O\left(N^3\right)$. Efficient representations have been used to make scene traversal more efficient. Niem and Buschmann [124] consider a row of voxels at a time and test the intersection of the projected image line with each silhouette. Spatial partitioning with an octree structure has been used in which the voxel space is defined as a hierarchy of subdivided cubes. Szeliski [156] describes a method to iteratively refine an octree model by dividing voxels that project to both silhouette and background elements of an image, the octree then corresponds to the surface elements of the scene with an order $O\left(N^2\right)$ rather than the entire volume.

A discrete representation of the visual-hull will introduce quantisation artifacts in the reconstructed 3D shape of a scene. An alternative approach is to explicitly construct the visual-cone for each silhouette and perform the intersection in 3D using constructive solid geometry. Matusik et al. [112] describe a technique to perform this intersection efficiently in image space and derive a polygonal model for the visual-hull. This technique removes the quantisation in a voxel representation and provides a model that exactly reproduces the silhouettes. The authors also demonstrate that models can be derived in real-time, although this will be at the expense of the resolution of the models which in turn introduces quantisation artifacts.

Silhouettes provide binary image information to constrain shape. Voxel colouring introduced by Seitz and Dyer [140] instead use image colour to constrain scene reconstruction. The assumption is made that the surfaces in a scene follow the Lambertian reflectance model [81] and have a constant reflected appearance across all possible views. The voxel occupancy problem is then solved by testing the colour consistency of each voxel across all views for which the voxel is visible. Voxels that are inconsistent are removed from the scene, a process termed voxel carving, leaving the coloured volume that reproduces the images. Kutulakos and Seitz [92] term this volume the photo-hull, the maximal volume that is colour consistent with the original images. This approach allows the reconstruction of concave regions in a scene where sufficient colour information is available to distinguish different surfaces as illustrated in Figure 2.3(b).

Voxel consistency is tested in all camera images from which a voxel is visible. The visibility of a voxel must therefore be checked in each camera view and the consistency retested with any changes in visibility. The *Voxel Coloring* algorithm presented by Seitz and Dyer [140] imposes the "ordinal visibility constraint" on camera locations to simplify the computation of visibility and allow the scene to be reconstructed in a single scan of the voxels. If cameras are positioned on one side of the scene volume then voxels can be visited as a set of planes in a near to far order. This constraint ensures that for any voxel, all the potential occluding voxels will have been already encountered and tested in a previous plane.

The ordinal visibility constraint provides an efficient method to perform voxel colouring. However, the restriction on camera position means that not all surfaces in a scene will be visible and so cannot be reconstructed. Kutulakos and Seitz [92] introduced *Space Carving* in which voxel colouring is performed in multiple plane sweeps to allow for arbitrary camera positions. Space Carving tests voxels in a near to far order for the subset of cameras that have already been passed by a plane of voxels. This plane sweep is performed iteratively along the positive and negative direction of each orthogonal axis of the scene volume, requiring six plane sweeps per iteration. The plane sweeps are terminated when no further voxels are eliminated. This algorithm does not test colour consistency in all views for which a voxel is potentially visible and so can include voxels that are not consistent. Culbertson et al. [42] introduced an alternative approach

termed *Generalized Voxel Coloring* which computes the visibility of each voxel exactly. The technique performs colour carving by iterating over the set of surface voxels in a volume and testing consistency of each voxel in all visible camera views until there is no change in the consistent volume. Culbertson et al. [42] demonstrate that this exact visibility can provide a smaller reprojection error in the original images and more visually realistic scene reconstructions compared to reconstruction with the approximate visibility in Space Carving.

The quality of scene reconstruction in voxel colouring is governed by the colour consistency test. The consistency check used by Seitz and Dyer [140], Kutulakos and Seitz [92], and Culbertson et al. [42] tests the statistical variance of the pixel colours to which a voxel projects across all views. This requires a number of restrictive assumptions.

- **Lambertian reflectance:** The Lambertian reflectance model is adopted such that surface colour can be assumed to be consistent across all viewpoints. Natural surfaces however have a non-uniform reflectance giving a view-dependent colour for a surface;
- **Colour consistency:** The consistency test uses the statistical variance of pixel colours which is not monotonic with visibility [42]. The test should ideally be monotonic such that if a voxel is inconsistent for a given set of images it will remain inconsistent for any superset of images [92]. If the consistency test is not monotonic then voxels may be carved that could then be classed as consistent with a change in visibility as surrounding voxels are removed; and
- **Consistency threshold:** A fixed threshold is applied to the colour consistency metric across all the voxels, leading to the incorrect removal of voxels at colour boundaries. With image quantisation, pixels will have a mixed colour at boundaries changing the apparent colour. With voxel quantisation, a voxel can project to multiple pixels with different colours across a boundary region.

The consistency test proposed by Seitz and Dyer [140] can incorrectly classify the colour consistency of voxels in a scene. Voxels will be incorrectly carved where the colour changes between views due to a non-uniform surface reflectance, or within a view due

pixel and voxel quantisation. Voxels will remain uncarved where there is insufficient colour information in the images to distinguish different surfaces. The consistency test for a single voxel will also influence the classification of the surrounding voxels. Where a voxel corresponding to a true surface is incorrectly carved then voxels inside the scene can become carved. Where a voxel external to a true surface remains uncarved, then the visibility of the surrounding voxels can be reduced leading to further uncarved voxels. Voxel colouring will therefore suffer from false cavities and false convexities in the reconstructed scene. The colour consistency threshold provides a trade-off between these two errors. A low threshold provides a closer fit to the true surfaces in a scene, but increases the chance of mis-carved voxels. A single ideal threshold is not necessarily feasible however as the available colour information and reflective properties in a scene will vary.

Extensions to the original *Voxel Coloring* algorithm have been proposed to overcome the restrictions in testing colour consistency. A comprehensive review of the different techniques for colour carving is presented by Slabaugh et al. [148]. The binary consistency test has been replaced by probabilistic techniques that instead assign an existence probability to each voxel [25, 26], or a multiple-hypothesis approach in which different hypotheses on the colour consistency of a voxel are considered [54, 154]. The reconstructed volume has also been optimised to minimise the reprojection error in the camera images using an iterative refinement of voxel consistency [148]. Inexact camera calibration has been accounted for by testing colour consistency for every pixel within the reprojection error of each camera [93]. Voxels have also been assigned colour plus opacity in order to address the problem of mixed colours with pixel quantisation [158]. These approaches can reduce the error in reconstructing a scene with colour carving, although at a greater computational cost compared to the *Voxel Coloring* algorithm.

Volumetric scene reconstruction has been applied to reconstruct the shape of a person in a multiple camera studio. Moezzi et al. [118, 119] describe a 17 camera system that performs a volumetric reconstruction and the derives a polygonal surface model for rendering. Matusik et al. [112] introduced a technique to reconstruct polygonal visual-hulls and demonstrated real-time reconstruction of a moving person from four cameras. Recently the *Voxel Coloring* algorithm has been applied by Vedula et al.

[175] to reconstruct the shape and motion of a person from 17 camera views. Vedula et al. [175] describe a system to render a virtual view of a person from a smoothed version of the voxel surface without explicit construction of a geometric model. These volumetric reconstruction techniques provide a simple method of recovering the shape or appearance of a person when compared to the 51 camera stereo-vision system used in the "Virtualized Reality" project [87]. The visual-hull provides an approximate shape model for a person from multiple views given a set of segmented foreground image silhouettes. The photo-hull implicitly performs a foreground / background segmentation based on colour and provides the means to reconstruct concavities in the shape where sufficient colour information is available. The visual-hull and photo-hull will however suffer from false convexities where the volume is consistent with the original images, and the photo-hull will also contain false concavities where colour consistency is not correctly determined.

2.1.1.3 Object-centred reconstruction

Image-based modelling suffers from visual ambiguities such as self-occlusion in image silhouettes, insufficient colour information for colour consistency, an insufficient local variation in surface appearance and the presence of surface discontinuities in stereo correspondence. Methods have been developed to overcome these ambiguities by combining different visual cues and incorporating geometric scene information to improve reconstruction.

Fua and Leclerc [61] introduced the concept of an *object-centred* approach to reconstruction in which an initial surface mesh for a scene is reconstructed and then refined. Fua and Leclerc [61] used an initial surface formed by smoothing and triangulating a stereo depth map, then optimised the surface shape to match shading and stereo data across multiple views. The object-centred representation allows multiple shape cues to be incorporated to refine the shape of the estimated surface. The geometric object representation can also account for occlusions between images. Fua and Leclerc [61] demonstrate improvements over the initial reconstructed scene using stereo alone. However, only a limited refinement can be obtained as the technique is based on a local

optimisation of the surface mesh that will fail if the model is more than a few pixels from the correct solution when reprojected to the images [61].

An object-centred approach has been used by Vedula et al. [176] to improve the reconstructed scene models in the "Virtualized Reality" system. Vedula et al. [176] proposed *model-enhanced stereo* in which an initial scene model is constructed by integrating multiple stereo depth-maps. This initial model is then used to restrict the search range for stereo correspondence to produce a refined model of the scene and account for occlusions between images. Vedula et al. [176] show a reduction in the noise on a recovered stereo depth map with *model-enhanced stereo* although it is not clear whether this improvement arises from the refined search for correspondence or the original integration of multiple stereo depth-maps. It is interesting to see from this work that a greater improvement is actually obtained when the authors use hand-edited image silhouettes to reconstruct the visual-hull rather than using stereo reconstruction.

Volumetric scene reconstruction has been combined with stereo-vision in an object-centred approach by Faugeras and Keriven [58]. Faugeras and Keriven [58] present multiple view stereo within a variational framework. A cost function is defined in set of voxels using a stereo consistency metric. Level-set methods are then used to carve away inconsistent voxels by evolving an initial surface to minimise the cost, and hence maximising the stereo correlation between images. This technique has the advantage that it removes the search for correspondence in stereo-vision through the use of a volumetric representation. The evolving surface for a scene also provides an object-centred representation that is used to account for geometric distortions and occlusions between camera views. The technique is however limited by the assumption of a sufficiently unique local intensity variation to avoid local minima in the cost function evolving the surface of the level-set. This approach will fail where there is uniform appearance in the images or a repeated variation in intensity.

2.1.1.4 Model-based reconstruction

Techniques for image-based modelling described so far have dealt with the general problem of reconstructing the arbitrary shape and appearance of an unknown scene

from a finite number of camera views. In *model-based scene reconstruction* a prior model of the expected scene geometry is refined for shape recovery. A model-based approach to reconstruction uses a-priori knowledge of scene structure and can overcome visual ambiguities that can make visual scene reconstruction an ill-posed problem for model-free techniques. Reconstruction of human models from multiple views presented in Chapters , , is based on a humanoid model and makes use of this prior geometric information to constrain the feasible shape of a person in reconstruction.

Optimisation of a surface model to match image data was first introduced in computer vision and computer graphics by Terzopoulos et al. [166]. Terzopoulos et al. [164] introduced a class of physically based models that describe the shape and motion of deformable curves and surfaces. Terzopoulos et al. [166] demonstrated that these deformable models can be applied to reconstruct the shape and motion of flexible objects from images. A deformable model is formulated as a continuous elastic surface that deforms dynamically to satisfy a set of imposed shape constraints. The advantage of this representation is that it provides the means to match multiple constraints derived from different views and interpolate the constraints with a continuous surface. The deformable model approach provides a robust method of shape recovery that can link sparse or noisy data and span missing sections of data to estimate a continuous, smooth surface approximation [115].

Model-based scene reconstruction with a deformable model is motivated by a physical approach where a model is composed of a simulated elastic material that deforms to satisfy a set of constraints. Model deformation is defined as an energy minimisation task with an external energy that applies the constraints on model shape and an internal energy that penalises the elastic deviation in the model surface to regularise deformation. Terzopoulos [161] notes that the inverse problem of 3D reconstruction from 2D observations can be ill-posed. That is the shape constraints do not guarantee that a solution will exist, or that it is necessarily unique. Regularisation methods [170] provide a systematic approach to reformulate ill-posed problems in a well-defined framework that can be solved. The deformable model approach can ensure that a unique solution exists for the reconstruction problem at the global minimum of the energy function for the model [161]. The drawback for the approach lies in defining the

degree of regularisation required to define the trade-off between fidelity in fitting the constraints and penalising the model deformation [169].

There has been significant research on deformable models, Montagnat et al. [121] review the different representations of deformable models that have been proposed and McInerney and Terzopoulos [115] provide a review on the application in the domain of medical image analysis. Deformable contour models, known as "snakes" [88], have been widely used for 2D image segmentation. Deformable 3D surface models have been applied for segmentation and shape recovery from 3D range data [50] and 3D medical imaging data-sets [122]. A variety of shape representations have been explored to achieve this, including surface meshes [107], particle systems [159], superquadrics [163] and implicit representations [179]. Surface evolution has been realised through discretisation of the deformation energy functions using finite differences [107] and finite elements [37] or through level-set evolution of an implicit surface [109]. Constrained surface deformation has been imposed through the use of local shape constraints [120], restricted global transformations [122], free-form deformations (FFDs) [18] and reduced shape parameters [40]. Adaptive models have also been proposed to allow a surface to adapt and provide a high degree of freedom in data-fitting [94].

Deformable models have the advantage that geometric information on the scene can be imposed with the prior model. This provides a robust approach that can allow reconstruction in the presence of visual ambiguities where model-free and object-centred techniques will fail. However, the prior model must provide a sufficiently close approximation such that the true shape of the target scene can be reconstructed within the feasible space of the model. The model must also be aligned with the target scene and the degree of regularisation controlling the feasible space of shapes must be predefined.

2.1.2 Image based rendering

Geometric scene reconstruction addresses the problem of deriving a 3D model for a scene which can then be combined with a texture map to give a visually realistic appearance. Image-based rendering on the other-hand performs image synthesis directly from the original camera images without explicitly deriving geometric information.

This approach can provide the complex view-dependent appearance that is captured in camera images and avoids the ill-posed nature of visual scene reconstruction.

Image-based rendering interpolates the plenoptic function, a function that gives the light intensity and colour at any point in space from any viewing direction. McMillan and Bishop [116] define the problem of image-based rendering as:

Given a set of discrete samples (complete or incomplete) from the plenoptic function, the goal of image-based rendering is to generate a continuous representation of that function.

A photograph provides a discrete sampling of the plenoptic function and if there are sufficient photographs to densely sample the complete function then new images can be synthesised directly by resampling the original images. If image positions are further apart then the correspondence between image pixels at each captured position is required in order to construct a continuous function from the samples. In this section techniques for direct interpolation from dense samples and the use of image correspondence for interpolation are reviewed.

2.1.2.1 Interpolation from dense samples

The light in a scene can be described by the plenoptic function, first described by Adelson and Bergen [15]. The plenoptic function is a seven-dimensional (7D) function that gives the radiant energy that is perceived from the point of view of an observer. For a particular instant in time t, at a given 3D position in space (x, y, z), looking in a 2D viewing direction described by an azimuth and elevation angle (θ, ϕ), the eye will be sensitive to light at different wavelengths λ. In practise a fixed instant in time is considered and the light intensity and colour is sampled by integrating across all visible wavelengths giving a five-dimensional (5D) function (x, y, z, θ, ϕ). The 5D plenoptic function or light-field [100] completely describes a visible scene in terms of observer position and viewing direction.

A discrete camera image provides a set of samples of the light-field in which each pixel samples a particular viewing direction from the centre of projection for the camera.

If images could be acquired for every point in space, covering all viewing directions, then view synthesis would be a simple matter of looking up the light ray corresponding to each pixel in the virtual camera in the original set of images. The first example of this approach to view synthesis is demonstrated by the movie-map system introduced by Lippman [104]. In the movie-map, thousands of images were stored in a database and the system retrieved images according to the closest view-point to that of the user. This system is effectively a nearest-neighbour interpolation of the sample images. Such a system is unfeasible in practise due to the vast storage required for every conceivable view.

Environment maps reduce the dimensionality of the problem and the storage requirement by restricting the view-point to a fixed location in space. Environment maps record the incident light arriving at a single point and were first used to efficiently approximate the reflections of the environment on the surface of computer graphics models [71]. Chen [30] demonstrated that they can also be used to synthesise a view of the environment from a fixed position in the QuickTime VR system. QuickTime VR stitches together a set of images taken from a fixed location to form a higher resolution cylindrical image, a process termed image mosaicing [147]. This system allows for an interactive visualisation of the environment with control of viewing direction and field of view, with the restriction of a fixed viewing position. Shum and Szeliski [147] give a detailed description of different techniques to construct panoramic image mosaics from a sequence of camera images.

The dimensionality of the plenoptic function can also be reduced using the observation that in free space the light-field reduces to a 4D function across a convex surface that encloses the observed scene. An image can be synthesised from any point outside the surface by determining the light ray at the surface that corresponds to each pixel in the virtual view. Two approaches to the 4D plenoptic function were introduced simultaneously, Light-Field Rendering [100] and the Lumigraph [68]. Levoy and Hanrahan [100] capture sample images across a surface for Light-Field Rendering using a camera mounted on a motion control platform and Gortler et al. [68] construct regular samples for the Lumigraph from images captured at general camera positions by making use of estimated scene geometry. Both techniques parameterise light rays according to 2D

sample position on the surface and the direction by a 2D intersection with a parallel surface plane. New images are synthesised using a projective mapping of the viewing plane onto the two parallel surfaces. The 2D location on each surface plane can then be derived for the ray corresponding to each pixel of the virtual view and the captured samples can be interpolated.

Environment maps and light-field rendering or the Lumigraph are able to produce photo-realistic virtual views of a scene, making use of a large number of sample images to remove the requirement for any geometric information. This enables image synthesis for complex scene geometry such as hair or clothing where multiple view reconstruction can fail. Environment maps provide a compact representation of the images, however the view-point is restricted. Light-field rendering and the Lumigraph allow a change of view-point, however the scene must be over-sampled to avoid aliasing effects when interpolating the sample images. There is significant redundancy in the representation, and a large amount of data must be acquired and stored.

2.1.2.2 View interpolation and reprojection

There is a trade-off between the number of input images required for view synthesis by direct interpolation of images, and the amount of geometric information available for a scene allowing for synthesis with fewer input images. Where image sampling is insufficient for direct interpolation, geometry can be extracted from photographs either explicitly or implicitly in order to synthesise new images. Figure 2.4 illustrates the re-projection of a camera image using depth data in comparison with image interpolation.

Techniques have been proposed to interpolate between views by morphing the input images based on depth information or correspondences between multiple images. Image morphing is an established image-based technique in computer graphics where the goal is to generate transitions between reference images [181]. The morphing process involves changing the shape of one image to match the shape in another based on specified correspondences between the images. This does not however guarantee that the in-between images represent valid novel views for a projective camera in the original scene.

(a) Left image (b) Right image (c) Image interpolation (d) Image reprojection

Figure 2.4: Comparison of image interpolation without image correspondence and view reprojection using dense depth data derived from stereo correspondence as shown in Figure 2.1.

Chen and Williams [31] presented a method to interpolate between views based on image morphing with dense depth information. The depth information is used to construct dense pixel correspondences between two views so that intermediate views can be synthesised by linear interpolation between matched pixels. This linear interpolation provides valid intermediate views in the specific case where the camera moves perpendicular to the viewing direction. Seitz and Dyer [142] extended view-interpolation to non-parallel camera views by first reprojecting the reference images to parallel viewpoints where linear interpolation can be used. Seitz and Dyer avoided the need for explicit depth information using image morphing to linearly interpolate between matched image features. Depth information is still required however to avoid artifacts in the morphed images arising from occlusions [142].

Where depth is available for every point in an image, the image can be reprojected to a novel view-point allowing for the synthesis of a wide-range of views compared to view interpolation. Chen and Williams [31] used synthetic environments to generate exact depth values. The problem of recovering depth information in real images for reprojection has been addressed through stereo image correspondence [96], and using approximate geometry from silhouettes [113]. Laveau and Faugeras [96] describe a

backwards mapping approach to constructing a novel view from a disparity map without the need for a geometric description. Matusik et al. [113] present an image-based approach to synthesise a virtual view based on the depth to the visual-hull, without the need to explicitly reconstruct the hull from silhouette image data. Depth information for an image can only represent the closest object along the ray for each pixel and surfaces that are not visible cannot be produced in reprojecting to a new viewpoint. Layered-depth images [144] have been proposed to over come this, where the colour and depth from multiple images are combined into a single image representation with pixels at multiple depths. This has been used to synthesise novel views without occlusion.

The advantage of view interpolation and reprojection lies in the relatively small number of sample images required compared to plenoptic modelling. All techniques however require the correspondence between views, either in the form of a dense pixel depth or image feature correspondences. The techniques are therefore restricted by the acquisition of this data in the same way as techniques for geometric reconstruction from images. The set of feasible view-points that can be synthesised is also constrained to surround the sample views without the appearance of artifacts due to incorrect depth estimates and holes from regions that are not visible in the original images.

2.1.3 Hybrid representations

There are a number of different representations for image-based modelling and rendering in the literature spanning the range from geometric reconstruction through to image-based rendering. On the one end, geometric reconstruction of a texture-mapped 3D model allows rendering and manipulation in a conventional graphics pipeline. Passive reconstruction from images is however a difficult task and a texture-map provides only a fixed view-independent appearance. On the other end, image-based techniques can simulate the complex view-dependent appearance observed in images. However, image-based rendering has significant storage requirements, provides limited view synthesis and no control for editing of the scene content. In between these approaches there are a number of methods that vary according to the degree to which they use either geometric or image-based information. In this section hybrid representations

are considered to combine the view-dependent appearance captured in different camera images with a 3D model to provide a visually realistic view-dependent appearance for a geometric model.

2.1.3.1 Surface reflectance

The light-field captured in a dense set of camera images has been combined with the geometry of a scene as a surface light-field to give a complete description of the view-dependent appearance of a scene model. The surface light-field defines the set of rays reflected from a scene, the light-field, at the point of origin on the surface of a scene. Wood et al. [182] present a method to capture this representation for real-world objects. In the 3D photography system described by Wood et al. [182] the geometry of an object is first reconstructed using active range scanning. The light-field is then captured using photographs surrounding the object and resampled on the surface to define the colour at every point for every viewing direction. Chen et al. [32] show how this light-field representation can be compressed on the geometry and efficiently rendered with graphics hardware.

The surface light-field provides the view-dependent appearance of a model for the fixed illumination conditions for which the camera images captured the light-field. To synthesise a visually realistic novel view under differing illumination conditions the reflectance properties of the surface must also be measured. The reflectance of an opaque surface is described by the bi-directional reflectance distribution function (BRDF). The BRDF measures the ratio of the radiance exiting a surface in a given direction (θ_1, ϕ_1) to the incident irradiance from a set direction (θ_2, ϕ_2) for a particular wave-length of light λ. In practise the RGB reflected colour observed in an image is considered and the BRDF is expressed as a four-dimensional (4D) vector value function in $(\theta_1, \phi_1, \theta_2, \phi_2)$ giving the ratio of reflected light in each of the RGB colour channels [110].

In computer graphics, physically accurate BRDF models have been developed for realistic scene synthesis [24, 39, 76]. These analytical models are based on physically inspired material parameters that define observed effects such as specular highlights. Appropriate model parameters have been derived from real-world surfaces by sampling

the BRDF with a gonio-reflectometer [110, 44], and fitting a selected analytical model to the data [178, 183]. This approach provides a compact representation of surface reflectance. The drawback is that a parametric model is only an approximation and does not necessarily reproduce the complex reflectance properties of real surfaces. Parametric BRDF models cannot represent the complex reflectance functions of materials such as hair or cloth for example [114].

The reflectance of real-world surfaces can be reproduced using samples of the BRDF directly in rendering. This approach preserves the detail that is lost in fitting an analytical model to the data, although it will incorporate any noise or errors in the sampling process. Debevec et al. [46] described a technique to render the complex reflectance characteristics of the human face using the sampled BRDF. Debevec et al. [46] describe a *light-stage* in which a camera records the reflected light in a fixed viewing direction under 2000 different illumination conditions provided by a moving light source. The images sample the reflectance-field in the *light-stage*, the light-field observed under the different illumination conditions. Defined on the surface of a geometric model this provides the surface reflectance-field. A novel view can be synthesised for a desired lighting environment by resampling the reflectance-field in the same way as the light-field is resampled in image-based rendering.

Techniques have been proposed to measure both geometry and reflectance simultaneously. Sato et al. [137] describe a system using an active range scanner to capture geometry together with a camera and light-source to sample the surface BRDF. Sato et al. [137] fit a parametric reflectance model allowing the estimation of the BRDF from 120 colour samples obtained as an object is rotated in front of a fixed camera with a fixed light source. The technique however requires accurate geometric data from active reconstruction to define the surface position, orientation and correspondence in the sequence of camera images. Matusik et al. [114] describe a system that makes use of the approximate geometry in the visual-hull. Matusik et al. [114] capture the surface reflectance field using a rotating set of cameras and lights to give 53136 different images defining the reflectance function of every surface voxel on the visual-hull. The inexact geometry in the visual-hull is compensated by dense sampling of the reflectance-field, at the expense of capturing and storing a large number of sample images.

The surface light-field and surface reflectance-field or reflectance modelling combine the advantages of geometry-based and image-based representations. A complete description of the complex appearance for real-world scenes is obtained for a geometric scene model. This also has the disadvantages of both techniques, requiring in some cases accurate scene reconstruction and other cases dense sampling of the light-field or reflectance-field. All techniques also require a specialist capture environment and a fixed, rigid geometry. Capturing the surface light-field, reflectance-field or estimating a parametric BRDF model for non-rigid objects observed from a sparse set of views in a free environment with an approximate geometric model remains an open and challenging research problem.

2.1.3.2 View-dependent rendering

View-dependent rendering attempts to combine a geometric and image-based representations for a scene to over come the disadvantages of both techniques. This is based on the observation that geometric information can be used to interpolate a sparse set of images in a similar way to view interpolation and reprojection. The set of images can then provide a view-dependent model appearance that can compensate for insufficient or inaccurate detail in the model geometry and provide view-dependent appearance as illustrated in Figure 2.5.

Debevec et al. [49] introduced view-dependent texture mapping, an approach that provides surface light-field rendering with only approximate geometry and a limited set of sample image. Debevec et al. [49] presented the Facade system for modelling and rendering architectural scenes from a set of photographs. The geometry of the model is first constructed from geometric primitives such as boxes or arches, created interactively by user selection of point correspondences in multiple images. The model is then textured from the original images and rendered to a new viewpoint. The texture from different photographs is blended in rendering according the position of the virtual viewpoint, favouring the photographs taken from the closest viewing directions. This view-dependent texturing reproduces the complex view-dependent appearance effects captured in the original photographs and demonstrates realistic results even with simple

(a) Left image (b) Right image (c) Single texture (d) View-dependent texture

Figure 2.5: Comparison of view-dependent texturing with a static texture map on a reconstructed scene model with inaccurate geometry at the face.

geometric models.

View dependent rendering provides a trade-off between model-based and image-based representations. This can provide realistic view synthesis with only approximate scene geometry and images captured from a limited set of viewpoints. Pulli et al. [134] describe a technique to render a virtual view with view-dependent geometry and texture from multiple 2.5D depth-maps captured using a stereo camera system. Matusik et al. [112] describe a real-time system to render dynamic scenes using the visual-hull with view-dependent texture. The drawback of this approach to scene representation is that it relies on the geometry to interpolate the appearance in the images, it is therefore dependent on the trade-off between the accuracy of the image correspondence given by the geometric representation and the number of images sampling the appearance of the scene.

View-dependent rendering provides a hybrid geometry-based and image-based representation for a scene. This approach can allow for approximate scene geometry, requires significantly less storage requirements than surface light-fields or reflectance fields and provides a geometric model that can be rendered and manipulated in a standard graphics pipeline.

2.2 Human modelling from images

Computer generated virtual humans were first developed in the 1970's to test the ergonomics of design prior to manufacture in the automotive and aeroplane industries [167]. Simple models were used, consisting of an articulated skeleton to define human pose, with the body represented by volumetric primitives such as cylinders or boxes [52]. Highly realistic geometric models of people have now been developed in computer graphics for use in computer games, films and advertising. The art and technology for production of these human models form well established industries [38]. Human modelling from images is inspired by the need to provide such models with a "photo-realistic" appearance obtained from images of real people. In this section a brief overview of human modelling in computer graphics is given. The use of image-based modelling and rendering for realistic model capture is then presented from the literature.

2.2.1 Human models in computer graphics

Three-dimensional character models used in computer graphics in general follow the multi-layered model approach introduced by Chadwick et al. [29]. The multi-layered model consists of a skeleton structure with successive layers of detail representing the body tissue, surface shape and external detail such as clothing and hair. The skeleton provides an animator with high-level motion control and each layer of the model is then successively deformed with the skeleton to produce natural looking surface deformations. The body tissue for example produces the movement of the body surface which in turn influences the movement of clothing and hair. A variety of geometric modelling methods have been proposed to represent geometry, including the polygonal mesh [20], parametric surfaces [146] and implicit surfaces [168]. Deformation is achieved either by directly attaching the geometry to the skeleton, a technique termed Skeletal-Subspace Deformation [108], embedding geometry in a Free-Form Deformation (FFD) lattice [139], and through interpolation or morphing from example shapes based on skeletal pose [103, 90].

The traditional approach to model construction is through commercial computer aided

design packages such as Maya [2], 3D Studio Max [6], or Lightwave 3D [9] and SoftImage XSI [12]. There are two stages to the modelling process, geometric construction and creation of the model skeletal animation structure. An artist first creates the shape of the body, called the skin, and defines the surface properties such as colour, texture and light reflectance to give the model appearance. A skeleton structure is then constructed and the skin is attached to the skeleton so that the body deforms as the skeleton is animated. Manual construction of the complex geometry and appearance of the human body is difficult and simplified cartoon-like characters are often used with smooth surfaces and uniform regions of appearance. Realistic animation is as important as appearance and the 3D structure and placement of the skeleton together with the attachment of the skin is critical. This forms a lengthy manual process in which the surface deformation must be tested as the model is animated. The whole modelling process requires considerable time and skill, and teams of designers and animators are required to achieve realistic human models.

Active range scanning techniques are currently used to simplify the modelling process. Computer graphics models often start as clay "maquettes" that are digitised using range scanners. Clay provides an intuitive medium to construct a model and an artist can sculpt complex, detailed shape down to every pore on the skin surface. A range scanner then captures the shape of the maquette as a dense set of 3D points which can be imported into a modelling package. Significant post-processing is then required to construct a high-resolution surface from the points, together with the surface properties and animation structure needed to render and control the model.

2.2.2 Human model reconstruction from images

Human models are used extensively in computer vision for the problem of human motion analysis. The visual analysis of human motion forms a major area of research which addresses the detection, tracking and recognition of different people or actions in images. Human models are used to impose constraints on the geometry or motion of a person in a set of images in order to estimate human pose. These models are generally specified a-priori and use simplified geometric representations such as a stick

figure, 2D contour or 3D volumetric primitives that are unsuitable for representing the complex appearance of a person for computer graphics applications. There is extensive literature on human motion tracking in computer vision and surveys have been presented by Aggarwal and Cai [16], Gavrilla [66], Moeslund and Granum [117], and Wang et al. [177].

The reconstruction of detailed geometric and appearance models of people has been addressed in both computer vision and computer graphics. Techniques for visual scene reconstruction have been applied to reconstruct models from images. The different methods presented can be broadly divided into techniques for modelling the human face or the whole-body.

2.2.2.1 Face model reconstruction

The synthesis of realistic images of the human face has drawn considerable interest in computer graphics for facial character animation. The first computer generated images of the face were generated by Parke [127] who recovered crude polygonal face models by hand marking features in photographs and produced animated sequences by interpolating between the models for different expressions. Parke [128] went on to produce the first parameterised face model for animation and inspired models based on the skin and muscles of the face for physically-based facial animation [165, 99].

Individual face models have been constructed from features located in a number of photographs. Kurihara and Kiyoshi [91] manually labelled a limited set of features in multiple views and deformed a generic head model to match the reconstructed 3D feature locations. Akimoto et al. [17] used the silhouette template and automatic feature location in two orthogonal images to update a generic model, making use of the orthogonal views to generate 3D coordinates without explicit reconstruction. Lee and Magnenat-Thalmann [97] adopt a similar approach, using feature lines in two orthogonal images to update a generic model. These techniques take a prior generic model that has been parameterised for animation to produce an individual head model with an animation structure. A similar approach to model recovery has been used to capture a single model in different expressions to produce animation through shape

interpolation. Pighin et al. [129] used manually labelled features in multiple views to recover both camera pose and 3D feature locations to update a generic model to match a face with different expressions. Guenter et al. [73] placed markers on an actor's face to automatically locate features and conform a generic model to an animated sequence captured in multiple views.

Active range scanners have been used to produce high resolution geometric detail for face models. Lee et al. [99] matched a cylindrical projection of a generic head model to a cylindrical depth-map from a range scanner to update the geometry of the model. Range scanned clay maquettes are used in the film industry to produce detailed head models with different expressions for 3D facial character animation through shape interpolation. Dense 3D geometry of the face has also been captured from images using passive stereo reconstruction. Fua et al. [62] describe a technique to match a generic head model to multiple stereo depth-maps and a method [60] to recover both camera pose plus an individual head model from multiple views captured with a single camera. Stereo vision provides noisy 3D data and the reconstructed models do not demonstrate any improvement over the models generated from sparse image features. Active light projection has been used to improve 3D reconstruction in stereo and produce detailed geometric head models [55].

All of the approaches to face modelling described are based on conforming a generic polygonal head model to match the shape of an individual and recover the appearance from images as a texture map for the model. This has enabled the reconstruction of visually realistic models from a limited number of images. Blanz and Vetter [23] demonstrate 3D model generation from just a single photograph based on a database of models generated from an active range scanner. These techniques can be classified as model-based approaches to reconstruction. This follows the *functional modeling* paradigm proposed by Terzopoulos [162] in which a generic model is modified and retains the original structure, the mesh topolgy and animation parameters. This provides either a consistent set of models for shape interpolation or a model with a predefined parameterisation for animation.

2.2.2.2 Whole-body model reconstruction

Commercial active range scanners have been used to capture the geometry of the human body in a single static pose. These systems are currently used for applications such as anthropometric surveys, clothing design and medical research [28]. Ju and Siebert [82] proposed a method to match a generic human model with a control skeleton to a range scan and produce an individual animated model. Starck et al. [149] demonstrate the use of a generic model to recover the gross shape and fine geometric detail from a range scan to produce detailed animated models. Seo and Magnenat-Thalmann [143] match a generic model to a set of range scans to create a parameterised model for the synthesis of novel animated human models.

The shape and appearance of people has been captured from images in multiple camera studios. In the "Virtualized Reality" system presented by Kanade et al. [87, 123] a 51 camera dome was used to reconstruct dynamic sequences of a person through multiple view stereo. Moezzi et al. [118, 119] used a 17 camera system to construct the shape of a person from image silhouettes. Vedula et al. [175] used 17 cameras and performed *Voxel-Coloring* to capture the shape of a moving person. Matusik et al. [112] presented a technique for real-time reconstruction and rendering of a person reconstructed from 4 camera images using the visual-hull. These techniques have enabled the synthesis of novel views of a dynamic event through view-dependent rendering with the approximate reconstructed geometry [112, 123, 175]. However, these model-free techniques are restricted to replaying the captured event. Separate geometric models are constructed at each time-frame and there is no consistent structure that can be instrumented with a skeleton for animation.

Kakadiaris and Metaxas [84, 85] present a model-free approach to reconstructing an articulated model of a person from three orthogonal cameras. The technique uses a 2D deformable model to fit the silhouette of a person in orthogonal views. The person performs a sequence of movements in order to reveal different segments of the body and generate a separate 2D model for each segment. The 2D shape for each segment is then integrated from two orthogonal views to create a 3D segment model. This approach generates a segmented model of a person that could then be animated with a control

skeleton. While this removes the need for a generic model of a person it does require a detailed, predefined set of movements in order to generate all the necessary segments for the body.

Hilton et al. [78, 79] introduced a model-based approach to generating individual models from a single set of four images. This approach is analogous to techniques for modelling heads from orthogonal images. A generic body model is matched to the 2D silhouette of a person in two orthogonal images and the 3D shape is updated by integrating the 2D shape in orthogonal views. Model texture is then generated by blending between the appearance in four orthogonal images. This technique demonstrated the reconstruction of realistic whole-body models from images with the skeleton structure to use the model for animation. Lee et al. [98] proposed a similar technique to create a body model from two orthogonal image silhouettes and also used high-resolution orthogonal images of the face to create a model with both the animation structure for the face and body. Model-based reconstruction has enabled shape recovery from just two orthogonal images. The prior shape of the model is used to interpolate the shape given by the outline contour of the image silhouettes. This approach provides an approximate shape model for a person and the model texture provides a visually realistic appearance. The limitation of this approach is that the approximate shape can lead to a mis-match in the texture derived from different images which can have a large impact on the subjective realism of the models.

Stereo reconstruction has been used to derive dense geometric detail to generate body models. Wingbermuhle et al., [180] describe a simplified polygonal model of the human upper-body that is inflated to fit 3D stereo data. Fua et al. [131] make use of a parameterised body model to recover both the gross-shape of the upper body and motion from stereo sequences. This technique makes use of a generic model made of implicit surface primitives termed “metaballs” to represent the gross-anatomy of the body [168]. This model parameterises the shape of the body according to a reduced set of size parameters for the metaballs, giving robust shape recovery from noisy data. The technique has not been applied to whole-body shape recovery and cannot represent the shape of people wearing clothing.

2.3 Summary

The reconstruction of human models from images provides the potential for highly realistic representations of people. Currently the construction of realistic models is a manual process requiring a large amount of time and skill. Active range scanning technology has been applied to automatically acquire highly accurate geometric data of people [99, 28]. This still requires manual intervention to manipulate the data for animation and range scanners can only capture a single static pose of a person [149]. The use of multiple cameras on the other hand provides the potential for rapid creation of human models in a variety of poses with the visual realism of conventional video images. Current techniques for image-based modelling of people concentrate on model-based reconstruction with a single camera and provide only the approximate shape and appearance of a person [79, 98]. Reconstruction from multiple cameras can provide improved geometric data and the complex view-dependent appearance of a person to generate more visually realistic computer graphics models.

Multiple view reconstruction remains an active area of research in computer vision. Early work concentrated on the stereo reconstruction problem using correlation based matching between two camera images to derive 3D depth. Existing multiple-baseline stereo techniques [126] work best for scenes with large variations in the surface appearance and will fail in the presence of self occlusions between images. More recently volumetric representations have been used to derive shape from silhouette [95] or colour data [140] without correlation based matching. This can allow for surfaces with only a limited variation in appearance and large changes in visibility between views. Volumetric techniques also make no assumptions on planarity or continuity in a scene allowing for more complex structures. However, neglecting these regularising assumptions can make volumetric reconstruction susceptible to noise [26].

The shape and appearance of a person has been reconstructed from multiple views using multiple-baseline stereo [123], as well as volumetric reconstruction from image silhouettes [112] and image colour [175]. This geometric reconstruction has been combined with an image-based representation of appearance for view-dependent rendering [112, 175]. The drawback of these techniques is that the reconstructed geometry has

no structure for model animation. A model-based approach is required to provide the kinematic structure and surface parameterisation necessary to animate the captured models. In this thesis a model-based approach to multiple view scene reconstruction is developed to recover the shape and appearance of a person with the necessary structure for computer animation.

Chapter 3

Model Reconstruction from 2D Silhouette Matching

Human shape reconstruction from images has been addressed previously in the literature [131, 79, 85] and a technique has been presented for image-based reconstruction of whole-body animated models. Hilton et al. [79] introduced a model-based method to recover a whole-body model from image silhouettes captured using a single camera. Image silhouettes can be extracted from the controlled background setting in the multiple camera studio and provide a robust constraint on the shape of a person for model reconstruction. In this chapter the method is reviewed and extended to the problem of human shape reconstruction from multiple arbitrary camera views.

Section 3.1 presents the model-based method for shape from silhouette introduced by Hilton et al. [79]. The technique makes use of a 2D to 2D mapping from a generic humanoid model to an image silhouette and updates the 3D shape of a model to match the correspondence found in multiple silhouettes. The principal limitation for the general application of the technique is the strict requirement for a specific body pose and orthogonal camera views to derive the 2D mapping. In order to apply the technique we must solve the problem of establishing the 2D to 2D correspondence from a model to a silhouette in the general case of an arbitrary body pose and arbitrary camera position.

Problem Statement:

- **Given a prior humanoid model establish the 2D correspondence with an image silhouette of a person in an arbitrary pose captured from an arbitrary camera view.**

In Section 3.2 the problem of matching the 2D projection of a model to an image silhouette is addressed in the general case. The correspondence of a model in multiple views is then used to update the 3D shape of the model. This technique was presented in the paper *"Human Shape Estimation in a Multi-Camera Studio"* Starck et al. [152]. The approach is evaluated in Section 3.3 and inherent limitations identified. It should be noted that this technique treats the shape information from each image silhouette independently and in Chapter 4 an approach is introduced to match a model to multiple silhouettes simultaneously.

3.1 Shape reconstruction with a single camera

Model-based shape from silhouette introduced by Hilton et al. [79] matches a generic humanoid computer graphics model to the silhouette outline of a person in an image and retains the functional structure of the generic model for subsequent model animation. The 3D surface mesh of the model is first projected to the camera image plane to generate a silhouette shape for the model. A dense 2D to 2D correspondence is then constructed between the model silhouette and the image silhouette giving the mapping of the projected 2D vertex locations of the model mesh onto the image silhouette. This 2D to 2D mapping is finally used to deform the 3D vertex locations of the model orthogonal to the camera viewing direction so that the model reproduces the shape of the image silhouette. Hilton et al. [79] made use of four silhouettes from orthogonal viewing directions taken separately with a single camera to give three orthogonal components to update the model shape.

The process of constructing the dense 2D correspondence and updating the 3D model shape from a single image silhouette is divided into the following steps.

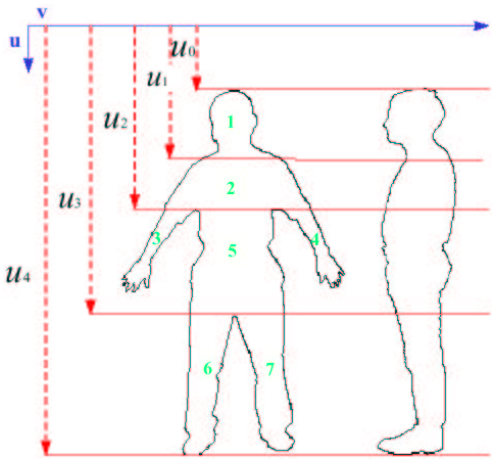
- **Body-part segmentation:** The projected silhouette for a model is first divided into separate body segments based on a known fixed body pose. Seven different segments are defined according to a set of features that can be extracted from a frontal view of a person as shown in Figure 3.1(a).

- **2D to 2D mapping:** A dense mapping is then constructed to define the correspondence for any point inside a body segment between the model and image silhouettes. Figure 3.1(b) illustrates how this 2D mapping is constructed by maintaining the ratio of lengths in horizontal and vertical image directions between the model silhouette and the image silhouette for each body part.

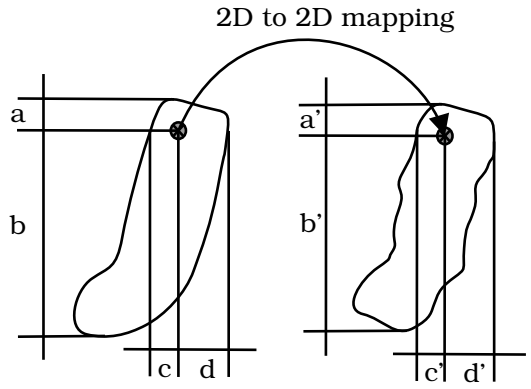
- **3D vertex deformation:** Finally the vertices of the model are updated orthogonal to the viewing direction of the camera to match the mapped 2D position in the silhouette as shown in Figure 3.1(c).

The technique for shape from silhouette [79] provides a fully automatic closed-form solution that allows the recovery of an animated human model from images taken separately with a single camera. This method for shape reconstruction has a number of important limitations: (i) A specific body pose is required such that a set of features can be reliably extracted in a frontal view of the body so that the frontal silhouette can be divided into separate body segments; (ii) Orthogonal camera views are required to provide the frontal view for body part segmentation and enable the estimation of body parts in lateral views where features are not available; (iii) The technique reconstructs only an approximate shape based on an approximate camera model and four orthogonal views; and (iv) The surface colour for the model is subsequently recovered from the approximate correspondence in each camera view leading to visual artifacts in the model appearance recovered from different views.

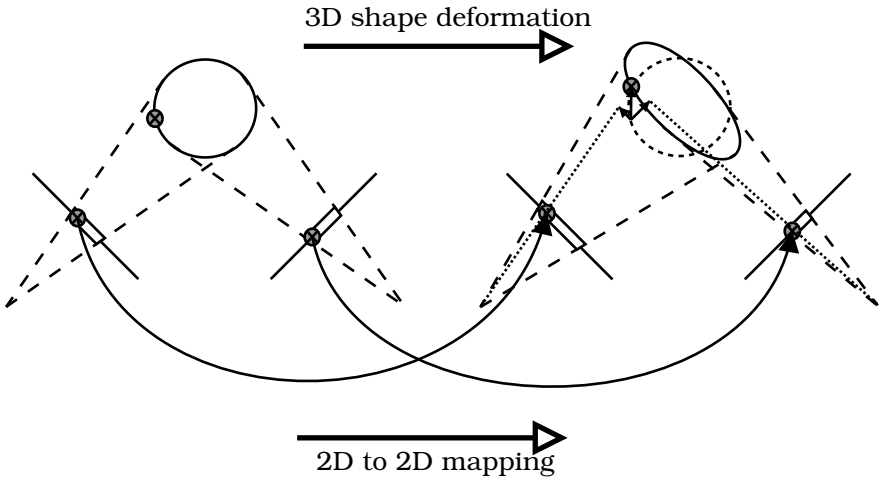
Shape reconstruction in the multiple camera studio has a number of advantages over the single camera approach. With multiple synchronous views the shape of a person can be captured without any potential movement that can occur when using a single camera. The cameras can also be fixed in the studio and calibrated in order to perform a metric reconstruction of the person's shape. Finally, many cameras can be used to



(a) Body-part segmentation



(b) 2D to 2D mapping for a body segment



(c) 3D model deformation from 2D to 2D mapping

Figure 3.1: Model-based shape from silhouette [79] based on: (a) Body-part segmentation from features on a frontal image; (b) 2D to 2D mapping from a model silhouette to image silhouette for a body segment with $\frac{a}{a+b} = \frac{a'}{a'+b'}$ and $\frac{c}{c+d} = \frac{c'}{c'+d'}$; and (c) 3D model deformation from the 2D to 2D mapping illustrated in cross-section.

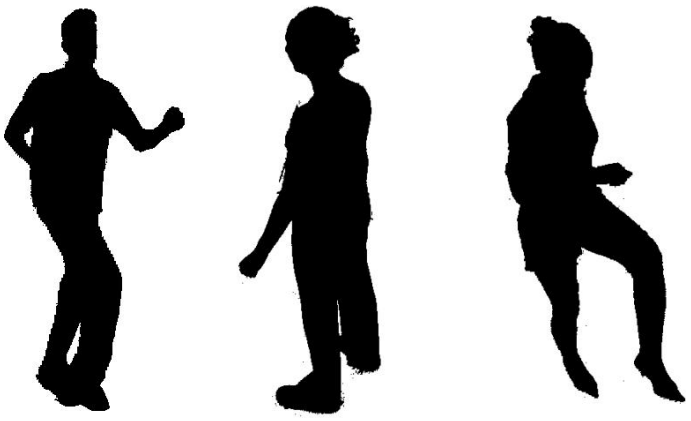


Figure 3.2: 2D image silhouettes captured in a multiple camera studio.

provide more shape information and improve the visual appearance recovered from the images. However, the technique introduced by Hilton et al. [79] cannot be applied to an arbitrary pose for a subject in a studio viewed from an arbitrary camera position. In the general case, as illustrated in Figure 3.2, it is not possible to obtain a consistent set of features to segment a silhouette into separate body parts. Complex self-occlusions arise with articulation of the limbs and different body segments are difficult to identify. Without body part segmentation it is not then possible to identify matching sections of the silhouettes between the model and captured image to construct the dense 2D to 2D mapping as illustrated in Figure 3.1(b). For an arbitrary body pose and camera position a general method is required to define the 2D to 2D mapping from a model silhouette to an image silhouette without body part segmentation.

3.2 Shape reconstruction for an arbitrary pose with arbitrary camera positions

In this section a technique is presented to define the 2D to 2D mapping from the projected silhouette of a model to a captured image silhouette in the general case where body part segmentation is not possible. Two different approaches to 2D matching are presented to define this mapping. The problem is solved using the deformable model framework introduced for shape from silhouette by Terzopoulos et al. [166] and applied

to human body part reconstruction by Kakadiaris and Metaxas [85]. The projected shape of a model is deformed to match an image silhouette in 2D with a minimum change in the shape of the model. The advantage of this approach is that it provides a general framework to update the projected shape of a model for any model pose or camera view. Two different techniques for model deformation are considered, the first makes use of geometric matches from a model to a silhouette and the second introduces a smooth shape transformation that removes the need to derive explicit matches. The correspondence in the target image silhouette is then defined by the final 2D vertex locations of the deformed model. The process of matching a model to a silhouette is applied independently for each camera view and the 2D correspondence in each view is finally used to update the 3D shape of the model.

3.2.1 A deformable model for image-based reconstruction

Shape recovery from 2D image silhouettes by model deformation was first proposed by Terzopoulos et al. [161] who introduced the concept of a *deformable-model*, an elastic model that dynamically deforms to fit a shape. The deformable model problem is posed as an energy minimisation task [161]. An energy function is constructed that consists of a potential energy in fitting the model to the target data and an internal energy that regularises the model deformation. The model is treated as a physical object that deforms according to the laws of Lagrangian dynamics to minimise the energy. The potential energy measures the deviation of the model from the data and gives rise to a set of forces that dynamically deform the model towards the data. The internal energy measures the elasticity in the model giving rise to elastic constraint forces that prevent excessive model deformation and regularise the dynamic deformation of the model.

The energy function $\mathcal{E}$ for a deformable model is constructed from a data fitting term $\mathcal{E}_{\mathcal{D}}$ and regularisation term $\lambda\mathcal{E}_{\mathcal{R}}$, where λ defines the relative weighting given to the two terms.

$$\mathcal{E} = \mathcal{E}_{\mathcal{D}} + \lambda\mathcal{E}_{\mathcal{R}} \tag{3.1}$$

The deformation of the model is then governed by the principles of Lagrangian mechanics. If the 2D surface position of the model is defined as $\underline{u}$, the mass density of the model $m(\underline{u})$ and the damping density $n(\underline{u})$ then the dynamic evolution of the model is defined as.

$$m(\underline{u})\frac{d^2\underline{u}}{dt^2} + n(\underline{u})\frac{d\underline{u}}{dt} + \nabla\mathcal{E} = 0 \tag{3.2}$$

Model evolution is simplified here by considering a zero mass system with unit damping. The model deformation then becomes a steepest descent solution to energy minimisation. The energy function is discretised at the 2D vertex locations $\underline{u}_i$ defining the shape of the model and the evolution of the vertex locations is derived.

$$\frac{d\underline{u}_i}{dt} = -\nabla\mathcal{E}_i \tag{3.3}$$

The regularisation energy constrains the deformation of the model. Here the elastic deviation of the model is used to preserve the original model shape under deformation. Every edge connecting one vertex to another is treated as a 2D elastic spring that applies a restoring force to preserve the original edge length in the model. An energy term is defined for each vertex measuring the spring energy for all the vertices edge connected to that vertex in the model mesh. For each vertex $\underline{u}_{i'}$, connected to a vertex $\underline{u}_i$, the original length of the edge is given by $l^0_{ii'} = \|\underline{u}^0_i - \underline{u}^0_{i'}\|$, where $\underline{u}^0_i$ is the original projected 2D vertex location. The spring energy is governed by a stiffness $k_{ii'}$ and the regularisation energy is then defined as.

$$\mathcal{E}_\mathcal{R} = \sum_i \frac{1}{N_{i'}} \sum_{i'=1}^{N_{i'}} k_{ii'} \left(\|\underline{u}_{i'} - \underline{u}_i\| - l^0_{ii'} \right)^2 \tag{3.4}$$

This 2D deformable model formulation is adopted for a triangulated model and two data energy terms are investigated to define the 2D to 2D transformation between the shape of the model silhouette and a target image silhouette.

3.2.2 Iterative closest point matching

The data fitting energy term for a deformable model is designed to minimise the distance to the target data. Here we wish to minimise the distance of the model to the set of pixels $\underline{p}$ forming the contour of the captured image silhouette. This requires geometric matching from the model to the silhouette contour to define a distance metric from the model to the data. A standard technique for matching termed *Iterative Closest Point* (ICP) introduced by Besl and McKay [21] is adopted. The algorithm has three steps: first for each model point the closest matching point is located, second the motion that minimises the mean squared error across all matched points is computed; and finally the motion is applied and the error recomputed. The process is iterated and converges to a local minima of the mean squared distance objective function [21].

A data fitting energy $\mathcal{E}_{\mathcal{D}}$ is constructed to match the vertices forming the projected silhouette contour of a model to match the closest pixels on the target silhouette contour. The set of contour vertices $\mathbf{C}$ on the model are found initially by rendering the model to the camera view to form the model silhouette and testing which vertices $\underline{u}_i$ form a contour pixel in the rendered image. For each contour vertex $\underline{u}_i \in \mathbf{C}$ the closest contour pixel $\underline{p}_i$ is then located and the mean-squared distance is minimised.

$$\mathcal{E}_{\mathcal{D}} = \sum_{i \in \mathbf{C}} \|\underline{p}_i - \underline{u}_i\|^2 \tag{3.5}$$

The 2D deformation of the model vertices is then given as follows with the closest matching pixels $\underline{p}_i$ iteratively updated at each step in the steepest descent energy minimisation to give an iterative closest point solution.

$$\frac{d\underline{u}_i}{dt} = \sum_{i'=1}^{N_{i'}} \frac{k_{ii'}}{N_{i'}} \left(\|\underline{u}_{i'} - \underline{u}_i\| - l^0_{ii'} \right) \times \left(\frac{(\underline{u}_{i'} - \underline{u}_i)}{\|\underline{u}_{i'} - \underline{u}_i\|} \right) + \begin{cases} \left(\underline{p}_i - \underline{u}_i\right) & \text{if } i \in C \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

This data energy term defines the deformation for the projected vertex locations $\underline{u}_i$ based on iterative closest point matching from the 2D contour of the model to the silhouette contour. The closest point heuristic for matching requires that the shape

of the model must initially be close to the target shape of the silhouette to ensure convergence to the correct shape [172]. The technique will therefore be sensitive to the initial alignment of the model and an alternative data energy term is now considered to provide a 2D mapping without the need for explicit matching.

3.2.3 2D shape morphing

The deformation of a 2D deformable model is now defined to satisfy a transformation from the 2D contour of the model to a target silhouette contour without geometric matching. A smooth 2D shape transformation is constructed between the projected silhouette for a model and the target image silhouette and the deformable model is then constrained to follow this transformation in shape.

The 2D shape morphing technique introduced by Turk and O'Brien [174] is adopted to define a shape transformation without explicit matches. The technique constructs a smooth transition between two shape contours by treating the contours as slices through a single 3D surface. The 2D contours are stacked on parallel planes in 3D and the shape is interpolated between the contour constraints using radial basis functions for scattered data interpolation. The image plane position $\underline{u}$ is augmented with a depth component to give a 3D position $\underline{U}$. A 3D implicit function $\mathcal{H}(\underline{U})$ is then defined as the summation of a set of radial basis functions $\mathcal{R}_c(\underline{U})$ centred at the constraint points $\underline{U}_c$ defining the contours. Turk and O'Brien [174] make use of the 3D thin-plate radial basis function $\mathcal{R}_c(\underline{U}) = \|\underline{U} - \underline{U}_c\|^3$ to minimise the thin-plate energy in the 3D surface and provide a globally smooth surface that interpolates the constraints. The implicit function incorporates an additional affine basis $\mathbf{A}$ to account for global terms in the shape transformation.

$$\mathcal{H}(\underline{U}) = \sum_{c=1}^{N_c} \eta_c \mathcal{R}_c(\underline{U}) + \mathbf{A} \begin{bmatrix} \underline{U}^T & 1 \end{bmatrix}^T \tag{3.7}$$

The implicit function is generated by selecting a number of constraint points N_c evenly spaced around each contour and for each point selecting an additional constraint directed along the inner normal of the contour [174]. The contours are stacked in 3D

with a depth value H given to the model contour. The implicit function is assigned a value of zero on the contour constraint and a value of one at an internal constraint. The zero-valued iso-surface of the function then forms a surface that gives a smooth continuous transformation between the projected model shape at height H and target silhouette at a zero depth. The parameters $(\eta_c, \mathbf{A})$ defining the implicit function can be derived from a linear system of equations in terms of the known values for the iso-surface defined at the constraint points, with an additional set of constraints that remove the affine contribution from the radial basis functions as follows.

$$\sum_{c=1}^{N_c} \eta_c = \underline{0} \tag{3.8}$$

$$\sum_{c=1}^{N_c} \eta_c \underline{U}_c = 0 \tag{3.9}$$

The shape transformation defined by the implicit function $\mathcal{H}(\underline{U})$ is used to provide a hard constraint on the position of the contour vertices, $\underline{u}_i \in \mathbf{C}$, for the deformable model. In the 3D image space the model is initially positioned at the height H. The distance to the target plane of the silhouette contour is then minimised by introducing a constant error in the depth direction. The model then moves through the 3D space and surface deformation is terminated when a zero depth is reached. The contour vertices are constrained during deformation to move only along the zero-valued iso-surface of $\mathcal{H}(\underline{U})$ by removing the component perpendicular to the iso-surface. The evolution of the model vertices is then defined as follows.

$$\frac{d\underline{u}_i}{dt} = \sum_{i'=1}^{N_{i'}} \frac{k_{ii'}}{N_{i'}} \left(\|\underline{u}_{i'} - \underline{u}_i\| - l^0_{ii'} \right) \times \left(\frac{(\underline{u}_{i'} - \underline{u}_i)}{\|\underline{u}_{i'} - \underline{u}_i\|} \right) + \begin{bmatrix} 0 \\ 0 \\ -\triangle H \end{bmatrix} \tag{3.10}$$

$$\frac{d\underline{u}_i}{dt} = \frac{d\underline{u}_i}{dt} - \begin{cases} \frac{(\frac{d\underline{u}_i}{dt} \cdot \nabla\mathcal{H}(\underline{U}_i))\nabla\mathcal{H}(\underline{U}_i)}{\|\nabla\mathcal{H}(\underline{U}_i)\|^2} & \text{if } i \in C \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

3.2.4 3D reconstruction

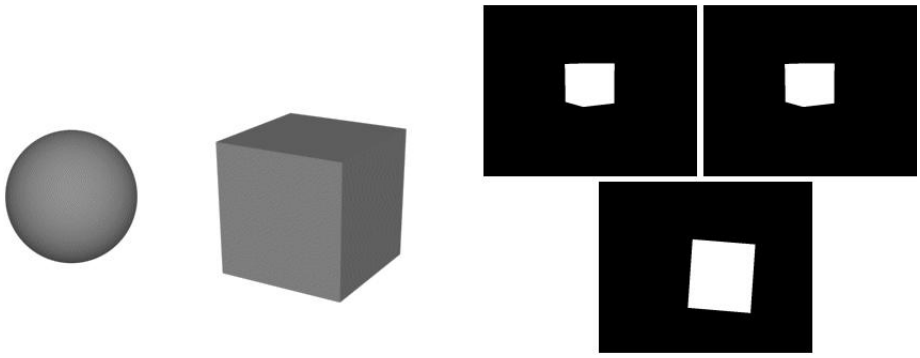
The deformable model framework matches the projected shape of a model to a target image silhouette in 2D and the final 2D vertex locations of the model provide a 2D to 2D mapping from the original projection of the model to the final image silhouette. The shape of the model defined by this mapping in multiple views is derived. For each model vertex the 3D position is reconstructed by triangulation of the 2D vertex locations.

3.3 Evaluation

The construction of a 2D to 2D mapping from the projected shape of a model to a target image silhouette with a 2D deformable model is examined in two ideal test-cases. In the first case a simplified geometric problem is considered in which a sphere is matched to the shape of a cube as shown in Figure 3.3. In the second case the technique is assessed for the problem of matching a humanoid model to the shape of a person in an arbitrary pose viewed from arbitrary camera positions created using a 3D range data-set as shown in Figure 3.4. Image silhouettes are generated for these test cases by rendering the models to the camera images with exact camera parameters. For each test presented in this evaluation a constant spring force, $k_{ii'} = 1$, is used to define the shape regularisation and equal weight is given to the data energy and regularisation energy terms, $\lambda = 1$.

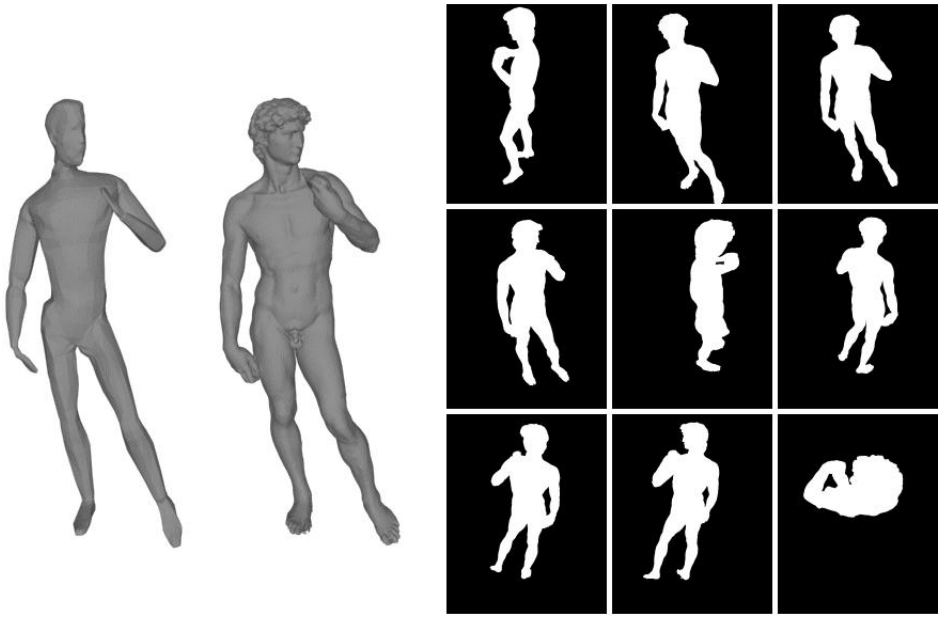
3.3.1 Closest point matching

The ICP algorithm iteratively updates the matches between the projected 2D contour vertices for a model and a target silhouette contour as the model deforms to minimise the mean-squared image plane error. This process is illustrated in Figure 3.5 which shows successive stages in deforming the projected shape of the sphere to match the cube. It can be seen from this example that where the initial matches are incorrect, the matches are updated to give a close fit between the two shapes in 2D. Closest point



(a) Sphere (b) Cube (c) Orthogonal image silhouettes

Figure 3.3: Test case: matching a sphere to the shape of a cube.



(a) Generic model (b) Range data-set (c) Camera image silhouettes

Figure 3.4: Test case: matching a generic humanoid model to the shape of a 3D range data-set courtesy of Stanford Computer Graphics Laboratory [101].

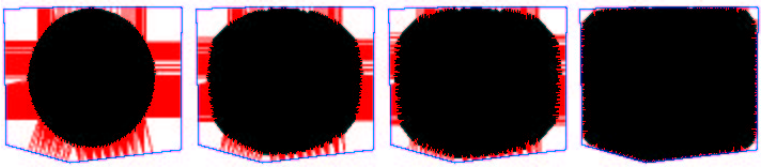


Figure 3.5: Successive stages from left to right in the 2D deformation of a sphere to match the contour of a cube with ICP matching, showing the closest point matches in red.

matching only fails in this test case at the corners of the cube which remain unmatched by the contour vertices of the sphere.

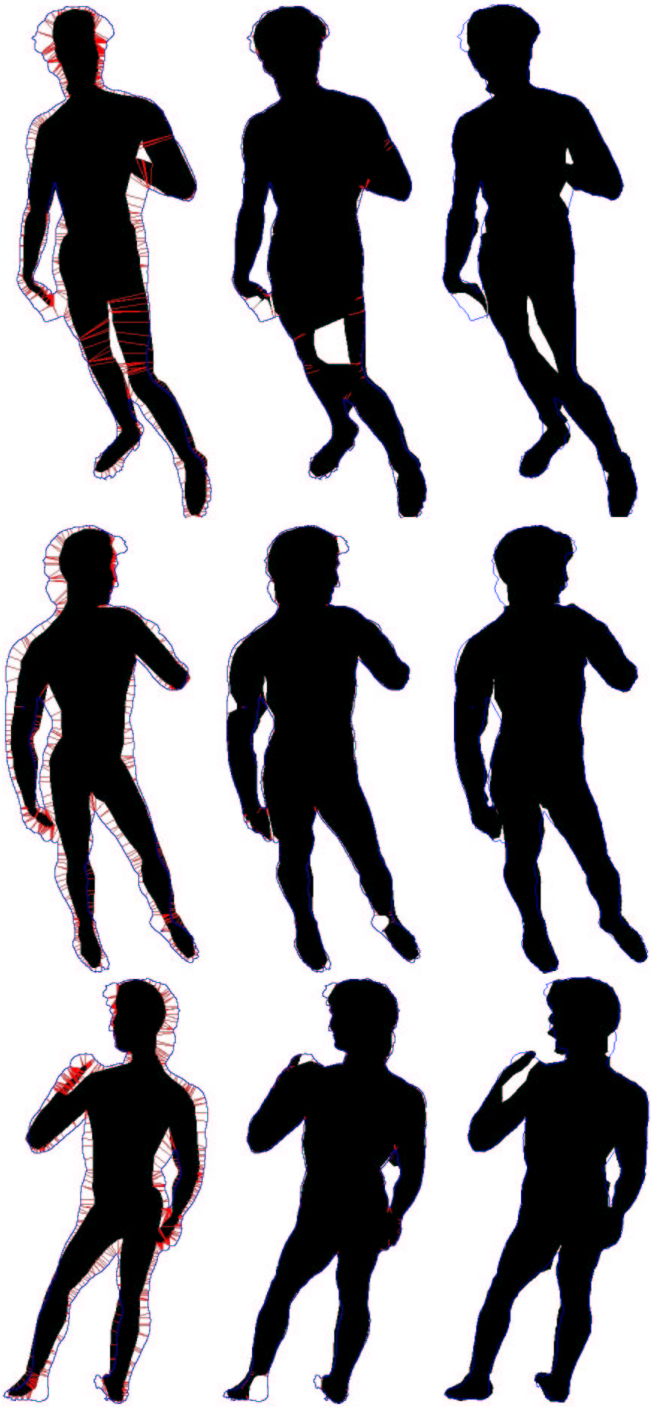
For the case of the human model the deformation with the ICP algorithm is shown in Figure 3.6(a) for 3 different camera views. There are several problems with closest point matching that are demonstrated in these views. Firstly, contour vertices on the model can be matched to the closest points that lie on the incorrect sections of the silhouette according to the initial alignment of the model with the target silhouette. Secondly, where the model is initially far from the target shape, vertices are then matched to closer points leaving some areas of the target silhouette unmatched. Finally, model vertices can be matched inappropriately where the 2D contour of the model is inconsistent with the target silhouette shape. Geometric matching with the closest point heuristic then leads to an incorrect model deformation as seen in the final shape of the model in each view, Figure 3.6(b).

Conclusion:

1. **The closest point heuristic for geometric matching can fail, leading to incorrect 2D model deformation.**

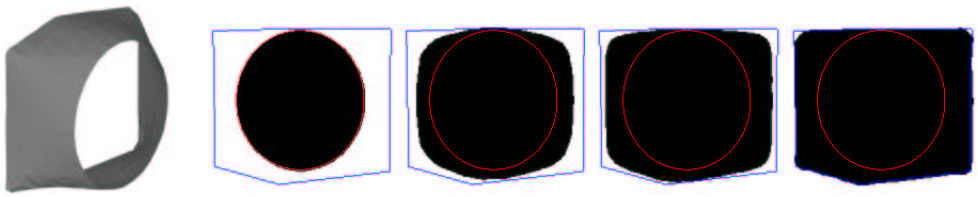
3.3.2 2D shape morphing

A shape morphing technique is introduced to remove the need for geometric matching. With the 2D shape morph the deformable model is constrained to follow a smooth continuous transformation from the model silhouette to an image silhouette. The process



(a) Initial ICP matches (b) ICP deformation (c) 2D shape morph

Figure 3.6: 2D deformation of a humanoid model to match three different image silhouettes, showing (a) the initial geometric matches for ICP, (b) the deformed shape with ICP matching, and (c) the deformed shape using the 2D shape morph.



(a) Implicit surface (b) 2D shape deformation following the implicit surface

Figure 3.7: Successive stages from left to right in the 2D deformation of a sphere to match the contour of a cube following a 2D shape morph.

is illustrated in Figure 3.7 in deforming the projected shape of the sphere to match the cube. The implicit surface shown in Figure 3.7(a) provides a transformation from the contour of the sphere to the cube and the model deforms to satisfy the target contour constraint. The final model shows a closer fit in the corners of the cube compared to ICP matching, although an exact fit is not obtained at all corners due to discrete sampling of the mesh.

There are two important parameters that control the shape of the 3D surface for the transformation between two contours, the number of contour constraint points N_c and the height H used to position the contours in 3D. In the ideal case we would take the complete set of contour pixels to define the constraints, however this would lead to a large system of equations to derive the coefficients of the implicit function. Figure 3.8 shows the transition surface obtained in one simulated view against the number of sample points N_c used on each contour. If the sampling rate is too low then the shape of the contours are not correctly represented in the final surface. In practise $N_c = 250$ is used to provide a trade-off between accuracy and the computational cost to solve the system of equations. For the height H, the contours should be positioned to give a feasible transformation between the two contour shapes. Figure 3.9 shows the different surfaces obtained against H as a proportion of the image height. Where the height is too great the intermediate surface provides a smooth shape that does not necessarily guarantee a valid transition between the two contours. As the height tends to zero the intermediate surface for transformation is lost. The height H is set here to 0.05 times the image height to assess model deformation.

(a) $N_c = 100$ (b) $N_c = 200$ (c) $N_c = 300$ (d) $N_c = 400$

Figure 3.8: Implicit surface for 2D shape transformation against the number of contour points N_c for $H = 0.05$ times image height.

(a) $H = 0.1$ (b) $H = 0.05$ (c) $H = 0.01$ (d) $H = 0.005$

Figure 3.9: Implicit surface for 2D shape transformation against height H as a fraction of the image height for $N_c = 250$.

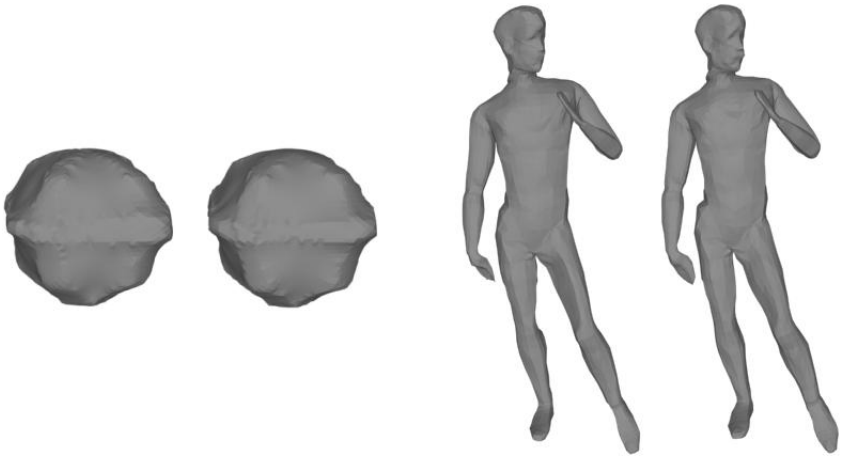
Deformation of the projected shape of a humanoid model with the 2D shape morph is shown in Figure 3.6(c) in comparison with ICP matching. The technique provides the correspondence between the model and the silhouette contour based on the similarity in the shape of the two contours. This can give an improved correspondence where a model is not correctly aligned with the target silhouette and closest point matching fails. The technique however suffers in the same way as ICP matching where the two contours are inconsistent. In the case of human reconstruction from multiple views this can easily occur due to small errors in model pose or orientation which cause the observed silhouette to have a different topology. The assumption is made that vertices forming the 2D contour for the model must lie on the target silhouette contour. As can be seen in Figure 3.6 this assumption fails where the two contours are different. With an incorrect model pose or model shape the silhouette of the model will be different to the silhouette and the vertices will not necessarily lie on the silhouette contour.

Conclusion:

2. **The assumption that the 2D model contour should match the contour of an image silhouette is not necessarily valid in the case of human modelling from multiple views.**

3.3.3 3D reconstruction from 2D mapping

The 2D to 2D mapping provided by the deformable model is performed independently in each image silhouette and this correspondence in each image is then used to reconstruct the 3D vertex locations of the final model. Figure 3.10 shows the reconstructed shape for the two test cases considered using the correspondence found in each silhouette image firstly through ICP matching then through the 2D shape morph. It is apparent that while the 2D shape of the model deforms to fit the silhouettes in each 2D view, the reconstructed shape does not reproduce the captured silhouette shape. This problem arises from treating the correspondence of the model in each image independently. The independent 2D to 2D mapping provides an inconsistent correspondence across multiple camera views, the mapped 2D vertex locations in each image silhouette do not correspond to consistent points in 3D.



(a) ICP (b) 2D morph (c) ICP (d) 2D morph

Figure 3.10: 3D model shape reconstructed from multiple views using the correspondence derived using ICP matching and the 2D shape morph.

Conclusion:

3. **Establishing multiple view correspondence by independent matching to multiple image silhouettes is invalid.**

3.4 Summary

The whole-body modelling technique introduced by Hilton et al. [79] has been considered to reconstruct a model from multiple images of a person in a studio. The technique is extended in this chapter to derive the 2D to 2D mapping from the projected shape of a model to the target shape of an image silhouette in the general case of an arbitrary body pose with camera images taken from arbitrary viewpoints. The problem is formulated as an optimisation task in which the model is deformed in 2D to match the silhouette contour either through explicit geometric matching using the ICP algorithm or through the use of a smooth 2D shape transformation. The advantage of this framework is that it can be applied to any projected model shape allowing for an arbitrary model pose in an arbitrary camera view.

Several important conclusions are drawn from this work: (i) The closest point heuristic

in ICP can fail to correctly match an observed silhouette; (ii) The 2D shape morph provides a shape transformation without geometric matching under the assumption that the model contour matches the silhouette contour; (iii) The assumption that the contour vertices for a model should match the silhouette contour is not necessarily valid with errors in model shape or pose; and (iv) Deriving the 2D correspondence independently for a set of image silhouettes results in inconsistent 2D correspondence across multiple views. This leads to the important conclusion that establishing correspondence between projections of a 3D model and an observed image silhouette in 2D is ill-posed. The problem that must be solved is to derive the 3D model that satisfies the shape constraint imposed by a set of image silhouettes while providing a consistent multiple view correspondence. In Chapter 4 model deformation is therefore considered in 3D as the task of matching a model to multiple image silhouettes simultaneously.

Chapter 4

Model-Based Reconstruction from Multiple View Silhouettes

In Chapter 3 model-based shape from silhouette [79] was presented to reconstruct an animated human model from multiple images of a person. The approach matched a generic humanoid model in 2D to image silhouettes from multiple views. The model geometry was treated independently in each 2D view leading to an inconsistent correspondence for the model across the images. In this chapter a technique is introduced to optimise a generic humanoid model in 3D to match the shape in multiple silhouettes simultaneously. The technique provides a single model that satisfies the shape imposed by the silhouettes with a consistent correspondence across multiple views.

Problem Statement:

- **Given a prior humanoid model, update the 3D shape of the model to match the image silhouettes in multiple views.**

In Section 4.1 the shape information provided by multiple view silhouettes is considered. The bounding shape given by each silhouette is integrated across multiple views by reconstructing the visual-hull, the 3D volume that reproduces the set of silhouettes. A deformable model technique is then presented to optimise the 3D shape of a generic humanoid model to match the shape of the visual-hull. In Section 4.2 a shape-constraint is introduced for a triangulated surface mesh to regularise the deformable model. The

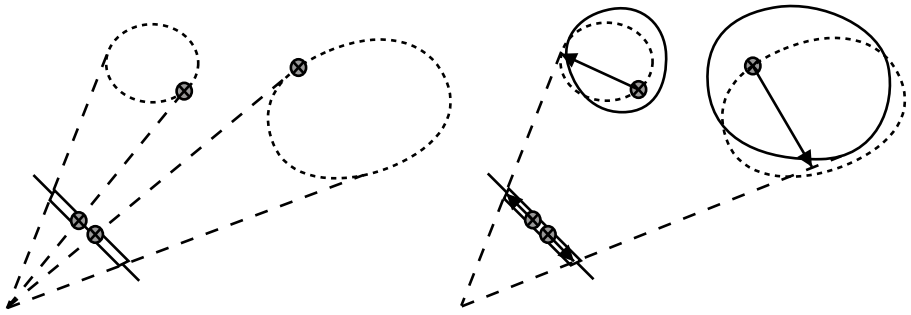
shape-constraint is formulated to preserve the prior shape of a model in fitting the approximate shape of a person given by the visual-hull. The shape-constraint is also designed to preserve the relative parameterisation of the surface with respect to the kinematic structure of the model for subsequent model animation. In Section 4.3 a data-fitting term for a deformable model is introduced to minimise the mean-squared distance from the model to the surface of the visual-hull. The data term is formulated to overcome the limitations of closest point matching in the ICP algorithm and relax the strict assumption that the surface of the model should match the approximate shape in the visual-hull.

The algorithm for the shape-constrained deformable model is outlined in Section 4.4 and evaluated in Section 4.5. This work was presented in *"Reconstruction of animated models from images using constrained deformable surfaces"* Starck et al. [153], and *"Animated Statues"* Starck et al. [149]. The assumption is made here that an approximate shape model can be derived by fitting a generic humanoid model to the visual-hull of a person in a multiple camera studio. Colour texture can then be derived from the camera images to give a visually realistic model appearance. In Chapter 5 the model-based technique is developed further to incorporate stereo data to match appearance between images for accurate texture recovery.

4.1 Shape from Silhouette

4.1.1 3D shape from 2D silhouettes

A silhouette provides the bounding shape of a person in camera image. In Chapter 3 the projected shape of a model was deformed to fit a silhouette under the assumption that the model contour should exactly match the silhouette contour. As demonstrated in Section 3.3, this assumption can fail where the projected shape of the model is inconsistent with a silhouette. With even small changes in model pose or shape there can be large differences in the projected shape and a model can have a different topology to an image silhouette. Figure 4.1 illustrates the type of problem that can arise. The figure shows a small difference in shape and position for a model in comparison with a



(a) 2D contour vertices on a model (b) 3D deformation from 2D contour matches

Figure 4.1: Illustration of incorrect 2D matches obtained where a model silhouette is inconsistent with an image silhouette leading to incorrect 3D model deformation.

target shape. In this case the model has two contour points that are not observed in the target silhouette and contour matching in 2D would lead to an incorrect 3D model deformation.

Incorrect matching in 2D can be overcome by considering additional information on self-occlusions from multiple views. The visual-hull is the 3D volume that reproduces the multiple view silhouettes. The surface of the visual-hull is used to integrate the shape from multiple silhouettes to give a single constraint on the shape of a model. The same problem is now shown in Figure 4.2 where the model is matched to the surface of the visual-hull in 3D rather than the silhouette contours in 2D. Where matching to a contour is ambiguous in 2D as shown in Figure 4.1, the contours from other views can be used to derive the correspondence in 3D based on the shape information from additional views in the visual-hull as shown in Figure 4.2. The visual-hull combines the shape in separate multiple view silhouettes to give a single constraint on the 3D shape of a person with all the information on self-occlusions that is available in the original silhouettes.

The visual-hull is reconstructed through the volume intersection of the occupied region of 3D space represented by each 2D image silhouette [95]. A silhouette describes an occluding contour that encloses the projected shape of the observed scene. Back-projection of the contour into space forms a solid cone that encloses the scene in 3D. The intersection of the solid cones from a set of image silhouettes provides the visual-

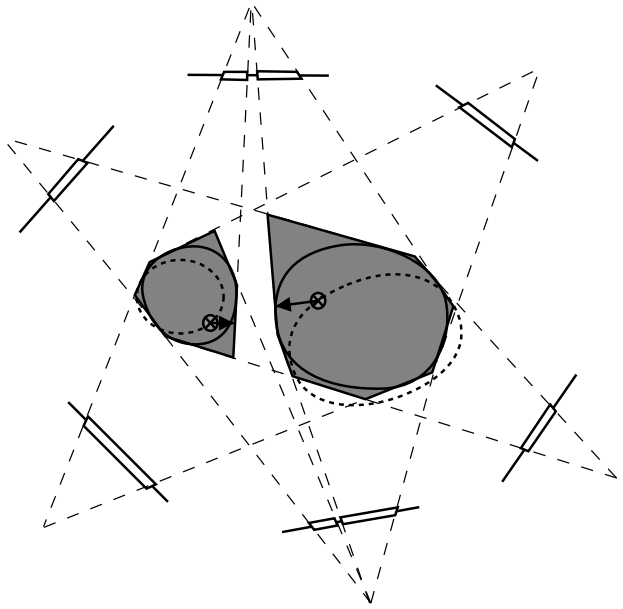


Figure 4.2: Reconstruction of the visual-hull, shown shaded, from multiple image silhouettes gives improved matching in 3D.

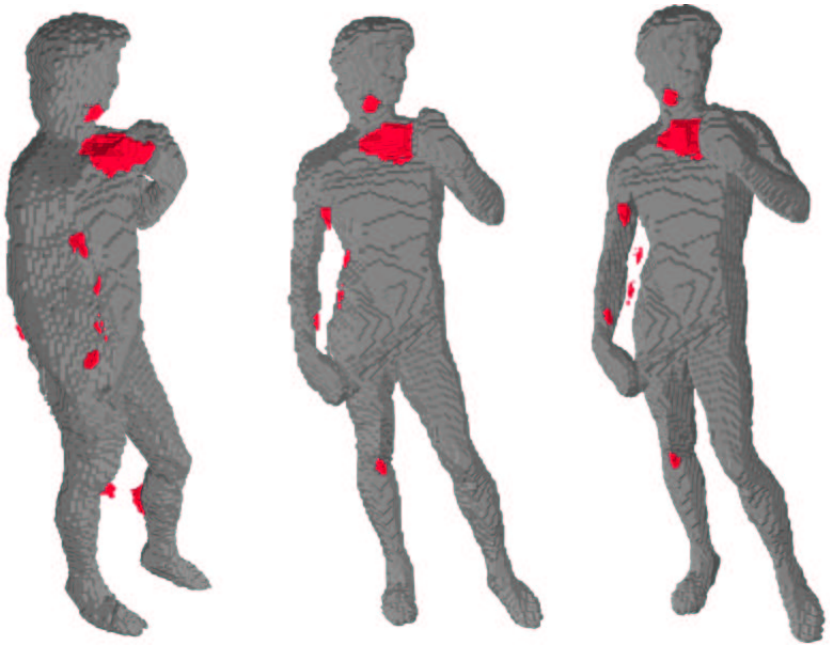


Figure 4.3: The surface of the visual-hull reconstructed at a 1cm voxel resolution from the image silhouettes shown in Figure 3.4. Protrusions and phantom volume sections are highlighted in red.

hull. In this work a volumetric approach is used for reconstruction. A set of discrete volume elements termed voxels are defined in space. The set of occupied voxels that lie inside the visual-hull are then derived by testing the overlap of each voxel with the silhouettes. If the projected shape of a voxel overlaps all the silhouettes it is set as occupied, otherwise if a voxel falls outside any silhouette the voxel is set as unoccupied. The surface voxels for a scene are extracted as the set of occupied voxels that are adjacent to unoccupied voxels. The procedure for reconstructing the set of surface points on the visual-hull is outlined in *reconstruct_visual_hull*. The image region corresponding to a voxel is simplified here as the rectangular region enclosing the projected corners of a voxel. These image regions can be pre-computed to speed up the procedure.

Input: Camera parameters, $\mathbf{P}_m$
Image silhouettes, I_m

Output: Surface points, $\underline{y}_j$

Procedure: *reconstruct_visual_hull*

```
1.  set (all voxels = occupied)
2.  for (each voxel)
3.      for (each image)
4.          project (each voxel corner to image)
5.          set (image region containing voxel corners)
6.          if (no silhouette pixels in image region)
7.              set (voxel = unoccupied)
8.  for (each voxel)
9.      if (voxel = occupied)
10.         if (any 6-connected voxel = unoccupied)
11.             set (surface voxel point y)
```

4.1.2 Model-based shape from silhouette

The visual-hull provides a 3D bound on the shape of a person given by a set of image silhouettes. Figure 4.3 shows the visual-hull reconstructed for the ideal test case considered in Chapter 3 with a person in an arbitrary pose viewed from a set of arbitrary camera positions. The surface of the visual-hull demonstrates the 3D shape information that is available from multiple view silhouette images and highlights a number of important limitations in shape from silhouette: (i) The visual-hull provides only an

approximate estimate of shape due to the discretisation of space as a set of voxels and the limited number of silhouettes used to reconstruct the volume; (ii) Concave regions of the body are self-occluded in all image silhouettes and are not represented; (iii) The visual-hull can contain protrusions or "phantom" sections of volume that are consistent with the silhouettes but do not correspond to the underlying body; and (iv) Different parts of the body may merge in the visual-hull and the surface of the hull does not necessarily represent the complete surface shape of the body.

In this chapter a model-based technique is presented to deform a generic humanoid model to match the surface of the visual-hull in order to reproduce the shape in multiple image silhouettes. The technique accounts for the approximate shape in the visual-hull and the ambiguities from self-occlusions using prior information on human shape in reconstruction. The assumption is made that the shape of the generic humanoid model represents the target shape of the person to be reconstructed. The model is deformed to match the visual-hull as a shape-constrained deformable model, preserving the prior shape in the model while fitting the visual-hull. In Section 4.2 a shape-constraint for the triangulated surface of a model is presented to form the regularisation energy term for a deformable model. In Section 4.3 a data energy term is presented for the model to minimise the distance from the surface of the model to the surface of the visual-hull while allowing for potential missing sections in the visual-hull surface.

4.2 Shape Regularisation

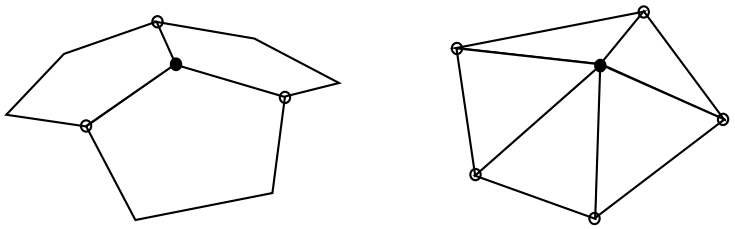
The internal energy for a deformable model serves to regularise the deformation of the model shape in optimisation. The shape of our generic humanoid model is defined by the 3D vertex locations of the triangulated surface mesh, $\underline{x}_i$. The model has a large number of degrees of freedom and can represent a wide variety of surface shapes according to the infinite number of different vertex positions that are possible. This shape variability makes model deformation sensitive to noise or outliers in the target data and potentially undesirable solutions where there is a local minimum in the deformation energy function. Terzopoulos et al. [161] formulated the surface of a deformable model as an elastic thin-plate material under tension in order to recover a model with

a minimum surface area and minimum surface curvature that fits the data. In this section a shape regularisation energy term is introduced instead to preserve the prior shape of a generic model during deformation, our prior knowledge of the target shape to be recovered from the approximate data in the visual-hull.

One important consideration in the recovery of an animated surface model is the the requirement to preserve the relative position of the vertices with respect to the kinematic structure of the model. The generic humanoid model has a predefined animation structure in which the correspondence between each vertex and an underlying skeleton is specified. The surface is then animated by manipulating the skeleton control structure. Shape regularisation must therefore preserve the mesh parameterisation, the relative position of the surface vertices, to enable animation of the recovered surface model with the predefined animation structure.

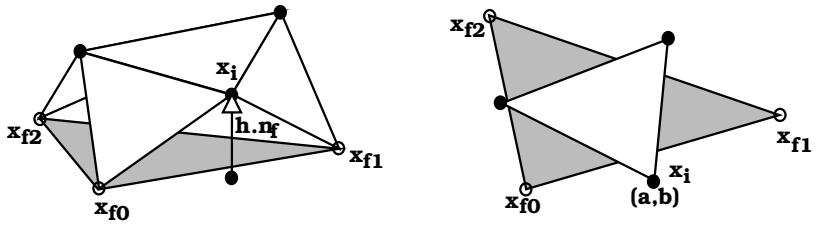
In Chapter 3 an elastic regularisation energy was constructed for a model to preserve the edge length connecting vertices in a mesh. The energy term served to spread the effect of vertex deformation across the mesh and to preserve the relative geometric relationships of the vertices in 2D. For 3D deformation this regularisation energy would cause the surface mesh of the model to act like an elastic sheet. The drawback of this formulation is that the constraint imposes only a minimum deviation in the surface area of the model and the model is free to take any shape with a similar surface area. Different approaches have been proposed to incorporate prior shape information into the deformation process to constrain the deformation space of a model. Techniques have used global rather than local model transformations [122], a restricted set of shape parameters [131], or a prior set of training shapes to describe the space of feasible models [40]. Global shape constraints or shape parameters are unfeasible here as we wish our humanoid model to take on the wide variation in body shape with subjects wearing different clothing. A local constraint is therefore considered to preserve the prior shape in a model during deformation and preserve the surface parameterisation of the model vertices.

Montagnat and Delingette [120] presented a surface constraint that preserves the local shape and position of vertices on a deformable surface mesh. The technique is based



(a) 2-simplex mesh (b) Triangular mesh

Figure 4.4: The vertex neighbourhood on a 2-simplex mesh and a triangular mesh.



(a) Height offset from triangle frame (b) Barycentric coordinates in triangle frame

Figure 4.5: Definition of a vertex position for an irregular triangular mesh in a local triangle centred frame.

on the fixed connectivity of a "simply-connected" mesh or *simplex* mesh. A 2-simplex mesh has 3 edge connections for every vertex as shown in Figure 4.4(a). With 3 points in space a local coordinate frame can be constructed to specify a relative 3D position. Each vertex on a 2-simplex mesh can then be defined in the local frame of the 3 edge-connected vertices, the 1-neighbourhood of a vertex [120]. Here we must consider the problem of defining position and location of a vertex in the general case of an irregular triangular surface mesh typically used for graphics models. With more than 3 connected vertices in the 1-neighbourhood as illustrated in Figure 4.4(b), a single coordinate frame cannot be uniquely defined at each vertex.

A triangle centred local frame is presented to specify local vertex positions on an irregular triangular mesh. Each surface triangle has 3 edge-adjacent vertices from which a local coordinate frame can be constructed. The 3 triangle vertices can then be

specified in the local frame defined by these edge-adjacent vertex positions. A vertex $\underline{x}_i$ is defined locally by a barycentric coordinate α_{if}, β_{if} and height offset h_{if} in the frame defined by the 3 vertices $(\underline{x}_{f0}, \underline{x}_{f1}, \underline{x}_{f2})$ connected to a triangle facet f as shown in Figure 4.5.

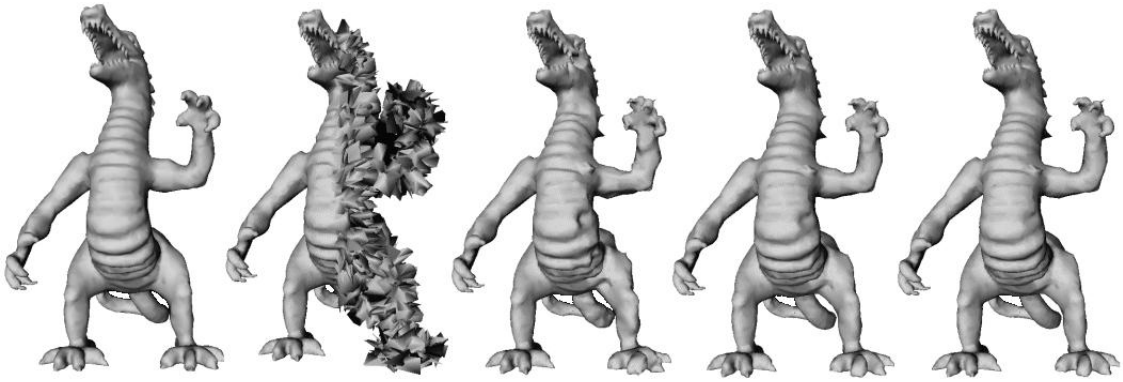
$$\underline{x}(\alpha_{if}, \beta_{if}, h_{if}) = \alpha_{if}\underline{x}_{f0} + \beta_{if}\underline{x}_{f1} + (1 - \alpha_{if} - \beta_{if})\underline{x}_{f2} + h_{if}\hat{\underline{n}}_f \tag{4.1}$$

$$\hat{\underline{n}}_f = \frac{(\underline{x}_{f1} - \underline{x}_{f0}) \otimes (\underline{x}_{f2} - \underline{x}_{f0})}{\|(\underline{x}_{f1} - \underline{x}_{f0})\|\|(\underline{x}_{f2} - \underline{x}_{f0})\|} \tag{4.2}$$

The default vertex locations representing the original shape of a model can be reconstructed using the default parameters $(\alpha^0_{if}, \beta^0_{if}, h^0_{if})$. A regularisation energy for a model is defined using an elastic constraint from the position of each vertex on a model to the default position in each triangle centred frame for a vertex. The mean-squared error to the default model shape is then minimised during optimisation. This shape constraint is scale dependent and will preserve the original model shape according to the local height offset h^0_{if} in each triangle frame.

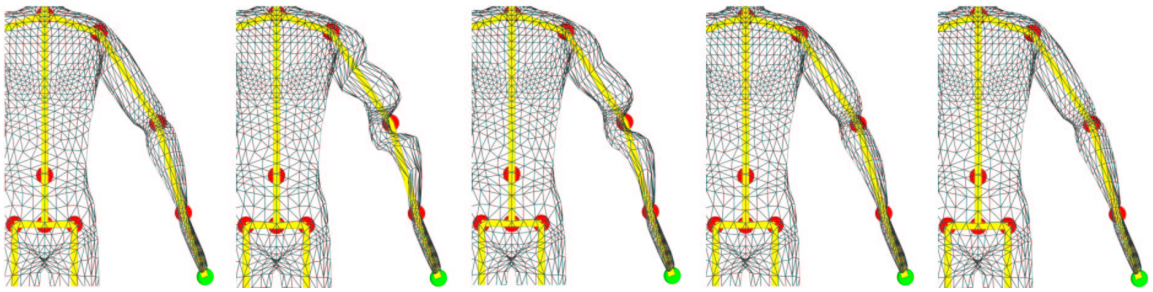
$$\mathcal{E}_{\mathcal{R}} = \sum_{i=1}^{N_i} \frac{1}{N_f} \sum_{f=1}^{N_f} \|\underline{x}_i - \underline{x}(\alpha^0_{if}, \beta^0_{if}, h^0_{if})\|^2 \tag{4.3}$$

The shape regularisation energy term minimises the deviation from the relative vertex positions in the original model, preserving the relative surface parameterisation and the local shape of the model. Figure 4.6 illustrates the deformation of a model to minimise the regularisation energy where the surface has been corrupted with random noise. This demonstrates the recovery of the model shape for a complex irregular triangular mesh with severe distortion. Figure 4.7 illustrates the deformation of the generic humanoid model to restore the default vertex locations. It is interesting to see from this that while the energy function is only defined locally the accumulated influence provides a global effect that restores the surface parameterisation.



(a) Model (b) Corrupted (c) 50 (d) 100 (e) 150

Figure 4.6: Deformation of a complex model with added surface noise at 50, 100 and 150 iterations in minimising the regularisation energy term.



(a) Generic (b) Corrupted (c) 10 (d) 100 (e) 500

Figure 4.7: Deformation of the generic humanoid model at 10, 100 and 500 iterations in minimising the regularisation energy term.

4.3 Surface Point Matching

In this section the data energy term for a deformable model is formulated to deform the shape constrained humanoid model to fit the surface shape of the visual-hull. To deform the model in 3D we must obtain the correspondence between the model vertices and the visual-hull surface. In Chapter 3 two different approaches were considered to obtain the correspondence between a model and a target silhouette in 2D. The Iterative Closest Point (ICP) algorithm [21] was used to derive geometric matches from the model to the silhouette and minimise the mean-squared distance between the two shapes. A 2D shape transformation was also presented to deform the model to match a silhouette

without explicit geometric matching. The shape transformation makes the assumption that the two shapes should match exactly and cannot be applied in matching the visual-hull where sections of the body surface may not be represented. Geometric matching is therefore performed and the assumption that the surfaces match exactly is relaxed.

The ICP algorithm solves for the geometric correspondence between two shapes based on a nearest-neighbour heuristic to assign matches. With a nearest-neighbour assignment the model can be incorrectly matched to the target shape as demonstrated in Section 3.3 and will always converge to the nearest local minimum in the least-squares distance function. A multiple point assignment technique is adopted instead to remove the nearest-neighbour heuristic in matching and increase the range of convergence for model deformation [83, 35]. A discrete representation of the visual-hull surface is used by taking the surface voxels $\underline{y}_j$ of the visual-hull. The model vertices $\underline{x}_i$ are then matched to multiple points on the surface given by $\underline{y}_j$ with an assignment weight w_{ij} that varies continuously ($0 \leq w \leq 1$), where $w = 0$ represents no match and $w = 1$ gives a one-to-one match to a surface point. A one-to-one correspondence between the two surfaces can then be obtained by imposing the constraints, $\sum_j w_{ij} = 1$ and $\sum_i w_{ij} = 1$. The assumption that a vertex exactly matches the visual-hull is relaxed by allowing $\sum_j w_{ij} \leq 1$, and outliers are accounted for in the visual-hull by allowing $\sum_i w_{ij} \leq 1$. The mean-squared distance from the model to the visual-hull surface is then minimised using the following energy function for multiple point assignment subject to these constraints.

$$\mathcal{E}_{\mathcal{D}} = \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} w_{ij} \|\underline{y}_j - \underline{x}_i\|^2 \tag{4.4}$$

$$\begin{aligned} w_{ij} &\geq 0 \\ \textstyle\sum_{i=1}^{N_i} w_{ij} &\leq 1 \\ \textstyle\sum_{j=1}^{N_j} w_{ij} &\leq 1 \end{aligned} \tag{4.5}$$

The *Robust Point Matching* algorithm presented by Rangarajan et al. [135] is used to derive the unknown set of assignment parameters w_{ij}. The advantage of this technique for matching lies in the use of deterministic annealing for optimisation as proposed

by Chui and Rangarajan [34] to overcome local minima in the assignment problem. Deterministic annealing is an optimisation method that attempts to avoid local minima in minimising non-convex energy functions. Local optimisation techniques such as gradient descent lead to local minima that depend on the initial parameters for optimisation. Global optimisation requires the location of all such local minima in order to identify a global minimum, implying an exhaustive search of the parameter space of the function. Simulated annealing is one technique used for global optimisation that avoids an exhaustive search. The technique is analogous to the thermodynamic process of annealing that enables liquids and metals to reach a global minimum energy state when cooled from a high temperature. In simulated annealing a degree of randomness is introduced to the path for energy minimisation allowing the path to overcome local minima. Deterministic annealing provides a more efficient framework to optimise functions of continuous variables by replacing the random annealing steps with a deterministic parameter update.

Deterministic annealing was first introduced as a method to solve for data-point assignment in data clustering [136]. A similar optimisation technique was presented previously by Blake and Zisserman [22], and termed *Graduated Non-Convexity*. The technique introduces a control temperature T to the energy function for optimisation and performs local optimisation at successively reduced temperatures. At a high temperature the energy function is smoothed such that the local minima lies in the region of the global minimum of the function. The local minima are then tracked as the temperature is reduced to zero, at which point the original energy function is minimised with the aim that the final local minima should coincide with global minimum. This approach to global optimisation is not necessarily guaranteed to converge to a global minimum with a finite starting temperature and a finite reduction in the temperature T. In Section 4.5 the performance of the technique with different annealing schedules is evaluated.

Deterministic annealing is applied to the assignment problem by introducing an entropy term to the data energy function with a control temperature T [34]. The entropy term measures the level of randomness in the assignment parameters [136]. At an infinite temperature T the entropy term dominates leading to a completely uniform distribution

in the assignment. As the temperature is lowered there is trade-off between the entropy and the mean-squared error function providing greater discrimination in the assignment [136]. At a zero temperature a hard association is obtained equivalent to the nearest-neighbour assignment used in ICP.

$$\mathcal{E}_{\mathcal{D}} = \sum_{i=1}^{N_i}\sum_{j=1}^{N_j} w_{ij}\|\underline{y}_j - \underline{x}_i\|^2 + T\sum_{i=1}^{N_i}\sum_{j=1}^{N_j} w_{ij}\left(\log(w_{ij}) - 1\right) \tag{4.6}$$

The assignment parameters that minimise the energy function are derived deterministically at a fixed temperature given a fixed model configuration.

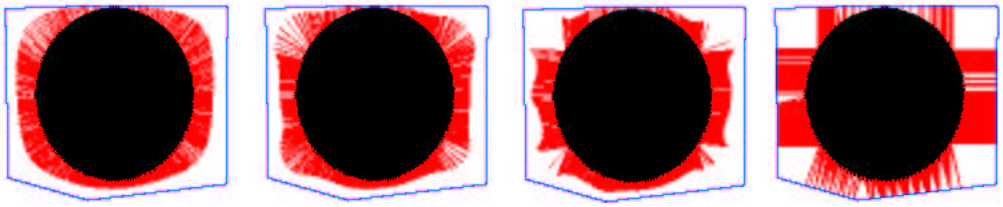
$$\frac{d\mathcal{E}}{dw_{ij}} = \|\underline{y}_j - \underline{x}_i\|^2 + T\log(w_{ij}) = 0 \tag{4.7}$$

$$w_{ij} = \exp\left(-\frac{\|\underline{y}_j - \underline{x}_i\|^2}{T}\right) \tag{4.8}$$

The assignment parameters w_{ij} must satisfy the constraints given in Equation 4.5. The inequality constraints are converted to an equality constraint by the introduction of a set of slack assignment parameters w_{i,N_j+1}, $w_{N_i+1,j}$ [135].

$$\begin{gathered} w_{ij} \geq 0 \\ \textstyle\sum_{i=1}^{N_i+1} w_{ij} = 1 \\ \textstyle\sum_{j=1}^{N_j+1} w_{ij} = 1 \end{gathered} \tag{4.9}$$

The set of assignments w_{ij} form a matrix of parameters $\mathbf{W}$ for which the i^{th} row defines the multiple point assignment for a model vertex $\underline{x}_i$ and the j^{th} column defines the multiple point assignment of a target point $\underline{y}_j$. The slack parameter w_{i,N_j+1} for a vertex determines the degree to which the vertex is unassigned to the target points. Similarly the slack parameter $w_{N_i+1,j}$ for a target point defines the degree to which the target point is an outlier that should be rejected in assigning the model. The equality constraints on the assignment in Equation 4.9 can be satisfied given the deterministic estimate for the parameters in Equation 4.8 through a process of row-column normalisation of the match matrix $\mathbf{W}$ [34]. The non-negativity constraint is automatically satisfied as the initial estimates for the parameters in Equation 4.8 are positive.

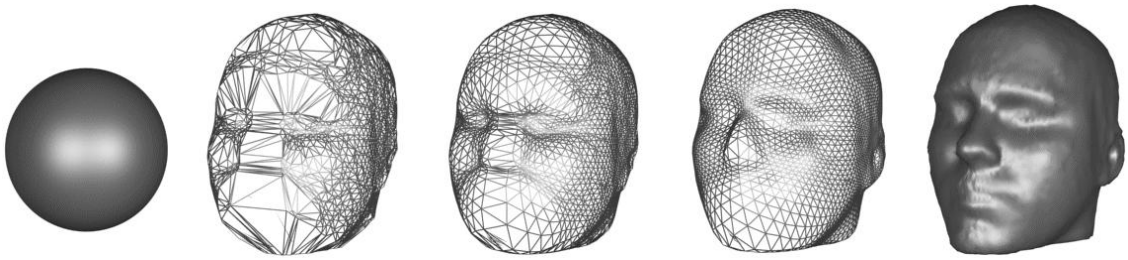


(a) $T = (0.25D)^2$ (b) $T = (0.125D)^2$ (c) $T = (0.033D)^2$ (d) ICP

Figure 4.8: Multi-point assignment of a sphere to a cube in 2D showing the sum of weighted assignments for different simulated temperatures T as a proportion of the 2D diameter D of the sphere, compared with closest point matches for ICP.

The unknown assignment parameters w_{ij} are derived at a fixed temperature T given the estimate in Equation 4.8 with row-column normalisation of the matrix $\mathbf{W}$ to satisfy the constraints in Equation 4.9. For a fixed assignment the model is then updated to satisfy the geometric matches. This process of assignment and transformation is repeated and the temperature gradually reduced to give a deterministic annealing approach to minimise the energy of the deformable model.

The technique provides a coarse-to-fine method of recovering the multiple point correspondence from the model vertices to the set of target surface points. The deterministic estimate for the assignment parameters in Equation 4.8 gives a weighting according to the relative distance to a target point $\|\underline{y}_j - \underline{x}_i\|$ with the temperature T defining the effective scale of matching in space, $T \sim \|\underline{y} - \underline{x}\|^2$. At a high temperature a wide range of target points are assigned a similar influence on a vertex. As the temperature is reduced the region of influence narrows. At the limit when $T \to 0$ the nearest point will dominate the estimated assignment giving a nearest-neighbour match as illustrated in Figure 4.8. The technique has the advantage that the initial scale for matching can be set according to the expected error in the shape of a model, avoiding matches to protrusions or "phantom" volume sections of the visual-hull. The scale can then be reduced as the model deforms to match the visual-hull, refining the model matches. The technique provides a mechanism to deal with the approximate shape information in the visual-hull with the slack assignment parameters defining the degree to which each vertex should be matched to the voxel data as the model deforms.



(a) Model (b) ICP (c) ICP + shape (d) Multi-point + shape (e) Target

Figure 4.9: Deformation of a uniformly triangulated sphere to match the vertex positions on a head model, comparing model deformation with ICP, shape-constrained ICP, and shape-constrained multiple point matching.

Multiple point matching is illustrated for a deformable model in Figure 4.9 in comparison with ICP matching. A sphere is deformed to fit the shape of a head and the nearest-neighbour heuristic in ICP is unable to recover the detailed geometry that does not lie close to the initial surface of the sphere. The shape constraint described in Section 4.2 attempts to preserve the even parameterisation of the surface vertices during deformation and constrains the independent vertex deformation in ICP to improve the recovered shape. The multiple point assignment, however, recovers the detailed geometry in the head by allowing the sphere to match multiple points that do not necessarily lie close to the sphere. The loss of detail in Figure 4.9(d) is due only to the resolution of the sphere and a greater number of vertices would provide greater surface detail.

4.4 3D Model-Based Reconstruction

The process of model optimisation using the data energy introduced in Section 4.3 and the shape regularisation introduced in Section 4.2 is outlined in the algorithm *match_visual_hull*. The visual-hull is first reconstructed to define the set of target surface voxels $\underline{y}_j$ for matching. The default local coordinates $(\alpha^0_{if}, \beta^0_{if}, h^0_{if})$ are then calculated for each triangle centred frame to define the original shape of the model. The model is then deformed to match the voxels by assigning the model vertices to the voxels and deforming the model to satisfy the assignment at a set of successively

reduced temperatures T until convergence. The slack parameters are initialised at each temperature T equivalent to the largest expected match at that scale using $\|\underline{y} - \underline{x}\| = \sqrt{T}$ in Equation 4.8. This encourages matches within the scale defined by the temperature T.

For a fixed assignment at a specific temperature T, the vertices of the model are deformed by gradient descent to minimise the energy function of the deformable model. Explicit Euler integration steps are taken in gradient descent requiring only local estimation of the energy gradient and deformation of the model vertices in parallel. The parameter λ is introduced to control the degree of shape regularisation during deformation and δ is used to define the step length taken in descent.

$$\frac{d\underline{x}_i}{dt} = -(\nabla \mathcal{E}_{\mathcal{D}} + \lambda \nabla \mathcal{E}_{\mathcal{R}}) \tag{4.10}$$

$$\frac{d\underline{x}_i}{dt} = \sum_{j=1}^{N_j} w_{ij}(\underline{y}_j - \underline{x}_i) - \lambda \frac{1}{N_f} \sum_{f=1}^{N_f} \left(\underline{x}_i - \underline{x}(\alpha_{if}^0, \beta_{if}^0, h_{if}^0) \right) \tag{4.11}$$

A coarse-to-fine approach to model deformation is introduced by setting the convergence criteria in optimisation according to the scale of matching defined by the temperature T. Convergence is defined where the maximum component of the gradient falls within the next scale of matching, $\|\frac{d\underline{x}_i}{dt}\|_{max} \leq \sqrt{cT}$ such that the assigned voxels will remain within the matching range at the next iteration of model deformation. Optimisation is also terminated where the energy function increases $\triangle \mathcal{E} > 0$ to prevent over-shoot in steepest descent minimisation. The final temperature scale for minimisation is defined as the desired error tolerance on the final shape of the model. In matching the discrete set of surface voxels this is set to be the voxel size, for a $1cm(0.01m)$ voxel size $T_{final} = 0.01^2$.

In practice the number of surface voxels can be large and a large subset will be redundant in matching where the distance from a model vertex to a voxel greatly exceeds the current temperature T and the match parameter is effectively zero. The matches are therefore restricted to a subset of surface voxels for each model vertex, reducing the number of match parameters to be updated and stored. The match matrix $\mathbf{W}$ becomes compact and additional book-keeping is required to track which surface voxels

match which vertex, with all unmatched voxels having a zero assignment. The subset of matches for each vertex is obtained as the set of N_v closest surface voxels. Spatial partitioning with an octree representation [156] is used for efficient retrieval of the closest points and only the voxels with a surface normal in the same half-plane as the vertex normal on the model are retrieved in order to avoid inconsistent surface matches.

Input: Model, $\underline{x}_i$
Octree, $\underline{y}_j$
Temperature, T

Output: Assignment, w_{ij}

Procedure: *set_voxel_assignment*

1. for (*each vertex* i)
2. read (N_v *closest voxels from octree*)
3. set (*assignments* $w_{ij} = \exp(-\|\underline{y}_j - \underline{x}_i\|^2/T)$)
4. set (*slack assignments* $w_{N_i+1,j}, w_{i,N_j+1} = \exp(-1)$)
5. while (*change* $\triangle w >$ *small_number*)
6. *normalise_rows*(w_i)
7. *normalise_columns*(w_j)

Input: Model, $\underline{x}_i$
Camera parameters, $\mathbf{P}_m$
Image silhouettes, I_m

Output: Updated model, $\underline{x}_i$

Procedure: *match_visual_hull*

1. *reconstruct_visual_hull*($\underline{y}_j$)
2. *construct_octree*($\underline{y}_j$)
3. set (*local coordinates* $\alpha_{if}^0, \beta_{if}^0, h_{if}^0$)
4. set (*temperature* $T = T_{init}$)
5. while ($T > T_{final}$)
6. *set_voxel_assignment*(w_{ij})
7. while ($\|\frac{d\underline{x}_i}{dt}\| \geq \sqrt{cT}$ and $\triangle\mathcal{E} < 0$)
8. set ($\frac{d\underline{x}_i}{dt} = -(\nabla\mathcal{E}_\mathcal{D} + \lambda\nabla\mathcal{E}_\mathcal{R})$)
9. set ($\underline{x}_i = \underline{x}_i + \delta\frac{d\underline{x}_i}{dt}$)
10. set ($T = c \times T$)

Optimisation of a model to match the surface of the visual-hull has a complexity of order $O\left(N_i N_v N_\nabla N_{anneal}\right)$ where N_∇ is the number of steps in steepest descent and N_{anneal} is the steps taken in deterministic annealing. At each temperature scale T the assignment

parameters w_{ij} must be computed at each vertex for every surface voxel with a complexity $O(N_i N_j)$. This cost is reduced using only N_v closest points rather than the complete set of N_j voxels. For each vertex, N_v closest voxels are retrieved by visiting $O(N_v)$ nodes in an octree structure and maintaining a sorted queue of closest nodes with a cost $O(\log(N_v))$ to insert each node into the queue. Using spatial partitioning and only N_v voxels per vertex, the cost of assignment is therefore reduced to $O(N_i N_v \log(N_v))$. Iterated row plus column normalisation of the now compact matrix $\mathbf{W}$ has a lower complexity $O(N_i N_v)$. The model is then updated to satisfy the assignment. If N_∇ steps are taken in gradient descent the cost of updating each vertex location to match the assignment to N_v voxels is in the order $O(N_i N_v N_\nabla)$. In practise $N_\nabla > \log(N_v)$ and the total cost at each temperature has a complexity $O(N_i N_v N_\nabla)$. The alternate process of assignment and optimisation is performed N_{anneal} times in deterministic annealing and the final complexity is $O(N_i N_v N_\nabla N_{anneal})$. The number of annealing steps depends on the annealing schedule defined by c, $N_{anneal} = \log(T_{final}/T_{init})/\log(c)$. The number of steps in steepest descent is inversely proportional to the step length taken $N_\nabla \sim (1/\delta)$.

4.5 Evaluation

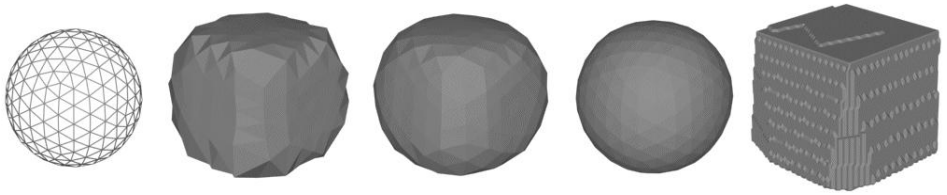
A shape constrained deformable model framework has been presented to match a generic humanoid model to the shape information in multiple view silhouettes in the presence of visual ambiguities. There are two key considerations in the application of this framework. Firstly, the regularisation term for the deformable model defines a trade-off between fidelity in fitting the available data and preserving the shape of the generic model in the presence of visual ambiguities. The influence of the shape constrained regularisation in model deformation is evaluated in Section 4.5.1. The second consideration is the trade-off obtained between the complexity of the algorithm and minimisation of the reconstruction error in data fitting. The technique uses a multiple point matching approach that is refined in a coarse to fine framework using deterministic annealing. The complexity of the algorithm can be reduced by reducing the number of points in matching and using a faster annealing schedule. However, this can lead to a local minima in the energy function of the deformable model and a higher

reconstruction error. The effect of algorithm complexity versus reconstruction error is evaluated in Section 4.5.2. The ideal test cases used in Chapter 3 and shown in Figures 3.3 and 3.4 are used to test the model-based reconstruction framework and to perform a quantitative analysis of reconstruction error. The performance of the technique is then demonstrated in Section 4.5.3 both for the ideal case and real data. In all cases the visual-hull is reconstructed at a 1cm voxel resolution. A 1cm voxel size encompasses the reprojection error in the real data, ensuring that a voxel in the visual-hull will project to the captured image silhouettes with the inexact camera calibration data.

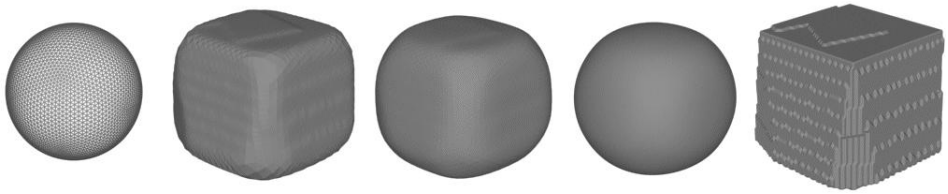
4.5.1 Shape constraint

The shape constraint for the deformable model is designed to preserve the prior shape and parameterisation of a surface model in fitting a target shape. The effect of the shape constraint is controlled by the parameter λ which defines the trade-off between data-fitting and shape regularisation in the deformable model. The influence of the parameter λ is examined with ideal values used for the other parameters in the algorithm. The annealing constant $c = 0.5$ is chosen to obtain a conservative change factor of 0.71 in the matching range at each step of annealing, a large number of target voxels are used with $N_v = 1000$, a small step length is taken in steepest descent with $\delta = 0.01$, and an initial error of $10cm$ is assumed on the shape of the model. It should be noted that in this test example an exact match is not expected as the generic model shape is different to that of the data.

The effect of the shape constraint is illustrated in Figure 4.10 in fitting the shape of the sphere to the visual-hull for the cube. The constraint preserves the even surface parameterisation and round shape of the sphere. At a high value for λ, the regularisation term dominates in the deformable model and the shape of the sphere is preserved. It is interesting to see that the resolution of the triangulated model surface influences the choice of the parameter λ due to the scale dependence of the shape constraint. With a low resolution model the triangles span a larger surface area and the shape constraint has a greater relative influence. The value of λ which preserves the shape of the sphere is therefore reduced with a lower resolution triangulation.

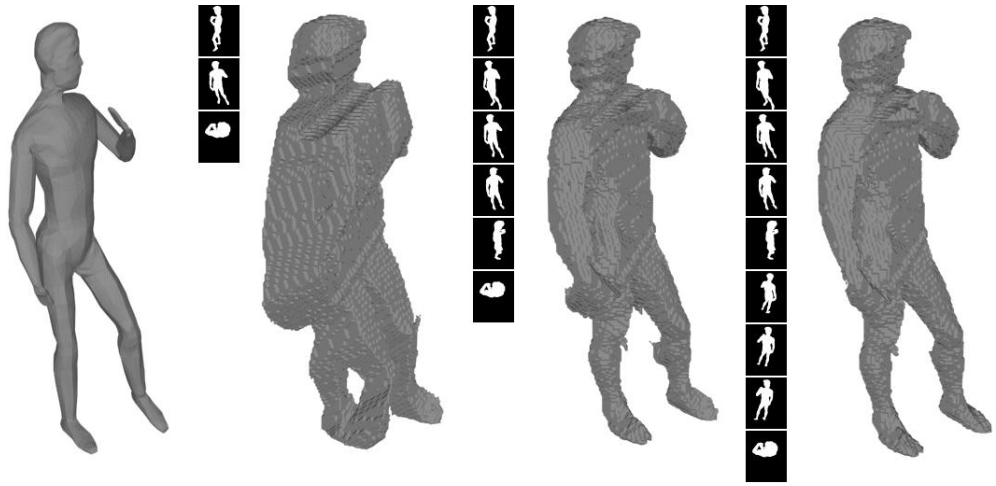


(a) Model (b) $\lambda = 0.1$ (c) $\lambda = 1.0$ (d) $\lambda = 10.0$ (e) Visual-hull



(a) Model (b) $\lambda = 1.0$ (c) $\lambda = 10.0$ (d) $\lambda = 100.0$ (e) Visual-hull

Figure 4.10: Deformation of a sphere to fit the visual-hull for a cube with different degrees of shape constraint at two different resolutions in the triangulated model.

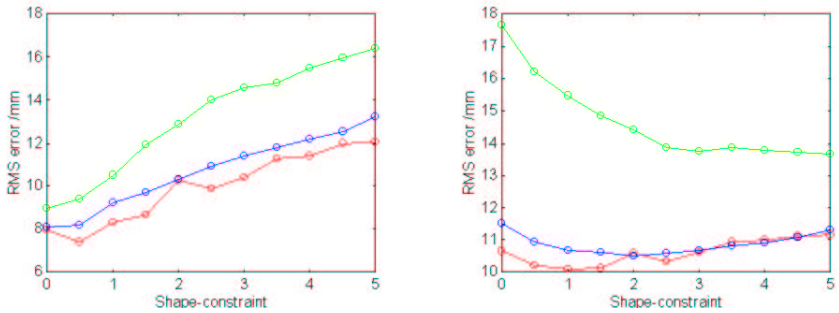


(a) Model (b) 3 views (c) 6 views (d) 9 views

Figure 4.11: The initial generic humanoid model and the visual-hull reconstructed from 3, 6 and 9 image silhouettes.

The shape-constraint is assessed in reconstructing a person using our generic humanoid model and the ideal data-set shown in Figure 3.4 as ground truth. The initial model is shown in Figure 4.11, in comparison with the visual-hull reconstructed from three, six and nine camera views. The model is deformed to match the surface of the visual-hull for each set of views with different values of the control parameter λ. Figure 4.12 gives the root mean squared (RMS) error on the model surface, first in comparison with the visual-hull and then with the underlying surface of the original range data. The error at a point on the model surface is defined as the minimum distance to each target surface and the RMS error is computed across the entire model surface using the Metro tool [36]. The error in fitting the visual-hull shown in Figure 4.12(a) increases with λ as the model is increasingly constrained to preserve the initial model shape. The error in fitting the underlying range data in Figure 4.12(b) demonstrates the trade-off between shape information in the prior model and the accuracy of the shape data in the visual-hull. With three camera views the visual-hull shown in Figure 4.11(b) provides limited shape information and the prior shape of the model is needed to constrain shape reconstruction. With nine camera views the shape constraint for a minimum reconstruction error is reduced as the shape in the visual-hull provides a closer approximation to the underlying surface as shown in Figure 4.11(d). The formulation for a shape constrained deformable model enables the approximate shape of the underlying surface to be recovered for three, six and nine camera views even with extensive visual ambiguities in the shape as shown in Figure 4.11.

As the number of camera views is increased the shape of the visual-hull provides a closer approximation to the underlying shape of a person. However, even with a large number of views the visual-hull can contain ambiguities were surfaces are self-occludeding. Where a surface is not represented in the visual-hull, the shape of the generic model must be preserved. This is demonstrated at the left-elbow in the visual-hull shown in Figure 4.11 where the inside surface of the arm is folded. The model recovered in fitting the visual hull for 9 camera-views is now shown in Figure 4.13 for a different animated elbow position. Even though the minimum RMS error for the entire model is obtained at a value $\lambda = 1.0$ the shape constraint must be increased to preserve the original shape of the model where the surface data is not present at the elbow. Figure 4.13



(a) Error to visual-hull (b) Error to range-data

Figure 4.12: The RMS error from the deformed model to (a) the visual-hull and (b) underlying range data against the shape constraint parameter λ, in fitting the visual-hull reconstructed from 3(green), 6(blue), and 9(red) views.

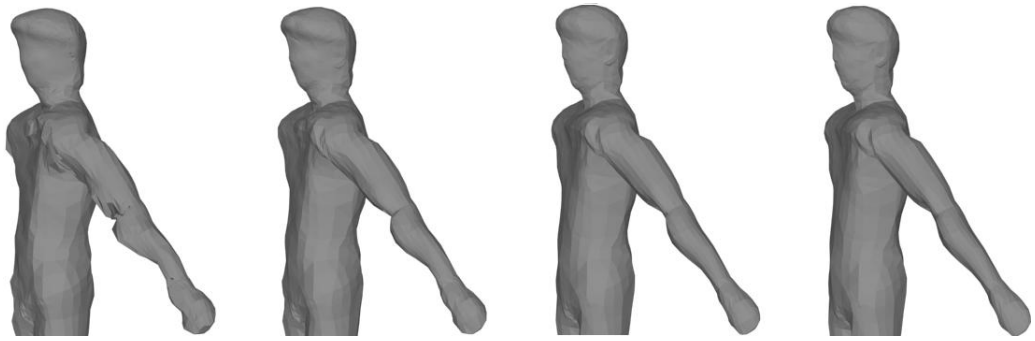
demonstrates that the constraint term also preserves the relative parameterisation of the surface allowing the model to be animated with the predefined kinematic structure for the generic model. A value of $\lambda = 3.0$ is chosen to provide a trade-off between a minimum reconstruction accuracy in Figure 4.12 and preserving the model shape and parameterisation in the presence of ambiguities as shown in Figure 4.13.

Conclusion:

1. **The regularisation energy term for the deformable model provides a shape constraint that preserves the original shape and parameterisation of the generic model;**

2. **The degree of shape regularisation required in the deformable model is governed by the trade-off between the accuracy in the shape of the visual-hull and the requirement to preserve the original model geometry where the shape data is inaccurate due to visual ambiguities; and**

3. **The shape constrained deformable model can provide an accurate estimate of underlying surface shape in the presence of large ambiguities in reconstruction from a limited set of camera views.**



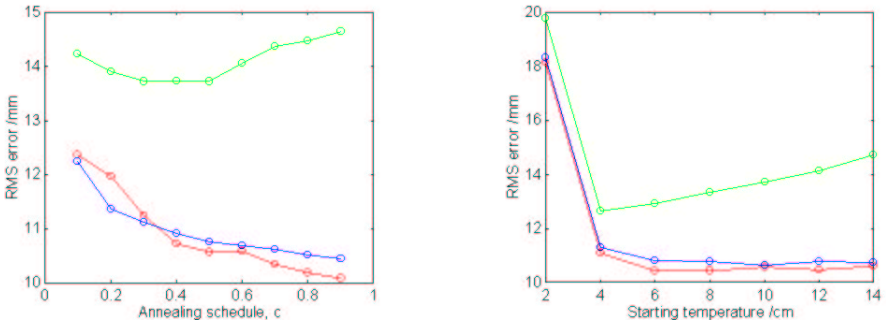
(a) $\lambda = 0.0$ (b) $\lambda = 1.0$ (c) $\lambda = 3.0$ (d) $\lambda = 5.0$

Figure 4.13: An animated pose for the model reconstructed in fitting the visual-hull for nine camera views demonstrating the preservation the model animation structure with the shape constraint term λ.

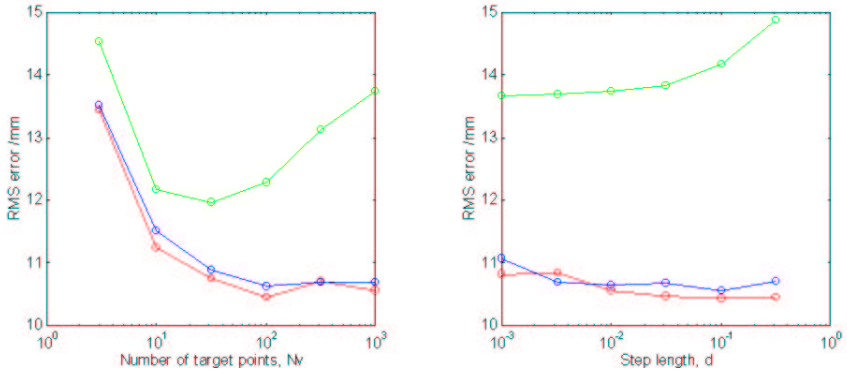
4.5.2 Complexity

The reconstruction error in fitting the ground truth data is now evaluated with changes to the remaining control parameters that influence the complexity of the deformable model algorithm. Figure 4.14 shows the RMS error computed to the range data in fitting the generic humanoid model to each visual-hull shown in Figure 4.11 for $\lambda = 3.0$. We can see from the graphs that a slow annealing schedule c, a high initial temperature T_{init}, and a large number of target points N_v all provide a lower reconstruction error and that the error is relatively stable with the step-length δ. The exception arises in fitting the visual-hull from three camera views, where the shape of the visual-hull provides a poor approximation of the underlying surface and a greater accuracy in data fitting actually provides a worse reconstruction error. The complexity of the algorithm increases with a slow annealing schedule c, a high initial temperature T_{init}, a large number of target points N_v and a small step length δ. The parameters must therefore be chosen to give a trade-off between the computational cost and the reconstruction error. The reconstruction errors shown in Figure demonstrate that the technique is relatively insensitive to the exact values of the parameters used. Intuitive values of the parameters can therefore be chosen, $c = 0.5$, $T_{init} \equiv 10cm$, $N_v = 100$, $\delta = 0.1$.

Conclusion:



(a) Error vs. annealing schedule c (b) Error vs. starting temperature T_{init}



(a) Error vs. matched points N_v (b) Error vs. step length δ

Figure 4.14: RMS reconstruction error to range-data against changes in the deformable model parameters in fitting the visual-hull reconstructed from 3(green), 6(blue), and 9(red) views.

4. **The reconstruction error is relatively insensitive to the exact values of the parameters defining the optimisation of the deformable model.**

4.5.3 Shape reconstruction

The generic model is expected to be aligned with the target data for model-based reconstruction. The range of convergence for the technique is now examined by simulating a range of errors on the pose of the generic humanoid model in fitting the visual-hull reconstructed from nine camera views shown in Figure 4.11. A random rotation is applied to the shoulder, elbow, hip, knee and neck joints of the generic model up to a maximum angular limit. In order to increase the range of convergence in data fitting

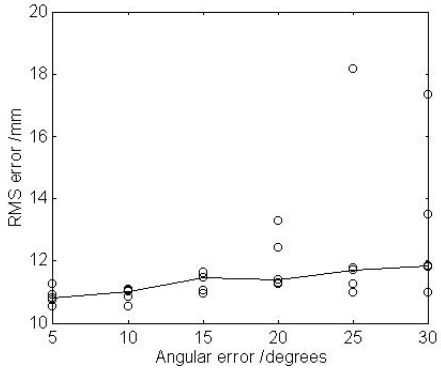


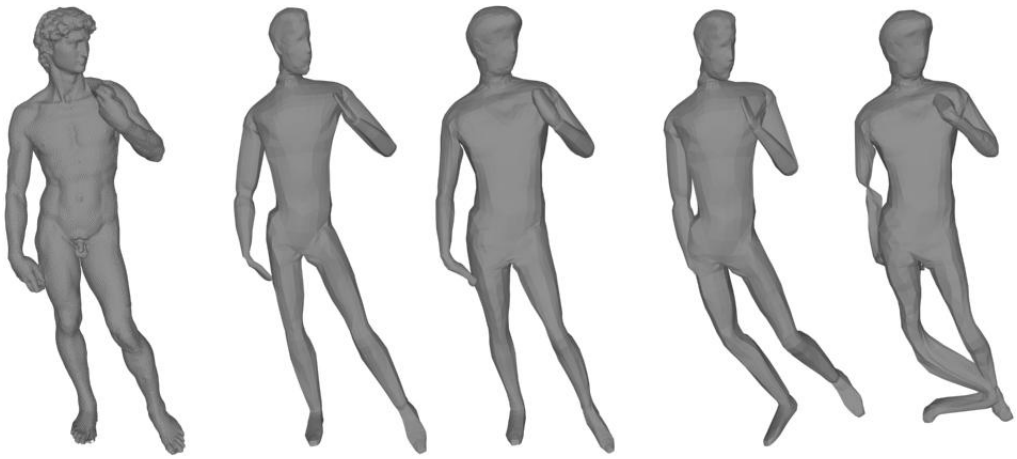
Figure 4.15: RMS reconstruction error to range-data with different angular errors introduced on the pose of the arms, legs and head of the generic model. The median error is marked across the test cases.

the initial starting temperature for the deformable model is set equivalent to $30cm$, an estimate of the average maximum distance to the target data across all test cases. Figure 4.15 shows the RMS error to the underlying range data against the maximum error introduced in the model pose. It can be seen that the error in reconstruction increases with the error in pose. The worst-case reconstruction errors arise where the model is initially aligned with an incorrect surface as shown in Figure 4.16(e) for a 30^o error in pose. To put this in context a 10^o error over the length of the arm is equivalent to a positional error of approximately $10cm$. This already represents a relatively large error in pose. Figure 4.16(c) demonstrates that the model with the worst case reconstruction error at 10^o does reproduce the target surface shape.

Conclusion:

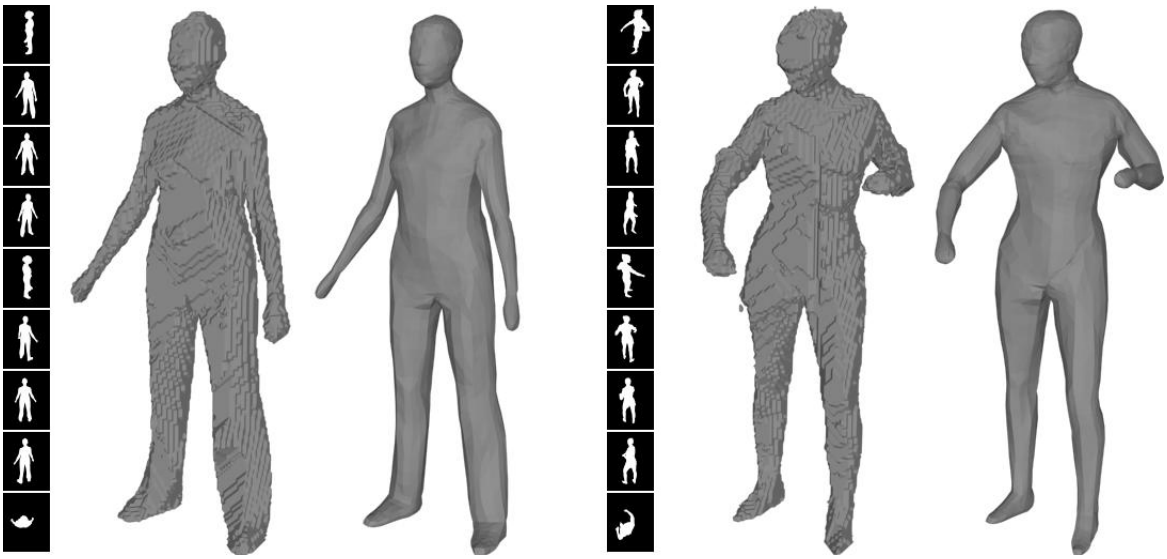
5. **The deformable model will fit the target data within the expected range of errors in the initial pose of the humanoid model.**

The performance of the technique is now demonstrated in Figure 4.17 in fitting the visual-hull of a person reconstructed at a 1cm voxel resolution from nine cameras in a studio. The technique subjectively provides a visually realistic shape model of a person that reproduces the shape observed in multiple view silhouettes. The next step is to derive the model texture map from the images to give a visually realistic



(a) Range data (b) 10^o Error (c) Reconstruction (d) 30^o Error (e) Reconstruction

Figure 4.16: Reconstructed models with the worst-case error to the (a) range data for (b) 10^o maximum error in pose and (d) a 30^o maximum error in pose.



(a) Visual-hull (b) Deformed model (c) Visual-hull (d) Deformed model

Figure 4.17: Reconstructed model in fitting the visual-hull reconstructed from nine camera views for two different subjects in different poses.

model appearance. The drawback found for the technique is that the models derived from the visual-hull can only provide an approximate shape for a person. Figure 4.18 demonstrates the problems that can arise in deriving a texture map for a model with only an approximate shape. The projected location of a model vertex in different images will not correspond exactly to the same image point due to errors in the model geometry derived from the visual-hull. The colour texture derived from different images will then not be in correspondence.

Figure 4.18: The image plane projection of model vertices showing incorrect image correspondence due to errors in the approximate model derived from the visual-hull.

Conclusion:

6. **The model-based framework provides a smooth approximation to the shape of a person from multiple view silhouettes; and**

7. **Shape from silhouette does not provide an accurate shape model for the recovery of multiple view appearance.**

4.6 Summary

The problem of adapting a generic humanoid model to match multiple image silhouettes has been addressed. A new technique is presented to deform a model to fit multiple 2D silhouettes simultaneously by matching to the 3D surface of the visual-hull. This

approach avoids inconsistent matches that can arise between a model and a silhouette in 2D due to self-occlusions as illustrated in Chapter 3. Model fitting is formulated as a constrained energy minimisation task and a model is optimised to fit the shape of the visual-hull. A multiple point matching scheme is introduced to obtain robust matches in the presence of visual ambiguities and avoid local minima in the assignment of the model to the surface of the visual-hull. A shape constraint is introduced to regularise the deformation of the model in data-fitting. A novel shape constraint is formulated for a triangulated surface model to preserve the relative position of the vertices defining the surface shape. This ensures that the correspondence between the vertices and the animation structure of the model remains valid. The model-based reconstruction algorithm presented in this chapter enables an approximate shape model to be recovered from multiple view image silhouettes. However, the problem remains that with only an approximate shape the projection of a model into different camera images will not be in exact correspondence for the recovery of colour texture. This is problematic in integrating appearance information from multiple views as misalignment results in visual artifacts.

Chapter 5

Model-Based Reconstruction to match Multiple View Appearance

Model-based reconstruction of animated human models has been presented using shape from silhouette in Chapter 4. Image silhouettes can be extracted from the controlled background environment in a studio and provide a robust constraint on shape. However, the shape information available from multiple silhouettes is limited resulting in only an approximate shape model for a person. Where the shape of the model is incorrect the vertices are not then in exact correspondence between camera images for the recovery of model texture. In this chapter alternative multiple view reconstruction techniques are considered to refine the shape estimated from multiple views to give correct correspondence for all surface points between views. A model-based technique is introduced to optimise a generic humanoid model to recover both the shape and appearance of a person.

Problem Statement:

- **Optimise the 3D surface of a generic humanoid model to match both the shape and appearance in multiple images.**

Existing multiple view reconstruction techniques are evaluated in Section 5.1 to give improved shape data over the visual-hull for model optimisation. Voxel colouring [140] and multiple view stereo [123] have been used previously to reconstruct the shape of

a person in a studio [175, 123]. Both voxel colouring and multiple view stereo are considered to reconstruct the target shape with a consistent appearance across multiple views. These techniques are found to suffer in the presence of visual ambiguities such as self-occlusions or a limited variation in appearance in the images. In Section 5.2 a model-based approach is introduced to combine the complementary shape data from image silhouettes and stereo correspondence. The model-based technique makes use of the prior shape information in the generic humanoid model to overcome visual ambiguities in multiple view reconstruction. In Section 5.3 sparse feature matching is also introduced to constrain the reconstruction of fine geometric detail that cannot be recovered with the limited resolution whole-body images of a person. The final technique is evaluated in Section 5.5 and compared with voxel colouring and multiple view stereo. This work is presented in *"Towards a 3D Virtual Studio for Human Appearance Capture"* Starck and Hilton [151] and the paper *"Model-Based Multiple View Reconstruction of People"* Starck and Hilton [150].

5.1 Multiple View Reconstruction

There are two general approaches in the literature to the problem of geometric reconstruction of a scene from multiple calibrated cameras as outlined in Chapter 2. Techniques are based either on searching for the correspondence between image points to reconstruct 3D position in a scene from the image plane locations, or on searching for the surface with a consistent appearance when projected to the images. Both approaches have been applied to reconstruct models of people in a studio, either through multiple view stereo [87], or voxel colouring [175] respectively. These two techniques are evaluated in this section as a means to refine the shape obtained from image silhouettes to provide more accurate shape data for the model-based framework described in Chapter 4. In Section 5.1.1 the appearance of a person in multiple images is defined in terms of a surface reflectance model. Voxel-colouring is then assessed in Section 5.1.2 and multiple view stereo in Section 5.1.3 to derive the surface in a scene with a consistent multiple view appearance.

5.1.1 Multiple view appearance

A camera image samples the complex view-dependent light-field reflected from the visible surfaces in a scene. The appearance of a surface is defined by the surface shape, the bi-directional reflectance distribution function (BRDF), and the illumination. In this work the spatially varying BRDF is unknown and the complex lighting environment in the studio is undefined. A set of simplifying assumptions are therefore made to define surface appearance in multiple views for shape reconstruction.

Image-based reconstruction is often based on the assumption that all surfaces in a scene follow the Lambertian reflectance model [81]. The Lambertian model states that light is reflected equally in all directions. If the assumption is also made that the response of each camera is equal then a surface point will have the same colour in all images. However, the Lambertian model only applies to perfectly diffuse surfaces and can fail to represent the reflectance of the human body where non-diffuse surfaces such as skin and clothing show specular highlights in images.

In this work the dichromatic reflectance model [145] is adopted to define the appearance of a person across multiple views. The dichromatic model is this simplest reflectance model that accounts for specular surface reflections. The RGB surface colour $\underline{I}$ observed in an image is defined as the sum of two components, a body reflectance and a surface interface reflectance. The body component is the diffuse surface colour $\underline{I}_D$ given by the Lambertian model and the interface component is the proportion of the illuminant light colour $\underline{I}_L$ reflected from a surface to give a specular highlight.

$$\underline{I} = \underline{I}_D + \varepsilon \underline{I}_L \tag{5.1}$$

The cameras in a studio are "white-balanced" such that the illumination is normalised to be white $\underline{I}_L = \frac{1}{\sqrt{3}}\{1,1,1\}$. The assumption is made that the specular component does not saturate the individual colour components. The dichromatic reflectance model then states that there is a linear change in the colour observed from different viewpoints proportional to $\underline{I} = \{1,1,1\}$.

5.1.2 Voxel colouring

Voxel colouring was introduced by Seitz and Dyer [140] as a method to derive the occupied voxels in a scene that have a consistent colour across a set of camera images, providing the photo-hull of the scene rather than the visual-hull. Seitz and Dyer [140] proposed the *Voxel Coloring* algorithm. The technique uses cameras placed on one side of the scene and traverses voxels one plane at a time in a single pass away from the cameras. Voxels that are not occluded in the images by occupied voxels found in a previous plane are tested for colour consistency and set as occupied if consistent. Colour consistency is tested for the unoccluded pixel colours at the projection of a voxel in each image using a fixed threshold on the acceptable variance in the pixel colours. Voxel occlusion is tested by maintaining a visibility image for each camera in which the image pixels are set as occluded for each colour consistent voxel found.

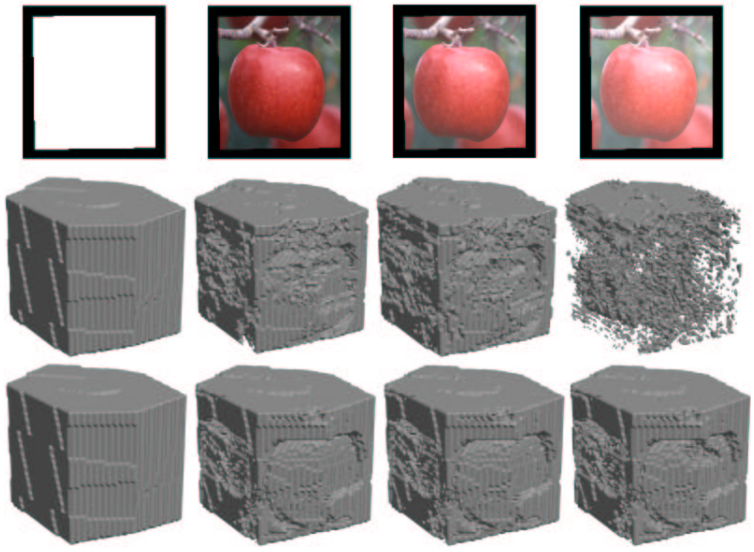
Voxel Coloring is used here to refine the shape of the visual-hull from the image silhouettes. The plane-sweep is performed and colour consistency is tested for each voxel in the visual-hull. The voxels forming the visual-hull that are not colour consistent are therefore removed, refining the shape. Colour consistency is tested in the algorithm for the UV colour components of the YUV colour space rather than in RGB. In the YUV colour space the luminance Y of a colour is isolated from the chrominance UV through a linear transformation of the RGB colour components. A specular component of a white light source $\underline{I} = \frac{\varepsilon}{\sqrt{3}}\{1,1,1\}$ is therefore removed from the UV components as $YUV = \{\frac{\varepsilon}{\sqrt{3}}, 0, 0\}$. This provides an illuminant invariant colour consistency test and removes the influence of specular highlights in the images.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5.2}$$

Voxel colouring is illustrated in Figure 5.2 for the ideal test case of reconstructing a cube from a set of simulated camera images. Five colour images for the cube are created using a colour photograph mapped onto each face as shown in Figure 5.1. The photo-hull reconstructed with the *Voxel Coloring* algorithm is compared with the visual-hull

obtained from the image silhouettes in Figure 5.2. Reconstruction using an RGB colour consistency test and a UV colour consistency is also compared at different simulated levels of specular highlight in one camera image for one face of the cube. It is apparent from Figure 5.2 that voxel colouring carves away the protrusions in the visual-hull and that voxel colouring can be performed in the presence of specular highlights using the UV colour consistency test. However, the resulting voxel model becomes a noisy representation of the true surface. Voxel colouring suffers from holes in a scene with incorrectly carved voxels, and protrusions where there are similarly coloured image regions or the voxels become occluded in the images and consistency cannot be tested.

Figure 5.1: Five views of a texture mapped cube to test multiple view reconstruction.



(a) Visual-hull (b) Photo-hull $\varepsilon = 0.0$ (b) $\varepsilon = 0.25$ (c) $\varepsilon = 0.50$

Figure 5.2: Reconstruction of the photo-hull for a cube with a simulated specular highlight in one camera view. Showing the photo-hull from an RGB (middle-row) and UV (bottom-row) colour consistency test in comparison with the visual-hull.

5.1.3 Multiple view stereo

In this section multiple view stereo is considered as an alternative to volumetric reconstruction with the visual-hull or the photo-hull. Area-based stereo establishes the correspondence between camera images to reconstruct geometry and so provides the surface shape in a scene that has a consistent image correspondence. The multiple view stereo approach presented by Narayanan et al. [123] used in the "Virtualized Reality" to reconstruct scenes of people [87] is evaluated. Dense area-based stereo matching is performed to derive the correspondence between a pair of cameras in the studio and construct a 2.5D depth to the visible surface in the scene. The depth map derived for each pair of cameras in the studio is then combined into a single 3D surface model using a volumetric fusion technique. In this work the approach is extended by making use of the visual-hull to constrain the search space for stereo matches in order to recover the scene geometry that lies inside the bounding constraint of the visual-hull.

Area-based stereo uses a correlation score to quantify the similarity of two images across a small window area in each image. If the assumption is made that the geometry of the scene is locally planar then the corresponding pixels in two image windows can correspond to the same point in the scene. Many different matching scores have been proposed to determine the correlation of pixels between two image windows for matching [57]. Under the dichromatic reflectance model in Equation 5.1 and a planar surface we can expect a linear change in pixel intensity between images. The normalised cross-correlation score is therefore used, which allows for an affine transformation in intensity between images [57]. The normalised cross-correlation $\mathcal{C}(\underline{u}_1, \underline{u}_2)$ of pixel intensity I across a window centred at pixels $\underline{u}_1, \underline{u}_2$ in two intensity images I_1, I_2 is given as.

$$\mathcal{C}(\underline{u}_1, \underline{u}_2) = \frac{\sum_{\underline{w}} \Big(I_1(\underline{u}_1 + \underline{w}) - \overline{I}_1(\underline{u}_1) \Big) \times \Big(I_2(\underline{u}_2 + \underline{w}) - \overline{I}_2(\underline{u}_2) \Big)}{\sqrt{\sum_{\underline{w}} \Big(I_1(\underline{u}_1 + \underline{w}) - \overline{I}_1(\underline{u}_1) \Big)^2} \times \sqrt{\sum_{\underline{w}} \Big(I_2(\underline{u}_2 + \underline{w}) - \overline{I}_2(\underline{u}_2) \Big)^2}} \tag{5.3}$$

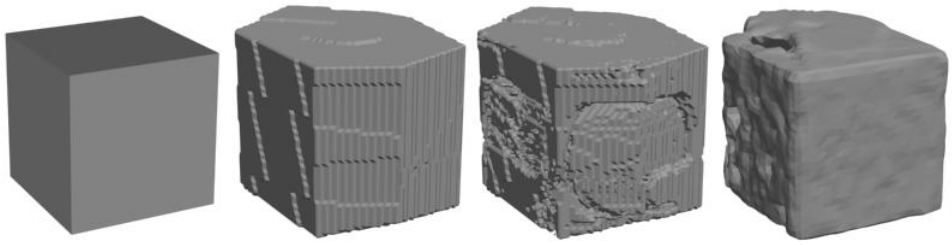
A depth map is reconstructed for each stereo pair using the epipolar, ordering and continuity constraints. The normalised cross-correlation is first calculated for each

pixel in the left-hand image of a stereo pair, with each pixel along the epipolar line in the right-hand camera image. Image rectification is performed so that the epipolar lines for the cameras correspond to the pixel rows in the left and right images [64]. The correlation scores are then calculated using square image windows of equal size shifted in 1D along pixel rows in the images. With a square image window the assumption is made that the scene is locally fronto-parallel to the rectified images and viewed at an equal scale in each image.

Pixel matches in the right image are derived from the correlation scores subject to the ordering and continuity constraints using dynamic programming. The epipolar rows in the right-hand image provide the disparity component for the pixels in the left image. The correlation scores are stored as a 3D data-set $\mathcal{C}(\underline{u}_l, d_r)$ with a left-hand image position $\underline{u}_l$ and right-hand disparity d_r. Here the feasible disparity range for the matches is constrained by only computing the correlation scores at points in the data-set that lie within the visual-hull. The connected surface in the disparity space with the maximum total correlation score is then extracted through the two stage dynamic programming (TSDP) technique introduced by Sun [155]. The disparity component of the extracted surface provides the corresponding pixel match in the right-hand image for each left image pixel. From each match the depth in the scene is calculated from the triangulated 3D position of the matches and a 2.5D depth map is constructed. Sub-pixel accurate disparities are derived by fitting a quadratic parabola in the region of each match to give greater depth resolution [63].

$$d'_r = d_r + \frac{1}{2}\frac{\mathcal{C}(\underline{u}_l, d_r - 1) - \mathcal{C}(\underline{u}_l, d_r + 1)}{\mathcal{C}(\underline{u}_l, d_r - 1) - 2\mathcal{C}(\underline{u}_l, d_r) + \mathcal{C}(\underline{u}_l, d_r + 1)} \tag{5.4}$$

Multiple pairs of cameras are required to sample the complete surface of a scene. The 2.5D depth maps for all the stereo pairs are merged into a single 3D model using a volumetric fusion technique [123]. The fusion technique averages the depth to the surface of the scene at a discrete set of points on a volumetric grid and extracts the shape of the scene as the zero-distance surface inside the volume. Here the discrete volume defined by the visual-hull is used. At the corner of each occupied voxel in the visual-hull a 3D depth value is derived. The depth value is calculated by projecting

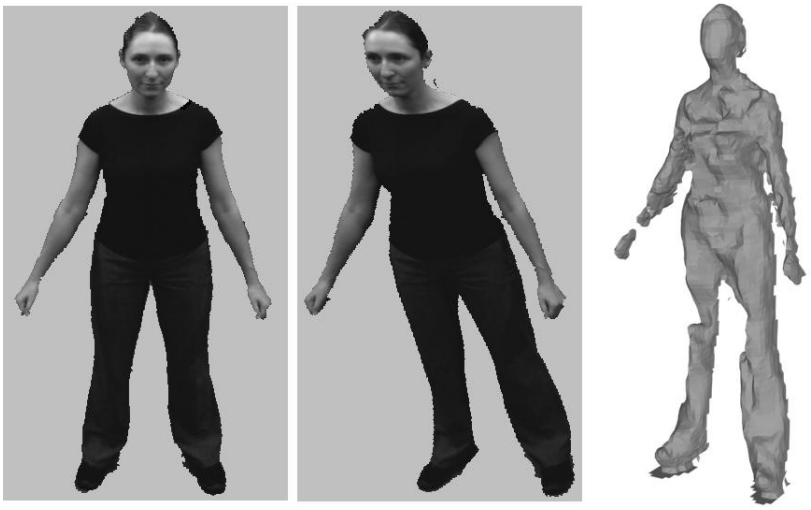


(a) Target shape (a) Visual-hull (b) Photo-hull (c) Merged stereo

Figure 5.3: Multiple-view stereo reconstruction in comparison with the visual-hull and photo-hull given five simulated views of a coloured cube.

the corner to all the depth maps and searching for the closest 3D surface point in each view. An average is then taken for the depth to each 3D point within a set tolerance of the closest surface point across all views. The tolerance is automatically set as the size of the voxels used in volumetric fusion in order to average the surfaces that fall within each voxel. A signed distance function is constructed by assigning a positive depth value where a corner lies outside a surface and a negative value where a corner falls inside the surface. The surface of the scene is then extracted as the zero-valued iso-surface of the distance function using the *Marching Cubes* algorithm [106].

Multiple view stereo reconstruction is illustrated in Figure 5.3 for the problem of reconstructing the coloured cube shown in Figure 5.1 using four stereo pairs formed from adjacent cameras in the five simulated images. Multiple view stereo provides an improved shape on the front face of the cube compared to the photo-hull by matching the local appearance between images rather than matching image colour. The stereo technique is based on two principal assumptions. The first is that the scene consists of locally fronto-parallel surfaces. This assumption breaks where there are depth discontinuities from self-occlusion or where the scene is non-planar or has a slanted surface with respect to a stereo pair. The second assumption is that there is a sufficiently distinct local intensity variation within an image window to correctly locate image correspondence. In regions of low intensity variation the correlation of image windows is ambiguous. In Figure 5.3(d) the assumption of fronto-parallel surfaces is incorrect for the side faces of the cube and the stereo correspondence provides a noisy surface esti-



(a) Left rectified image (b) Right rectified image (c) Stereo reconstruction

Figure 5.4: Reconstruction of a person observed from a stereo pair of images.

mate. For the reconstruction of people the assumption of adequate variation in image intensity can be a fundamental problem as people often wear clothing with a uniform appearance as shown in Figure 5.4.

5.1.4 Summary

Multiple-view reconstruction techniques have been presented from the literature to improve the shape data available from image silhouettes. Voxel colouring derives the photo-hull of a scene through colour matching and multiple view stereo derives the surface that maximise the correlation of local image windows between different views. These techniques suffer from visual ambiguities in reconstruction. Voxel colouring provides a noisy and limited refinement of the visual-hull, and stereo correlation will fail where there is a limited intensity variation in the images. Stereo matching does however provide an improved shape reconstruction in image regions where there is a local variation in appearance.

In the recovery of model texture the image correspondence is critical where the visual appearance in the images varies. Where there is a local variation in the appearance and the correspondence in the images is incorrect, the texture recovered from different

images will be out of alignment on the surface of the model. On the other hand, where the variation in appearance is only small the correspondence in the images is not critical and an approximate correspondence is sufficient. Stereo and silhouette data therefore provide complementary shape cues to recover a surface model with a consistent multiple view appearance. Stereo matching provides the image correspondence where there is a local variation in appearance and shape from silhouette provides an approximate correspondence where stereo matching fails. A model based approach is introduced to combine these shape cues in reconstruction.

5.2 Model based stereo

In this section a technique is introduced to optimise a prior model to maximise stereo correlation between pairs of camera images and satisfy the constraint on shape imposed by multiple view silhouettes. The prior model provides an initial estimate of the correspondence between images which is then optimised with multiple view stereo. The geometry of the model is used in optimisation to simplify the search for correspondence in stereo matching and account for self-occlusions. Silhouette data is incorporated in model optimisation where there is a limited local variation in image appearance and stereo matching fails. The technique provides a model-based approach to scene reconstruction to satisfy the available appearance information across multiple views.

5.2.1 Local surface optimisation

Stereo correlation has been presented previously to optimise the surface of a model to match the appearance in multiple images. Fua and Leclerc [61] introduced *Object-Centred Reconstruction* in which an initial estimate of a surface is optimised to maximise the stereo correlation and consistency of shading between a set of images. An object-centred approach is based on an initial reconstruction of a scene and differentiated from model-based reconstruction that uses a prior scene model. The technique can however be equally applied to optimise a prior model. The formulation for optimisation with respect to stereo correlation is briefly described here. The technique for shape from

shading is not presented as it requires the assumption of a Lambertian reflectance model and a-priori knowledge of the lighting in the scene.

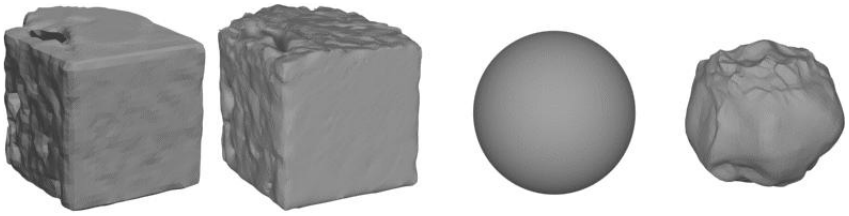
Fua and Leclerc [61] treated an initial triangulated scene model as a deformable surface and formulated a data energy term to minimise the variance in image intensity at the projection of a set of surface points into multiple images. For a sample point s on the model surface, the projected location in image m is defined as $\underline{u}_{sm}$ and the image intensity given by $I(\underline{u}_{sm})$. The data energy term of the model then minimises the squared difference in the image intensity at each projected point from the mean intensity $\overline{I}_s$ across all images.

$$\mathcal{E}_{\mathcal{D}} = \frac{1}{N_s} \sum_{s=1}^{N_s} \frac{1}{N_m} \sum_{m=1}^{N_m} \left(I(\underline{u}_{sm}) - \overline{I}_s \right)^2 \tag{5.5}$$

The sample points are defined in terms of the vertex locations of the model $\underline{x}_s(\underline{x}_i)$. The derivative of the data term with respect to the model vertices can then be derived to give the deformation of the model to minimise the energy function.

$$\frac{d\mathcal{E}_{\mathcal{D}}}{d\underline{x}_i} = \frac{1}{N_s} \sum_{s=1}^{N_s} \frac{1}{N_m} \sum_{m=1}^{N_m} \left(\mathbf{J}^T(I(\underline{u}_{sm})) - \mathbf{J}^T(\overline{I}_s) \right) \left(I(\underline{u}_{sm}) - \overline{I}_s \right) \tag{5.6}$$

The gradient of the data energy function gives the deformation of the model vertices to match the intensity between a set of images at a set of sample points on a model surface. The technique has the advantage that the prior model can be used to detect self-occlusion in the images by testing the surface visibility. The technique also avoids the assumption of fronto-parallel surfaces allowing for non-constant depth by performing correlation across the model surface. There are however two distinct problems. The first is the assumption of a Lambertian reflectance model such that a sample point has an equal intensity in each image. A normalised cross-correlation could be considered to account for specular highlights under a dichromatic reflectance model. The second assumption is that the model is close to the correct solution. The model is driven by the derivative of the image intensity with respect to the 3D location of the surface, $\mathbf{J}(I(\underline{u}_{sm}))$.



(a) Initial model (b) Local optimisation (c) Initial model (d) Local optimisation

Figure 5.5: Model optimisation to match image intensity using the technique proposed by Fua and Leclerc [61] for the five simulated images shown in Figure 5.1.

The problem of a close initial surface becomes significant in optimising a prior human model to match multiple images where the model can initially be far from the correct solution. Fua and Leclerc note that the gradient of the energy function becomes meaningless if the image plane distance is greater than a few image pixels [61]. If this assumption is violated then the initial surface converges to a local minimum. Figure 5.5 shows the implementation of the technique proposed by Fua and Leclerc [61] for the ideal case of reconstructing the cube shown in Figure 5.1 where intensity is consistent between views. The technique refines the mesh reconstructed from multiple view stereo, but fails with an initial model that does not lie close to the correct solution. The technique suffers from local minima in optimisation as illustrated in Figure 5.6. Figures 5.6 (a) and (b) show two rectified camera images for a stereo pair. Figure 5.6(c) shows the normalised cross-correlation score at every pixel in Figure 5.6(b) for the image window outlined in Figure 5.6(a). There is a local maxima at the correct pixel match indicated by the window shown in Figure 5.6(b). However, the correlation function has many local maxima and the area of convergence extends for only a few pixels around the correct match as shown in Figure 5.6(d).

An object-centred approach to stereo provides the advantage of incorporating prior shape information from a model to avoid matching between images in the presence of self-occlusions. However, the local optimisation technique proposed by Fua and Leclerc [61] suffers from local minima in optimisation.

(a) Left image (b) Right image (c) Correlation score (d) Convergence

Figure 5.6: The normalised cross-correlation score at every pixel in the right camera image for a (13×13) image window from the left camera image. Showing the matching image points marked in the left and right images and the region of convergence surrounding the local maximum.

5.2.2 A direct search for correspondence

In this work a model-based approach to multiple view stereo is introduced in which a model is optimised to match the appearance across multiple views with a direct search for stereo correspondence. A direct search can overcome the problem of local minima in recovering stereo correspondence to give a wider range of convergence.

An initial model is treated as a deformable surface and a data energy term is designed to maximise the stereo correlation. This data term is defined at the vertices of the model in order to recover the final correspondence for the vertices in each camera image. For each vertex $\underline{x}_i$, the camera with the closest viewpoint is selected such that the surface can be assumed to be locally fronto-parallel to the camera view. Here the closest camera, termed the key camera, is selected according to the viewpoint closest to the direction of the vertex normal on the surface mesh. Stereo matches are then located by a direct search in each adjacent camera forming a stereo pair with the key view, termed an offset camera. The search space is constrained along the epipolar line in each rectified offset image. A stereo match is then defined at the point with the maximum normalised cross-correlation with the rectified key image. A sub-pixel accurate match can be located by fitting a quadratic parabola to the correlation scores as described in Equation 5.4. From the matched image position in each offset image m a 3D data point can be reconstructed $\underline{z}_{im}$ for the model vertex. The mean squared error to the

reconstructed vertex locations is then minimised across the model.

$$\mathcal{E}_{\mathcal{D}} = \sum_{i=1}^{N_i} \frac{1}{\sum_m v_{im}} \sum_{m=1}^{N_m} v_{im} \|\underline{z}_{im} - \underline{x}_i\|^2 \tag{5.7}$$

A visibility term v_{im} is used here to define where a vertex is both visible in the image m and a stereo correspondence is derived for the image with respect to the key camera view selected for the vertex. The visibility of the model vertices in the camera images is determined using the visibility algorithm proposed by Debevec et al. [47]. The mesh is rendered to a camera viewpoint using hardware accelerated OpenGL rendering with the pinhole camera model. A unique colour ID is assigned to each triangle in rendering and depth-buffering is enabled so that occluded triangles are not rendered. The visibility of a vertex is then determined in a camera view by establishing the triangle at the image plane projection from the colour in the rendered image and testing whether the triangle occludes the vertex. The occlusion test is performed in the 2D image coordinates termed the screen-space. If the triangle encloses the projected vertex location then the vertex is occluded, $v_{im} = 0$, otherwise $v_{im} = 1$.

The visibility algorithm [47] derives visibility based on the estimated geometry of the model. The potential drawback is that this can lead to incorrect visibility estimates at occlusion boundaries. Where the geometry of the model is not correct at a boundary the occluded vertices on more distant surfaces can be incorrectly classed as visible. To overcome this problem a conservative visibility check is used by also testing for occlusion against the geometry of the visual-hull. The assumption is made that where a vertex is classed as visible, $v_{im} = 1$, but is occluded by the back-facing surface of the visual-hull, then the vertex is in the region of an occlusion boundary and should be classed as occluded, $v_{im} = 0$. This conservative test is implemented simply by rendering the back-facing surface of the visual-hull with a separate colour ID on top of the rendered model.

The search for correspondence in each offset view for which a vertex is visible is illustrated in Figure 5.7. The search is performed along the epipolar line in the rectified offset image up to a specified pixel error perpendicular to each epipolar line defined according to the expected accuracy in camera calibration. The search along each epipo-

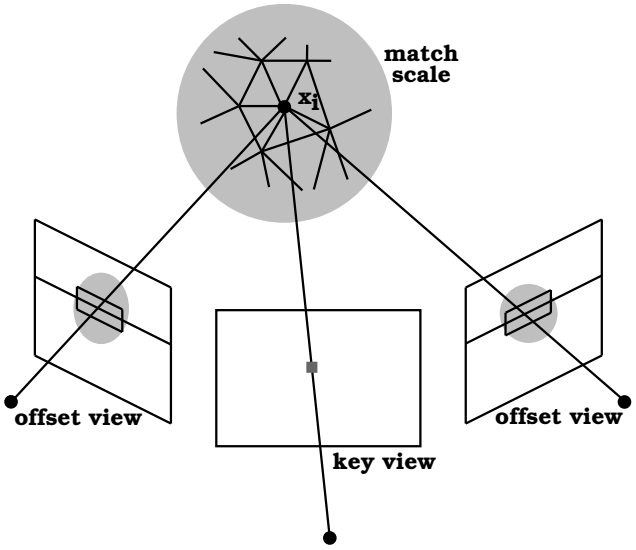


Figure 5.7: Direct search for stereo correspondence in offset camera images with respect to a key camera view. Showing the rectangular search region in a rectified offset image defined by the scale of matching along an epipolar line and the expected reprojection error perpendicular to an epipolar line.

lar line is limited to the expected error on the position of a vertex defined by a scale term $\sqrt{T}$. The horizontal 2D range $\triangle u_{im}$ along a rectified epipolar line either side of a projected horizontal location of a vertex u_{im} is derived from the projective transformation matrix $\mathbf{P}_{m'}$ to the rectified image m' given the 3D location of a vertex $\underline{x}_i$, and the normalised direction to the vertex from the key camera $\hat{\underline{n}}_i$. The search for correspondence is finally constrained to lie within the bounding constraint of the image silhouettes by clipping the search range against the volume of the visual-hull.

$$u_{im} = \frac{[P_{00}P_{01}P_{02}P_{03}]\left([\ \underline{x}_i^T \quad 1\ ]^T\right)}{[P_{20}P_{21}P_{22}P_{23}]\left([\ \underline{x}_i^T \quad 1\ ]^T\right)} \tag{5.8}$$

$$\triangle u_{im} = \frac{[P_{00}P_{01}P_{02}P_{03}]\left([\ \underline{x}_i^T \quad 1\ ]^T \pm \sqrt{T}[\ \hat{\underline{n}}_i^T \quad 0\ ]^T\right)}{[P_{20}P_{21}P_{22}P_{23}]\left([\ \underline{x}_i^T \quad 1\ ]^T \pm \sqrt{T}[\ \hat{\underline{n}}_i^T \quad 0\ ]^T\right)} - u_{im} \tag{5.9}$$

(a) Key image (b) Offset image (c) Correlation score (d) Local maxima

Figure 5.8: The normalised cross-correlation score along each epipolar line in a rectified offset image for the column of pixels marked in a rectified key image using a (13×13) image window. Showing the peak scores in red and in blue all local maxima within a tolerance $\tau = 0.9$ of each peak.

5.2.3 Multiple point stereo matching

So far only a single stereo match has been considered for each camera view. In practice many matches may be found along an epipolar line. Figure 5.8 shows the correlation scores along the epipolar lines in a rectified offset image for a column of pixels in a rectified key image. The peak correlation scores corresponding to the true surface shape can be seen, however it is evident that there are many local maxima along each epipolar line. Following the multiple point matching scheme introduced in Chapter 4 a multiple point stereo scheme is proposed in which the vertices for a model are matched to multiple reconstructed points $\underline{z}_{imk}$ corresponding to each local maxima k in the correlation score for the offset image m. A point $\underline{z}_{imk}$ is reconstructed for each local maxima where the correlation score lies within a set tolerance τ of the maximum score over the search range. Figure 5.8(d) shows the local maxima located with a tolerance of $\tau = 0.9$ and demonstrates that the target local maxima on an epipolar line does not necessarily have the peak correlation score.

An assignment parameter, $0 \leq w_{imk} \leq 1$, is introduced to define the degree of assignment of a vertex $\underline{x}_i$ to the multiple reconstructed 3D points $\underline{z}_{imk}$. An entropy term is also introduced to control the scale of matching given in the assignment as described

in Chapter 4. The data energy for multiple point stereo matching is then defined as.

$$\begin{aligned}\mathcal{E}_{\mathcal{D}} = & \sum_{i=1}^{N_i}\sum_{m=1}^{N_m} v_{im}\sum_{k=1}^{N_{imk}} w_{imk}\|\underline{z}_{imk} - \underline{x}_i\|^2 \\ & + T\sum_{i=1}^{N_i}\sum_{m=1}^{N_m} v_{im}\sum_{k=1}^{N_{imk}} w_{imk}\left(\log(w_{imk}) - 1\right)\end{aligned} \tag{5.10}$$

The assignment parameter that minimises the energy can be derived for a fixed model configuration. The set of matches for each vertex are independent and the constraint, $0 \leq w_{imk} \leq 1$, can be satisfied by simply normalising with respect to the total assignment value across all points for a vertex.

$$\frac{d\mathcal{E}_{\mathcal{D}}}{dw_{imk}} = \|\underline{z}_{imk} - \underline{x}_i\|^2 + T\log(w_{imk}) = 0 \tag{5.11}$$

$$w_{imk} = \exp\left(-\frac{\|\underline{z}_{imk} - \underline{x}_i\|^2}{T}\right) \tag{5.12}$$

$$w_{imk} = \frac{w_{imk}}{\sum_m \sum_k w_{imk}} \tag{5.13}$$

5.2.4 Combining stereo and silhouette data

Stereo matching is performed for every vertex defining the surface shape of a model. It is clear however that stereo correspondence will fail in regions where there is a limited variation in the local image intensity. The silhouette data is therefore incorporated where stereo matching is expected to fail. The data energy term for model optimisation is formulated to incorporate both the stereo and silhouette data terms with a per vertex weighting, $0 \leq \mu_i \leq 1$, defining the trade-off between the two complementary shape cues.

$$\begin{aligned}\mathcal{E}_{\mathcal{D}} = & \sum_{i=1}^{N_i}\mu_i\sum_{m=1}^{N_m} v_{im}\sum_{k=1}^{N_{imk}} w_{imk}\|\underline{z}_{imk} - \underline{x}_i\|^2 \\ & + \sum_{i=1}^{N_i}(1-\mu_i)\sum_{j=1}^{N_j} w_{ij}\|\underline{y}_j - \underline{x}_i\|^2 \\ & + T\sum_{i=1}^{N_i}\mu_i\sum_{m=1}^{N_m} v_{im}\sum_{k=1}^{N_{imk}} w_{imk}\left(\log(w_{imk}) - 1\right) \\ & + T\sum_{i=1}^{N_i}(1-\mu_i)\sum_{j=1}^{N_j} w_{ij}\left(\log(w_{ij}) - 1\right)\end{aligned} \tag{5.14}$$

The relative weighting μ_i represents a confidence measure in the stereo correspondence recovered at a vertex with respect to a key camera view. The correlation score at

a stereo match could be considered as a confidence score, however a high correlation can be obtained in matching regions of low image texture where the correspondence is ambiguous. The confidence measure is instead defined according to the ability to recover good correspondence where there is a high degree of intensity variation in the key camera view. A simple measure is defined in which the standard deviation σ_i of the pixel intensities at the projected vertex location $\underline{u}_i$ in the key camera view is used.

$$\sigma_i = \sqrt{\sum_{\underline{w}} \left(I(\underline{u}_i + \underline{w}) - \overline{I}(\underline{u}_i) \right)^2} \tag{5.15}$$

The standard deviation σ_i measures the degree of image texture within the key image window for stereo matching with a normalised cross-correlation score. This score is converted to the weight μ_i with a transformation based on a predefined point $\sigma_{0.5}$ where equal weighting is given to both the silhouette and stereo data $\mu_i = 0.5$. Figure 5.9 shows the weighting term derived for two rectified key images with $\sigma_{0.5} = 10$, demonstrating that the weight is reduced in regions of low image texture where stereo correlation will fail.

$$\mu_i = 1 - \exp\left(\frac{\sigma_i^2 \times \log(0.5)}{\sigma_{0.5}^2} \right) \tag{5.16}$$

5.2.5 Summary

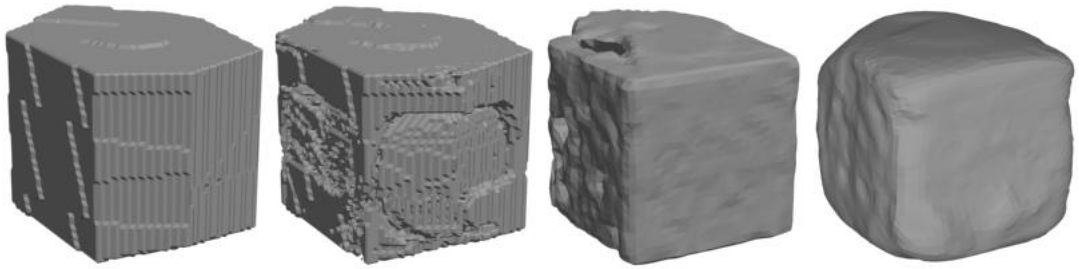
The model-based framework for reconstruction has been developed to incorporate both silhouette and stereo data in shape recovery. Reconstruction with a shape constrained deformable model is now shown in Figure 5.10 for the problem of reconstructing the cube presented previously in Figure 5.1. The technique recovers an accurate estimate of shape where there is sufficient variation in appearance for stereo matching and the shape from multiple view silhouettes is satisfied in fitting the visual-hull where the appearance cannot be matched between views. Stereo matches are obtained in the technique using a coarse-to-fine approach in which the search range for correspondence is controlled by the scale term $\sqrt{T}$ used in deterministic annealing. The temperature scale T is gradually reduced as the model deforms to fit the data and the stereo matches are

(a) Key image (b) μ_i image (c) Key image (d) μ_i image

Figure 5.9: The weighting score for stereo matching shown across two key images with a (13×13) image window.

refined. This coarse-to-fine search for correspondence allows the model to be initially far from the correct solution in optimisation and refines the stereo matches as the model converges to a solution.

The drawback for the technique lies in the limited capture resolution required to obtain whole-body images of a person. With a limited resolution, the fine geometric detail on the body cannot be recovered with stereo matching. The problem arises due to the trade-off between camera resolution and camera baseline for stereo reconstruction. A baseline in the order of $1m$ is required for PAL resolution images of a person to obtain a reconstruction accuracy in the order of $1mm$ with sub-pixel accurate matches in the order of 0.1 pixels. With a $1m$ camera baseline there can be a significant change in the appearance of a person between camera images leading to incorrect stereo matches. This proves to be a particular problem in reconstructing the shape of the face especially around the nose where there is a combination of distortion and occlusion between views. To solve this problem the framework is augmented with sparse feature matching to define the geometric detail at the face that cannot be recovered using stereo.



(a) Visual-hull (b) Photo-hull (c) Merged stereo (b) Model optimisation

Figure 5.10: Comparison of the visual-hull, photo-hull, and multiple-view stereo with the model-based technique in optimising a sphere to match stereo and silhouette data.

5.3 Surface Feature Matching

In this section, sparse feature matching is introduced to provide control for user defined model correspondence in the images. The deformable model framework is designed to incorporate a data-fitting term to match a model to both silhouette, stereo and feature data. Explicit feature matching is used specifically to define the fine geometric detail of the human face where stereo matching fails in the PAL resolution whole-body images.

5.3.1 Model-based feature matching

The model-based framework for human shape reconstruction uses a generic humanoid model registered to match the pose of a person in multiple camera images. In Appendix B a user interface is used to reconstruct a set of 3D feature points corresponding to the skeletal animation structure of the model and a model is posed to match the features. This interface is also applied to define the image correspondence for a set of surface features on the model and reconstruct the corresponding 3D feature locations. The model-based reconstruction framework should then satisfy these sparse constraints on the shape of the model.

The sparse set of observed features $\underline{o}_c$ provide a constraint on the 3D location of the corresponding model vertices $\underline{x}_c$. Applied individually the constraints would impose only a sparse, localised constraint on the shape of the surface mesh. The constraints

are therefore interpolated across the mesh to provide a global constraint. Sparse data interpolation is used to derive the constraints for the complete set of model vertices, $\underline{x}_i$. An error term $\underline{e}_i$ is introduced at the model vertices and the error term is interpolated using radial basis functions. The 3D thin-plate radial basis function $\mathcal{R}_c(\underline{x}) = \|\underline{x} - \underline{x}_c\|^3$ is used as described in Section 3.3.2, to provide a globally smooth interpolation of the error in 3D [174] with an additional affine basis $\mathbf{A}$ to account for global error terms.

$$\underline{e}_i = \sum_{c=1}^{N_c} \underline{\eta}_c \mathcal{R}_c(\underline{x}_i) + \mathbf{A} \begin{bmatrix} \underline{x}_i^T & 1 \end{bmatrix}^T \tag{5.17}$$

The parameters $(\underline{\eta}_c, \mathbf{A})$ for interpolation are derived from a linear system of equations in terms of the known errors $\underline{e}_c$ at the defined feature points, with an additional set of constraints that remove the affine contribution from the radial basis functions.

$$\underline{e}_c = (\underline{o}_c - \underline{x}_c) \tag{5.18}$$

$$\sum_{c=1}^{N_c} \underline{\eta}_c = \underline{0} \tag{5.19}$$

$$\sum_{c=1}^{N_c} \underline{\eta}_c^T \underline{x}_c = 0 \tag{5.20}$$

The parameters $(\underline{\eta}_c, A)$ are computed once only given the initial model shape and the set of sparse feature observations. The error to the feature points can then be interpolated to derive an observations for every model vertex, $\underline{o}_i$.

$$\underline{o}_i = \underline{x}_i + \underline{e}_i(\underline{\eta}, \mathbf{A}) \tag{5.21}$$

The optimisation framework for the deformable model is now formulated to incorporate a data term for sparse feature matching in which the mean squared error to the interpolated observations, $\|\underline{o}_i - \underline{x}_i\|^2$, is minimised.

$$\begin{aligned}\mathcal{E}_{\mathcal{D}} = \; & \textstyle\sum_{i=1}^{N_i} \nu_i \|\underline{o}_i - \underline{x}_i\|^2 \\ & + \textstyle\sum_{i=1}^{N_i} (1-\nu_i)\mu_i \sum_{m=1}^{N_m} v_{im} \sum_{k=1}^{N_{imk}} w_{imk} \|\underline{z}_{imk} - \underline{x}_i\|^2 \\ & + \textstyle\sum_{i=1}^{N_i} (1-\nu_i)(1-\mu_i) \sum_{j=1}^{N_j} w_{ij} \|\underline{y}_j - \underline{x}_i\|^2 \\ & + T \textstyle\sum_{i=1}^{N_i} (1-\nu_i)\mu_i \sum_{m=1}^{N_m} v_{im} \sum_{k=1}^{N_{imk}} w_{imk} \left(\log(w_{imk}) - 1\right) \\ & + T \textstyle\sum_{i=1}^{N_i} (1-\nu_i)(1-\mu_i) \sum_{j=1}^{N_j} w_{ij} \left(\log(w_{ij}) - 1\right)\end{aligned} \tag{5.22}$$

A weight ν_i is introduced to define the influence of the interpolated feature constraints at each vertex. The parameter ν_i defines the trade-off between matching specified feature data and the recovered data from silhouette and stereo. The framework deals with the general case were features are specified at arbitrary vertex locations and a global interpolation is performed to derive constraints across the whole surface of the model. In practice however, feature matches are used to constrain only a subsection of the surface such as the face of the model. A global interpolation can then provide incorrect feature matches at distant points on the model. The influence of the feature matching at each vertex $\underline{x}_i$ is therefore controlled according to the distance to the constrained vertices $\underline{x}_c$ across the surface of the model. The surface distance d_i at a vertex is measured as the minimum edge-connected distance between the vertex and a feature vertex. The region of influence is defined again according a control parameter $d_{0.5}$ giving the distance were the influence is reduced to half. The weighting is then defined as follows.

$$\nu_i = \exp\left(\frac{d_i^2 \times \log(0.5)}{d_{0.5}^2}\right) \tag{5.23}$$

5.4 Model-Based Reconstruction of Appearance

The final framework to optimise a model to the appearance in multiple views with a shape constrained deformable model is outlined in the algorithm *optimise_model*. The control temperature T now defines the scale of matching to both the discrete set of surface points $\underline{y}_j$ in the visual-hull and the recovered set of stereo matches $\underline{z}_{imk}$. The temperature scale also defines the search region for the recovery of stereo matches such that the multiple recovered matches are refined as the surface converges to a solution.

Initially the temperature is set at the expected error in the shape of the model and the final temperature is set equivalent to the estimated reprojection error for the cameras so that stereo matches can be recovered in the presence of inexact camera calibration.

Input: Model, $\underline{x}_i$
Camera parameters, $\mathbf{P}_m$
Stereo confidence images, μ_m
Temperature, T

Output: Stereo points, z_{imk}
Image correspondence, $\underline{u}_{imk}$
Assignment, w_{imk}
Confidence, μ_i

Procedure: *set_stereo_assignment*

1. for (*each image m*)
2. render (*model to camera view*)
3. render (*back-faces of visual-hull to camera view*)
4. set (*vertex visibility* v_{im})
5. for (*each vertex i*)
6. set (*key image as closest visible view*)
7. read (*stereo confidence* μ_i *from key image* μ_m)
8. for (*each offset image*)
9. set (*search range in rectified offset image*)
10. clip (*search range against visual-hull*)
11. compute (*all correlation scores in range*)
12. set (*stereo matches* z_{imk}*, correspondence* u_{imk})
13. set ($w_{imk} = \exp(-\|\underline{z}_{imk} - \underline{x}_i\|^2/T)$)
14. *normalise*(w_{imk})

The framework provides the correspondence for each vertex of a model in each visible image for the recovery of colour texture from the images. The correspondence in a visible image is defined as the final sub-pixel accurate correspondence recovered in stereo matching as given by the matched image point with the highest correlation score in each image. The technique therefore provides sub-pixel accurate image correspondence for subsequent recovery of model texture.

Input: Model, $\underline{x}_i$
Camera parameters, $\mathbf{P}_m$
Images and silhouettes, $\underline{I}_m, I_m$

Output: Updated model, $\underline{x}_i$
Image correspondence, $\underline{u}_{im}$

Procedure: *optimise_model*

1. *reconstruct_visual_hull*($\underline{y}_j$)
2. *construct_octree*($\underline{y}_j$)
3. set (*local coordinates* $\alpha^0_{if}, \beta^0_{if}, h^0_{if}$)
4. for (*each image* m)
5. set (*stereo confidence* μ *at each pixel*)
6. set (*interpolated features* $\underline{o}_i$)
7. for (*each vertex* i)
8. set (*feature weighting* ν_i)
9. set (*temperature* $T = T_{init}$)
10. while ($T > T_{final}$)
11. *set_voxel_assignment*(w_{ij})
12. *set_stereo_assignment*($z_{imk}, u_{imk}, w_{imk}$)
13. while ($\|\frac{d\underline{x}_i}{dt}\| \geq \sqrt{cT}$ and $\triangle E < 0$)
14. set ($\frac{d\underline{x}_i}{dt} = -(\nabla \mathcal{E}_\mathcal{D} + \lambda \nabla \mathcal{E}_\mathcal{R})$)
15. set ($\underline{x}_i = \underline{x}_i + \delta \frac{d\underline{x}_i}{dt}$)
16. set ($T = c \times T$)
17. for (*each vertex* i)
18. for (*each image* m)
19. set ($u_{im} = u_{imk}$ *with peak correlation*)

The cost of model optimisation is now increased compared to the algorithm for shape from silhouette presented in Section 4.4 with the requirement to search for stereo correspondence at each temperature scale T. The stereo correlation must be computed in each offset image that forms a stereo pair with the key image for each vertex. If N_o is the average number of offset cameras and N_p is the average number of pixels in an offset camera for which the correlation is computed then the complexity of the search for N_i vertices is $O(N_i N_o N_p)$. Optimisation of the model with gradient descent now matches each vertex to the surface points on the visual-hull and the reconstructed stereo points. As the number of surface points N_v used is in general larger than the number of recovered stereo matches, the complexity of optimisation with gradient descent is again $(N_i N_v N_\nabla)$ as defined in Section 4.4. The total complexity of the algorithm is therefore

governed either by the total number of pixels in stereo matching, $O\left(N_i N_o N_p N_{anneal}\right)$, or the number of matched points in gradient descent, $O\left(N_i N_v N_\nabla N_{anneal}\right)$. Optimisation will dominate if the search range for matches defined by the initial temperature T_{init} is small or a large number of surface points N_v are matched with a small step-length in gradient descent. Conversely, stereo matching will dominate if T_{init} is large or if a small number of surface points are matched with a large step length.

5.5 Evaluation

The model-based technique combines the shape information from multiple view silhouettes, stereo correspondence and sparse features to match the multiple view appearance of a person. The effect of each shape cue is evaluated along with the accuracy of reconstruction. In Section 5.5.1 optimisation of a model to fit stereo and silhouette data is assessed. The trade-off between the shape cues and the use of multiple point matching in recovering stereo correspondence is examined. In Section 5.5.2 the effect of introducing sparse feature matches to define the detailed geometry of the face is evaluated. Finally the technique is demonstrated for real images captured in a multiple camera studio in Section 5.5.3 and the reconstructed models are compared to shape reconstruction from the visual-hull, photo-hull and multiple-view stereo.

The ideal test case presented in Chapters 3 and 4 is used to assess reconstruction accuracy with the control parameters $\sigma_{0.5}$, τ and $d_{0.5}$. The remaining parameters for the deformable model are set as outlined in Section 4.5 ($\lambda = 3.0$, $c = 0.5$, $T_{init} \equiv 10cm$, $N_v = 100$, $\delta = 0.1$). Model deformation is terminated at the target reconstruction accuracy for the real camera calibration, $T_{final} \equiv 5mm$. Stereo correlation is derived using a (13×13) window and a 1 pixel reprojection error is assumed in the rectified camera images. The visual appearance in the ideal case is simulated by rendering the range data to the camera images with a texture map corresponding to four different subjects wearing a range of clothing as shown in Figure 5.11. It should be noted that these images were generated first by resampling real camera images to a texture map, then by resampling the texture map to a new camera image from the range data. Resampling reduces the fine detail in the images and the performance of stereo

(a) Subject A (b) Subject B (a) Subject C (b) Subject D

Figure 5.11: Range data rendered to an ideal camera image with a texture map corresponding to four different subjects A, B, C and D.

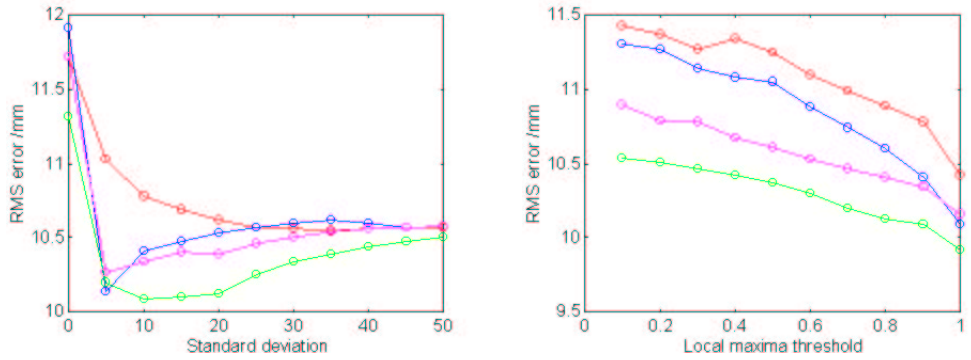
correspondence is expected to be degraded as a result.

5.5.1 Matching silhouette and stereo data

The reconstruction accuracy in fitting both silhouette and stereo data is now shown in Figure 5.12(a) for the ideal test case. It can be seen that the parameter $\sigma_{0.5}$ defines a trade-off between matching stereo data ($\sigma_{0.5} \rightarrow 0$) and matching the visual-hull ($\sigma_{0.5} > 50$). The parameter $\sigma_{0.5}$ controls the influence of stereo matching in regions of uniform appearance. A higher value of $\sigma_{0.5}$ restricts stereo matching to regions with a greater variation in appearance. Using stereo data alone leads to a higher reconstruction error as clothing tends to have large areas of uniform appearance as demonstrated in Figure 5.11. In fact for subject A in Figure 5.11(a), there is only a limited variation in appearance and the introduction of stereo data provides no improvement over the shape derived from the visual-hull. For the remaining subjects the stereo data does improve the reconstruction accuracy. A value of $\sigma_{0.5} = 10.0$ is selected where the reconstruction accuracy becomes relatively stable to the exact parameter value chosen as shown in 5.12(a).

Conclusion:

1. **The reconstruction error in fitting a deformable model to silhouette data is reduced using stereo correspondence to derive accurate esti-**



(a) RMS error vs. $\sigma_{0.5}$ ($\tau = 0.9$) (b) RMS error vs. τ ($\sigma_{0.5} = 10.0$)

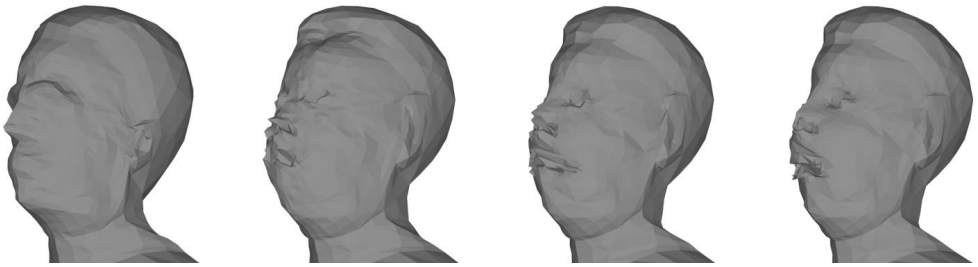
Figure 5.12: The RMS error from the deformed model to the range data with (a) the standard deviation $\sigma_{0.5}$ defining the influence of stereo matches and (b) the threshold τ defining the influence of multiple stereo matches, for subject A (red) B (blue) C (green) and D (magenta).

mates of surface position; and

2. **Stereo matching fails in regions of uniform appearance in the camera images and trade-off is required to define the relative influence of stereo matching in model deformation.**

The reconstruction accuracy with multiple-point stereo correspondence is shown in Figure 5.12(b). The parameter τ defines the threshold on the local maxima in the correlation score for a point correspondence. With a low value of τ a larger number of local maxima will be matched. Figure 5.12(b) demonstrates that matching multiple points increases the error in the shape of the final model. The stereo correspondence problem has one solution and additional local maxima in the correlation score represent incorrect point estimates. The multiple-point matching scheme aimed to allow a model to fit multiple potentially correct points and recover the correct correspondence as the scale of matching is refined. However, additional local maxima are in effect noisy estimates of the true vertex location and considering additional correspondences will corrupt the estimated location of a vertex in the weighted point match in Equation 5.11. Figure 5.12(b) demonstrates that this in turn leads to a higher reconstruction error.

(a) Front views of the face for stereo matching.



(b) $d_{0.5} = 5cm$, $\tau = 1.0$ (c) $\tau = 1.0$ (d) $\tau = 0.8$ (e) $\tau = 0.6$

Figure 5.13: The reconstructed shape of the face for multiple point stereo correspondence without sparse feature matching in comparison with the shape derived with sparse feature matching.

The performance of multiple point stereo matching is demonstrated in Figure 5.13 for a real test case using the three camera views shown in Figure 5.13(a) to form two stereo pairs. It can be seen that the multiple point approach provides no subjective improvement over recovering a single point correspondence between stereo pairs ($\tau = 1.0$). Multiple point matching in stereo correspondence is therefore neglected using a value of $\tau = 1.0$ to recover only a single correspondence between in images where the peak correlation score is obtained.

Conclusion:

3. **Multiple point stereo matching does not improve the recovered model shape.**

5.5.2 Matching feature data

The reconstruction error in using sparse feature matching for the ideal test case is shown in Figure 5.14. The parameter $d_{0.5}$ defines the region of influence for a sparse

feature along the surface of a mesh. Figure 5.14 shows that as this distance is increased the reconstruction error increases. However, with a low value for $d_{0.5}$ the features only provide a sparse constraint on shape, the face can then become corrupted with noisy stereo estimates due to the limited resolution images. Figure 5.15 shows the surface shape obtained for the subject in Figure 5.13(a) with different values for the parameter $d_{0.5}$. A value of $d_{0.5} = 5cm$ is chosen in this work to provide a reasonable range of interpolation for sparse matches and to minimise the potential reconstruction error.

Conclusion:

4. **Sparse feature matching provides a constraint on the detail shape of the face that cannot otherwise be recovered using stereo matching with limited resolution images; and**

5. **Interpolation of sparse feature constraints must be restricted to the vicinity of the constraint points to prevent incorrect interpolation to distant points on the surface.**

5.5.3 Real test cases

Optimisation of a generic humanoid model to match multiple view images of people is shown in Figures 5.17, 5.16 and 5.18 where the clothing worn provides a varying degree of image texture for stereo correlation. Nine camera views are used forming four stereo pairs. The technique demonstrates improved shape reconstruction compared to shape from silhouette and multiple view stereo. This improvement is obtained by using a prior model to define the expected shape for reconstruction and by combining complementary shape cues from silhouette, stereo and feature data.

The model-based technique provides accurate shape information where there is sufficient image texture to match appearance between images. Sub-pixel accurate image correspondence is achieved for the recovery of a model texture map. Figure 5.19 shows the correspondence now obtained for the problem initially considered in Section 4.5.3, demonstrating that the vertex locations for the model are now correctly matched between the camera images for the recovery of model texture.

Conclusion:

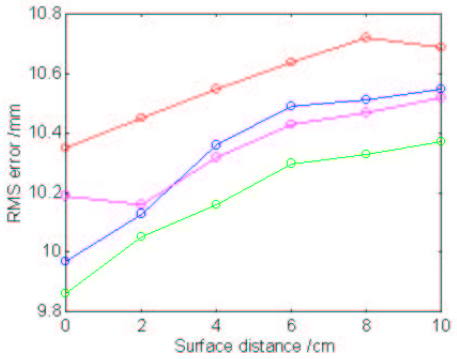
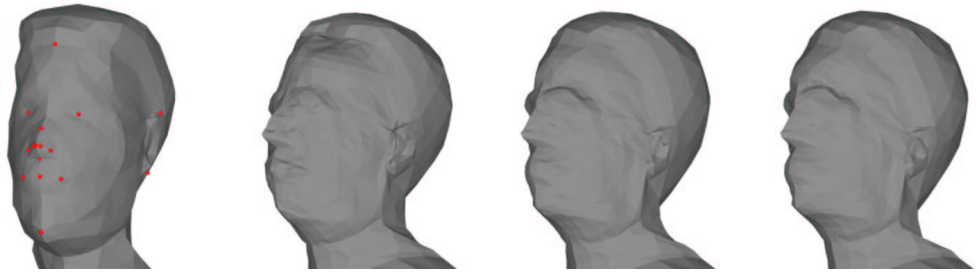
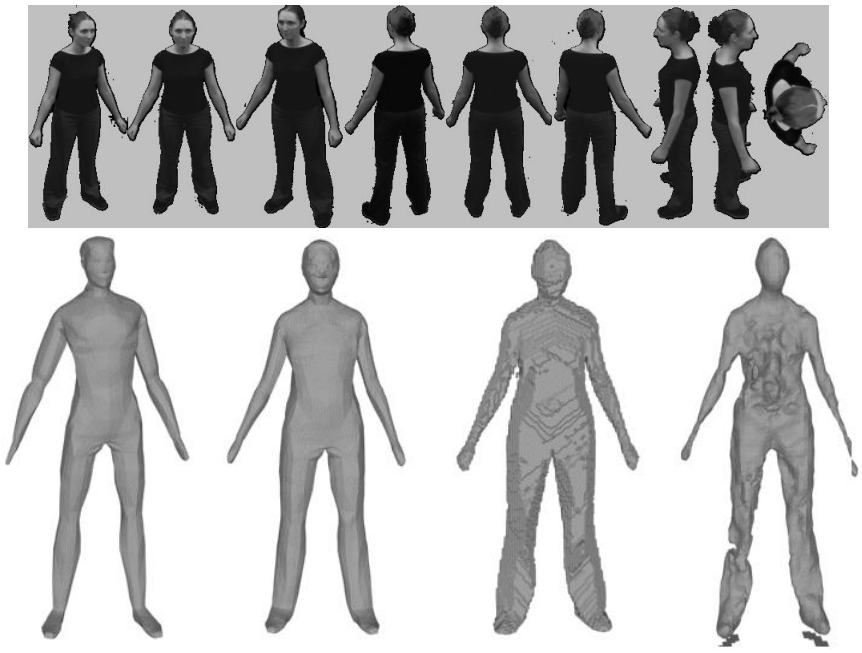


Figure 5.14: The RMS error from the deformed model to the range data with the distance $d_{0.5}$ defining the region of influence for sparse features, for subject A (red) B (blue) C (green) and D (magenta).



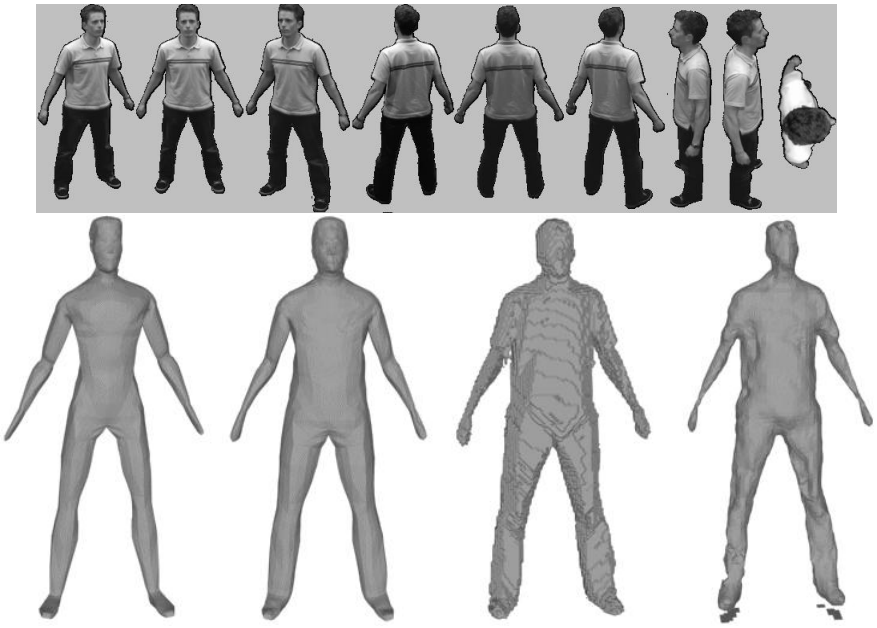
(a) Generic (b) $d = 2cm$ (c) $d_{0.5} = 4cm$ (c) $d_{0.5} = 6cm$

Figure 5.15: The influence of sparse feature matching in the reconstruction of the model face, showing the features points used on the generic model.



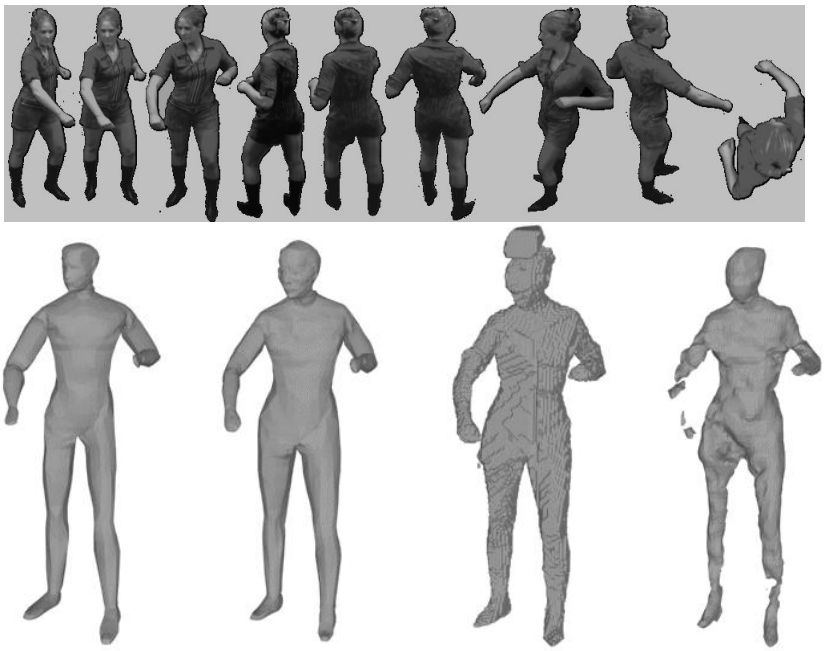
(a) Generic model (b) This work (c) Visual-hull (d) Merged stereo

Figure 5.16: Shape reconstruction with low clothing texture.



(a) Generic model (b) This work (c) Visual-hull (d) Merged stereo

Figure 5.17: Shape reconstruction with medium clothing texture.



(a) Generic model (b) This work (c) Visual-hull (d) Merged stereo

Figure 5.18: Shape reconstruction with highly textured clothing.

6. **Model-based reconstruction with stereo matching provides the image plane correspondence that matches the appearance of a person in multiple camera views for the recovery of model texture.**

5.6 Summary

In this chapter a model-based technique has been introduced to recover a model with the shape and appearance of a person. The model-based approach to reconstruction optimises a generic humanoid model to simultaneously match multiple shape cues. Multiple-view stereo is used to refine a model shape to match appearance between images. Silhouettes are used to provide a robust shape constraint in regions of uniform appearance where stereo matching fails. Image features are incorporated to provide a sparse set of constraints on the shape of the face where stereo matching fails in PAL resolution whole-body images. The technique provides improved shape reconstruction compared to model-free techniques such as *Voxel Coloring* [140] and multiple view stereo

Figure 5.19: Image plane correspondence recovered for 9 model vertices.

[123] in the presence of visual ambiguities. The reconstruction algorithm presented in this chapter introduces a novel model-based search for stereo correspondence. The model is optimised in a coarse-to-fine framework commencing at the expected error on the shape of the generic model and terminating at calibration accuracy of the camera system. Stereo matches are derived and refined as the shape-constrained model deforms to fit the data providing a greater range of convergence compared to previous local optimisation based techniques. The approach then provides sub-pixel accurate image correspondence for subsequent recovery of a model texture map.

Chapter 6

Multiple View Reconstruction of Appearance

The framework for model-based scene reconstruction presented in Chapter 5 addressed the problem of updating a prior humanoid model to match the shape and appearance of a person in multiple camera views. In this chapter the process of extracting the colour texture for the final model is presented. The surface colour of the model is essential to provide a visually realistic appearance in rendering. The aim is to achieve a "photo-realistic" model that reproduces the appearance of a person captured in the original camera images. The problem of deriving appearance has been simplified as the model geometry is optimised to match appearance across the multiple views. The input to the texture reconstruction process is the sub-pixel accurate image locations for the model vertices recovered from stereo matching in the model-based reconstruction algorithm. The problem is then to recover the colour across the model surface given this image correspondence.

Problem Statement:

- **Given the image plane correspondence for each model vertex, recover the colour at every point across the model surface that reproduces the appearance observed in the camera images.**

In this chapter both model reconstruction with a view-independent appearance and

image-based rendering with a view-dependent appearance are considered to define the surface colour of a model. In Section 6.1 a single view-independent texture map is constructed from the camera images and in Section 6.2 view-dependent rendering is presented, making use of the original camera images as multiple view-dependent textures. With a single texture map the model can be animated and displayed rapidly in a conventional computer graphics rendering pipeline. With view-dependent texturing on the other hand we can capture subtle changes in the reflected light-field across the surface of a model giving a potentially more realistic model appearance.

6.1 View-Independent Texture

In this section techniques are described to recover a single texture-map for a model from a set of camera images. Texture mapping is the process by which a colour texture image is mapped onto a surface of a 3D model in rendering. Every triangle in a model is parameterised with respect to the 2D coordinate system of an image with a 2D texture coordinate at each triangle vertex. The corresponding region of the texture image is then resampled onto the 3D triangle to define the surface colour in rendering. To recover a texture image for a model the inverse problem must be solved in which the view-independent colour must be derived across each 3D triangle from the camera images and resampled onto the 2D texture image. The texture image can then be used to render the model with the derived appearance.

The process of recovering a model texture is divided into the following tasks.

1. **Texture-map specification:** Specify the mapping between the 3D model surface and the 2D texture space;

2. **Triangle to image assignment:** Define the visibility of each triangle in the set of camera images;

3. **Image to texture resampling:** Resample each camera image onto the texture image from the visible surface of the model;

4. **Texture filling:** Synthesise missing sections of surface colour where triangles are not visible in any image; and

5. **Texture blending:** Blend the colour texture derived from different camera viewpoints to give a single view-independent appearance.

6.1.1 Texture-map specification

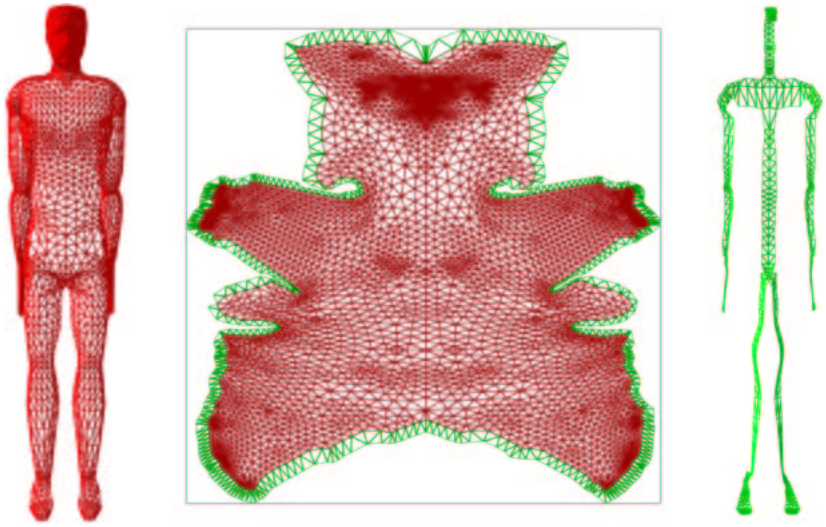
A texture map specifies the 2D mapping in a texture image for the surface triangles of a 3D mesh. The parameterisation of the 3D model space in the 2D texture domain should have several desirable properties: (i) the sampling of the model surface in texture space should be consistent; (ii) the texture-map should make an optimal use of the image space available; and (iii) the mapping should be continuous in the 2D domain. A simple approach is to treat each triangle individually and pack them separately into a single image, allowing uniform sampling and optimal use of the texture space. The drawback of the technique is that the mapping is not continuous, adjacent pixels in the texture image do not necessarily correspond to adjacent points on the surface of the model. This has two implications, firstly it is not possible to perform blending in the texture space between adjacent surface triangles textured from different camera images. Secondly, the texture image cannot be interpolated in rendering to perform anti-aliasing, a process termed texture mip-mapping. In order to perform blending and mip-mapping in the texture image, a continuous 3D to 2D mapping is required for a model such that adjacent points in texture space are adjacent on the 3D model surface.

For a general model shape, finding a continuous parameterisation of the 3D surface on a 2D domain is non-trivial. The problem has been addressed by segmenting a surface into local regions for which a continuous map can be constructed, termed a patch or chart, and packing the patches into a single texture image, termed an atlas [102]. This is analogous to the problem of defining a 2D map for the 3D surface of the world in cartography. It is not possible to define a continuous 3D to 2D mapping on the surface of a sphere such that each 3D point has a unique 2D point, instead the 3D surface must be split and unwrapped onto the 2D domain [130]. Here the surface of our generic humanoid model is converted to a 2D texture-map using the model “pelting” technique

introduced by Piponi et al. [130]. Model pelting mimicks the process by which animal hides are stretched to form a flat pelt and produces a single continuous texture patch for a model.

A polygonal model is pelted by introducing a split along a set of triangle edges such that the surface becomes topologically equivalent to a disk and stretching the model onto a 2D plane [130]. The split is defined manually and the surface is then treated as a deformable model with a set of spring forces imposed on the boundary vertices attracting the model to an external circular frame. The surface then deforms under the external constraints to form a flat disk. The vertex locations of the model can finally be projected onto a 2D plane to define 2D texture coordinates for the model. The 2D texture map obtained using the pelting technique described by Piponi et al. [130] is shown in red in Figure 6.1(b). As suggested by Piponi et al. [130] it was necessary to perform local refinements and manual modifications to improve the result of the mapping obtained from the pelting procedure. An additional pelting frame was introduced to flatten the shape of the face and obtain a greater texture sampling rate giving greater relative importance to the facial appearance. It was also necessary to flatten the surface under a strong set of external spring forces which resulted in a completely circular pelt. The external constraints were then relaxed to give a more intuitive shape in the texture map and to reduce the relative distortion of the triangular elements. Finally minor manual adjustments were required to correct triangles that were not completely flat. It should be noted that this operation only needs to be performed once per generic model.

The pelted texture map provides a single continuous mapping in the texture domain for the 3D surface of the humanoid model. The mapping does however contain a discontinuity at the boundary where the split was introduced in the model surface. At this boundary the continuity in the mapping is lost and texture values cannot be blended or interpolated. This discontinuity would result in a seam in the rendered model texture where the surface was split. To perform blending at this boundary, Piponi et al. [130] proposed the use of a second overlapping texture map for the region surrounding the seam. However, a separate texture map does not solve the problem of incorrect texture interpolation at the boundary in mip-mapping and does not allow for



(a) Generic humanoid model (b) Texture map (c) Triangles connected to the seam

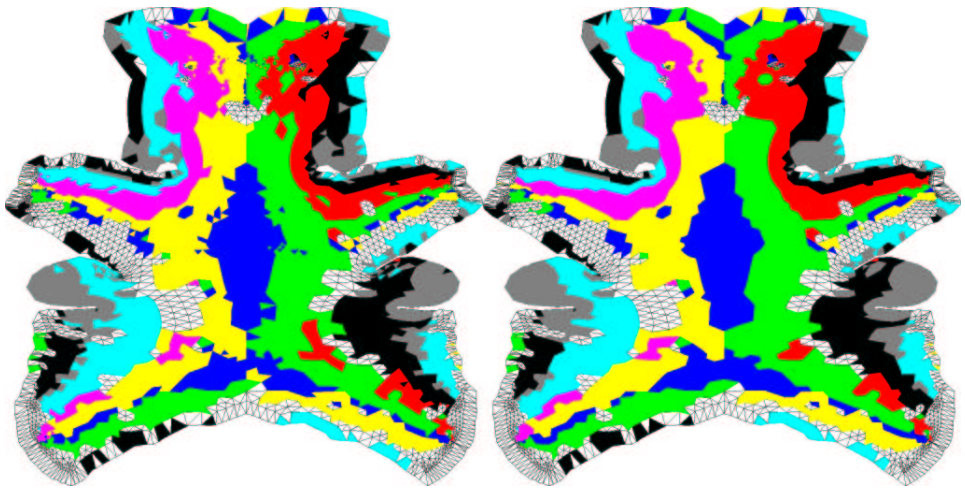
Figure 6.1: Texture map for the generic humanoid model showing the pelted texture region for the model in red with the duplicated texture region surrounding the pelt boundary in green.

blending in a single texture image as presented in Section 6.1.5.

The seam in the texture is removed by constructing a second texture surrounding the seam in the model and combining the texture with the model pelt to form a single texture map suitable for blending and texture mip-mapping. The set of model triangles connected to the triangle edges forming the seam in the model are duplicated in the second texture map and positioned to surround the corresponding boundary vertices in the pelted texture map. The second texture map then forms a continuous map as shown in green in Figure 6.1 that surrounds the boundary of the pelted texture, shown in red. Texture blending and interpolation can now be performed in a single texture image using the surounding duplicated texture.

6.1.2 Triangle to image assignment

The camera images are resampled onto the model texture map to provide the surface appearance in rendering. This resampling step as outlined in Section 6.1.3 is performed efficiently on a per-polygon basis rather than at every point on the model surface



(a) Target view assignment (b) Grouped target assignment

Figure 6.2: Colour coded assignment of triangles to target views showing (a) the initial target view assignment and (b) the grouped assignment minimising the boundary between the views. The assignment is missing where a triangle is occluded in all views.

individually. The goal of image assignment is firstly to define the valid set of images in which each model triangle is visible and the image plane locations of the triangle vertices so that the surface colour can be resampled from each camera image. The goal is then to define the target camera view for each triangle with the greatest resolution of the surface colour for the final texture image.

The input to the texture reconstruction process is the image plane locations for the model vertices recovered in stereo matching between the camera views. In views that do not form a stereo pair, the image plane location of the model vertices can be obtained by projection to the camera image plane. The fast hidden surface algorithm introduced in Section 5.2.2 is used as before to define the visibility of the model vertices in a camera view. We therefore have both the visibility v_{im} and the image correspondence $\underline{u}_{im}$ for each vertex i in image m. The valid set of images v_{fm} for each triangle facet f is defined as the subset of camera views in which all three triangle vertices are visible.

The target camera to resample texture is selected from the valid set of views according to the camera view most orthogonal to the surface to ensure the highest texture

resolution in the final texture image. Resolution is defined as the pixel sample rate in the camera image per unit surface area on the model, the foreshortening of the surface in projection to the camera image plane. If all cameras can be assumed to image the surface at the same scale then the sample rate is proportional to the cosine of the angle between the viewing direction and the surface normal of the triangle. The target image $\widetilde{v}_f$ for each triangle facet f is then selected as the valid view with the smallest viewing angle.

The target view for the model triangles can potentially form a highly discontinuous set in the texture image as shown in Figure 6.2(a). Distortions can arise at the boundaries between the texture derived from different views and so it is desireable to group the target views into larger contiguous sections in the texture domain. Here the patch growing technique presented by Niem and Broszio [125] is used to iteratively adjust the target view assignment of the triangles to minimise the number of triangle edges forming a boundary between different views in the texture map. The target views are grouped using the procedure *group_target_assignment* and the result shown in Figure 6.2(b).

Input: target view assignment, $\widetilde{v}_f$
valid view assignment, v_{fm}

Output: grouped target assignment, $\widetilde{v}_f$

Procedure: *group_target_assignment*

```
1. while (assignment updated)
2.     for (each triangle facet f)
3.         read (triangle assignment ṽ_f)
4.         read (edge-connected triangle assignments)
5.         if (two edges have a different assigned view ṽ'_f)
6.             if (ṽ'_f is a valid view)
7.                 set (ṽ_f = ṽ'_f)
```

6.1.3 Image to texture resampling

Once the triangles of a model are assigned to a valid set of cameras each camera image can be resampled to a texture map to give a texture with respect to the camera view. A straightforward approach to the resampling problem is to use ray-casting [129]. For

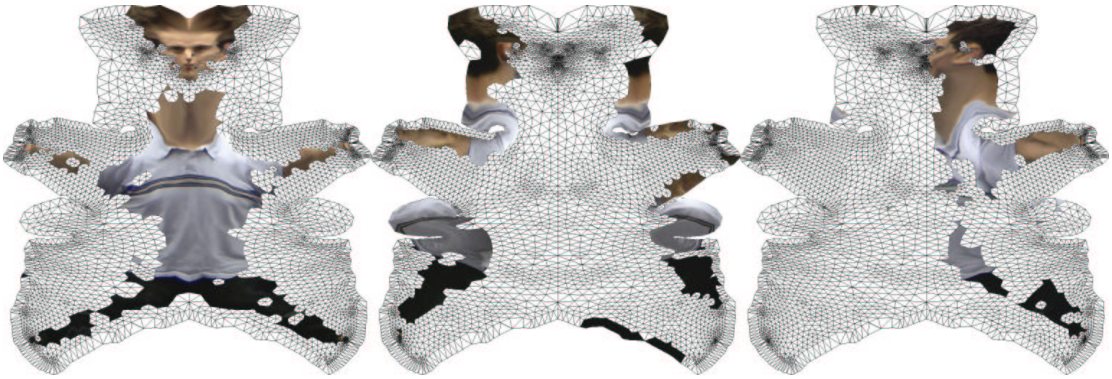


Figure 6.3: Texture resampled from three camera images. Texture is missing where each camera does not form a valid view for the corresponding model triangles.

each pixel in the texture map the corresponding triangle and surface point can be found on the model. The pixel colour can then be determined from the image plane correspondence of the triangle in the camera image. This resampling can be performed efficiently by making use of the hardware accelerated rendering pipeline to resample each model triangle rather than every texture pixel.

A camera image is resampled by converting a camera image to a texture map and using the image plane correspondence $\underline{u}_{im}$ to define the 2D texture coordinates in the camera image. The model is then rendered to an orthographic view using the vertex locations of the model pelt and each triangle textured from the camera image where the camera forms a valid view for the triangle. This rendering process resamples the surface colour in the camera view onto the plane of the model texture map, the algorithm is outlined in procedure *resample_texture*. Both the original and duplicated model triangles must be rendered to give both the pelted texture and the surrounding duplicated model texture. Figure 6.3 illustrates the texture map derived from three different camera images using the hardware accelerated resampling procedure.

It should be noted that a 2D texture image is mapped onto a 3D model with an affine transformation defined by the 2D texture coordinates for the vertices of each surface triangle. The proposed technique for resampling therefore performs a piece-wise affine transformation of a camera image to the model texture map. There is actually a perspective distortion in the projection of the model surface to the camera

image and a perspective transformation is required to correctly resample a camera view to the texture image. Perspectively correct texturing can be performed in hardware by making use of perspective texture coordinates defining a 3D texture coordinate with respect to a texture image [49]. However, the framework for shape recovery provides the image correspondence for the model vertices independent of the 3D vertex position to allow for inexact camera calibration. It is feasible to associate a corresponding depth value for perspective texturing, although in practise this is found to be unnecessary as the model triangles have a relatively small area and the distortion from an affine transformation within each triangle is not apparent.

Input: valid view assignment, v_{fm}
camera image, $\underline{I}_m$
image correspondence, $\underline{u}_{im}$
texture map specification, $\underline{u}_i^{TEX}$

Output: resampled texture image, $\underline{I}_m^{TEX}$

Procedure: *resample_texture*

1. load ($\underline{I}_m$ *as texture*)
2. set (*orthographic view for rendering*)
3. for (*each triangle facet* f)
4. if (*image is a valid view*)
5. set (*texture coordinates from* $\underline{u}_{im}$)
6. set (*vertex locations from* $\underline{u}_i^{TEX}$)
7. render (*triangle* f)
8. read (*rendered image* $\underline{I}_m^{TEX}$)

6.1.4 Texture filling

Texture filling is performed to provide a complete texture map for a model. It is feasible that some triangles on the model are occluded in all camera views and there will be no resampled texture in these regions. Texture is synthesised in these missing sections to fill the texture map and provide a complete model appearance.

Texture synthesis is performed using the texture derived from the target camera images. A single texture image is constructed to combine the texture from the target views and missing texture is interpolated from these visible regions. Each resampled texture image for a camera view is masked by the image region corresponding to the triangles

Figure 6.4: Four successive levels in a Gaussian image pyramid shown at the same image size to illustrate the decrease in image resolution and filling of missing image sections.

for which the view is the target. A target mask is constructed here by rendering the triangles that use a camera as the target view to the model texture map as described in Section 6.1.3 but by using a uniform model colour rather than the camera texture image. The resulting masks were shown previously in Figure 6.2 using a different colour for each view. A single texture image is constructed by combining the masked resampled texture images.

Interpolation in the texture image is performed using the "push-pull" algorithm proposed by Gortler et al. [68]. The algorithm first performs a "pull" phase in which a succession of lower resolution versions of the texture image are constructed where the visible regions of texture become more closely spaced. A "push" phase is then performed starting at the lowest resolution in which there are no missing sections of texture and filling in the missing texture pixels at each higher resolution image.

A Gaussian image pyramid [27] is used to provide a sequence of lower resolution images. Each image in the pyramid is convolved with a 5x5 Gaussian kernel, as described by Burt and Adelson [27], starting with the initial texture image . The result is a low-pass filtered version of each image in which the resolution is reduced by half at each step to give a pyramid of filtered images as shown in Figure 6.4. Missing sections of texture in the images are neglected in this convolution and the Gaussian kernel normalised to compensate. The missing texture pixels at each resolution are then filled starting at the penultimate resolution of the pyramid by bilinear interpolation of the pixel values from the preceeding lower resolution image. Figure 6.5 shows the missing sections of

(a) Combined resampled texture images (b) "Push-pull" interpolated image

Figure 6.5: The synthesised texture using "push-pull" interpolation with a Gaussian image pyramid.

the texture map filled using this "pull-push" process with a Gaussian image pyramid.

Input: image, $\underline{I}$

Output: filled image, $\underline{I}$

Procedure: *push_pull_interpolate*

1. *construct_Gaussian_image_pyramid*($\underline{I}$)
2. for (*Gaussian image* $\underline{I}_g$, $g = N_g - 2 \ldots 0$)
3. for (*each pixel* p *in* $\underline{I}_g$)
4. if (*pixel* p *empty*)
5. set ($p =$ *bi-linearly_interpolate*($\underline{I}_{g+1}, p$))
6. set ($\underline{I} = \underline{I}_0$)

6.1.5 Texture blending

The texture image constructed from the target camera views, as shown in Figure 6.5(a), provides the highest resolution texture that can be achieved with a single frame of camera images. This texture image can however contain distortions at the boundary between the triangles textured from different images due to any misregistration of the model surface with the images or where there is a different view-dependent appearance of the surface in each view. Texture blending is performed at these boundaries to ensure a smooth transition in the colour texture recovered from the different view points.

Texture blending is performed using a weighted average of the textures derived from different views. A smooth transition at the texture boundaries can be achieved with a smooth transformation of the relative weight given to each view. Techniques for blending make use of factors such as the relative orientation of the surface with respect to a camera view and the relative distance to the edge of the texture to define the weighting term [125, 129]. Here a multiresolution blending technique is used, as presented by Burt and Adelson [27] and used for texture blending by Lee and Magnenat-Thalmann [129] in human head modelling. The multi-resolution approach ensures that the extent of texture blending corresponds to the spatial frequency of the features in the texture image, preserving the higher frequency detail that can become blurred with other techniques as demonstrated in Section 6.3.

The multi-resolution spline described by Burt and Adelson [27] performs a weighted average for a set of images at a sequence of image resolutions. A Gaussian image pyramid is constructed for each image giving a low-pass filtered version of the original image at successively lower resolutions as shown previously in Figure 6.4. A band-pass filtered image sequence, the Laplacian image pyramid, is then constructed by taking the difference between each level in the Gaussian image pyramid. The Laplacian pyramid provides the image features at different spatial frequencies and has the property that the summation of the pyramid reconstructs the original image [27]. A weighted average is performed at the different levels of the Laplacian pyramid across all the original images. The averaged pyramid can then be summed starting at the penultimate resolution of the pyramid by bi-linear interpolation of the pixel values from the preceding lower resolution image to give the final blended image.

The relative weight given to each pixel in each Laplacian pyramid is defined using an additional pyramid structure [27]. A Gaussian pyramid is constructed for each target image mask, the mask defining the region of each texture image for which the corresponding camera image is the target view. The initial mask is given a binary value $(0, 1)$. The mask is then low-pass filtered in the Gaussian pyramid to give continuous weight defining the relative influence of each pixel in the pyramid. The pixels in each Laplacian pyramid are weighted by the corresponding value in the Gaussian pyramid for the target mask and summed to give the multi-resolution weighted average. The

effect of the Gaussian-filtered masks is to smoothly spread-out the relative weight given to corresponding pixels in each of the Laplacian pyramids. At the highest levels of the pyramid the low-pass filtered masks give only a localised weighted average, preserving the high-frequency image features. At the lowest levels of the pyramid the masks are smoothed to give a weighted average of the low-frequency features across the entire image.

Input: images, $\underline{I}_m$
image masks, $\underline{M}_m$

Output: blended image, $\underline{I}$

Procedure: *multiresolution_blend*

1. for (*each image* $\underline{I}_m$ *and mask* $\underline{M}_m$)
2. *push_pull_interpolate*($\underline{I}_m$)
3. *construct_Laplacian_image_pyramid*($\underline{I}_m$)
4. *construct_Gaussian_image_pyramid*($\underline{M}_m$)
5. (*Multiply Laplacian with Gaussian pyramid*)
6. (*Sum Laplacian pyramids*)
7. for (*Laplacian image* $\underline{I}_l$, $l = N_l - 2 \ldots 0$)
8. for (*each pixel* p *in* $\underline{I}_l$)
9. set ($p = p+$ *bi-linearly_interpolate*($\underline{I}_{l+1}, p$))
10. set ($\underline{I} = \underline{I}_0$)

Input: camera images, $\underline{I}_m$
image correspondence, $\underline{u}_{im}$
texture map specification, $\underline{u}_i^{TEX}$

Output: blended texture image, $\underline{I}^{TEX}$

Procedure: *view-independent_texture*

1. set (*valid view assignments,* v_{fm})
2. set (*target view assignments,* $\widetilde{v}_f$)
3. *group_target_assignment*($\widetilde{v}_f$)
4. for (*each camera image* m)
5. *resample_texture*($\underline{I}_m^{TEX}$)
6. *resample_mask*($\underline{M}_m^{TEX}$)
7. (*Synthesise missing texture*)
8. *multiresolution_blend*(*all images and masks*)

It is important to note that the multi-resolution technique presented by Burt and Adel-

son [27] requires images that completely overlap so that every pixel in each Laplacian pyramid is defined. The image masks must also be non-overlapping and cover the entire image range to ensure that the weight at each pixel across all images sums to unity. The resampled texture image for each camera view is therefore expanded to cover the entire image range using the "push-pull" interpolation technique described in Section 6.1.4. The images are also blended to synthesise missing texture with the mask for the missing texture constructed from the complement of the target image masks. The final view-independent texture map for the model is obtained with the algorithm *view-independent_texture*. Results for this approach are presented in Section 6.3.

6.2 View-Dependent Texturing

In view-dependent texturing the original camera images are used as a set of texture maps for a model and blended dynamically according to the view-point used in rendering. Each camera image samples the view-dependent appearance of the surface light-field for the model. This view-dependent appearance is then reproduced by blending the different camera textures according to the proximity of each camera to the rendered view-point. View-dependent texturing is presented in this section. A vertex-centred weight is first described in Section 6.2.1 to obtain a view-dependent colour across a surface mesh. Section 6.2.2 then describes a triangle centred weight to texture the surface triangles from a subset of closest cameras. Finally in Section 6.2.3 a multi-pass rendering algorithm is presented making use of view-dependent colour and texture to synthesise a virtual view.

6.2.1 View-dependent colour

The input to view-dependent texturing is the visibility and image plane correspondence for each model vertex in every camera view. This image correspondence is used to derive a colour at each vertex to give a view-dependent colour across the surface of the model. A blend weight b_{im} is calculated at each vertex i for each image m to favour the camera views closest to the desired viewing direction. The proximity of a

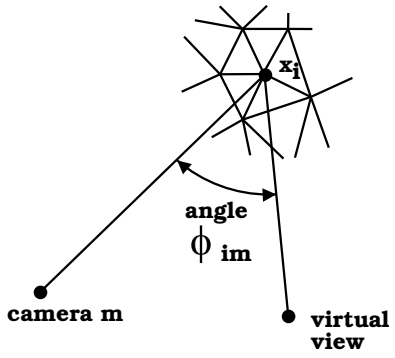


Figure 6.6: The virtual viewing angle ϕ_{im} used to define the view-dependent weight b_{im} of vertex i with respect to camera m for a virtual viewpoint.

camera to the virtual view is defined as the cosine of the angle from the camera viewing direction to the viewing direction of the virtual view $b_{im} = \cos\phi_{im}$ [134, 129] as shown in Figure 6.6. The vertex weights are normalised to sum to one across all visible views $\hat{b}_{im} = v_{im}b_{im}/\sum_m v_{im}b_{im}$. The colour at each model vertex $\underline{I}_i$ is finally defined as the weighted average of the image colour in each image, $\underline{I}_i = \sum_m \hat{b}_{im}\underline{I}_{im}$. Some vertices may be occluded in all camera views, in which case a vertex colour cannot be derived. Each vertex with no colour assignment is therefore iteratively assigned an average of the adjacent vertex colours to give a complete description of the surface appearance.

6.2.2 View-dependent texture

Techniques for view-dependent texturing make use of the subset of the available camera images closest to the rendered viewpoint [47, 112, 134]. In the general case where cameras are located at arbitrary positions in space, camera selection has been based on the three closest cameras surrounding the virtual viewpoint. In our studio the cameras are located in a circle in order to surround a person from a limited set of views. The two closest cameras to the desired virtual view are therefore selected for view-dependent texturing [129]. A view-dependent weight is derived at the triangle vertices of the mesh to define the relative influence of these two closest views in texturing each triangle.

The view-dependent vertex weight b_{imf} for each vertex i on each triangle facet f is again defined by the proximity of the camera viewing direction to the virtual view given by

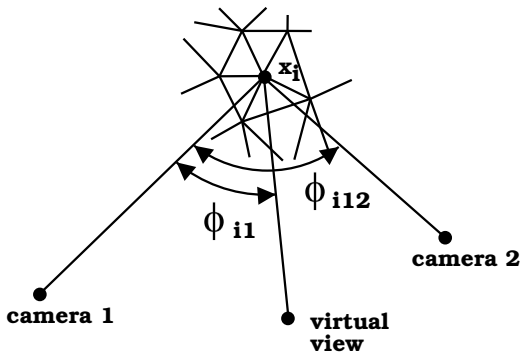


Figure 6.7: The virtual viewing angle ϕ_{im} used to define the trade-off between two cameras using the difference $(\cos\phi_{i1} - \cos\phi_{i12})$.

the angle ϕ_{im}. The blend weight is now defined as $b_{imf} = \cos\phi_{if} - \cos\phi_i$, where ϕ_i is the angle between the two viewing directions to the cameras used for texturing at the vertex [129] as shown in Figure 6.7. Blending now favours the original camera views exactly when the virtual viewing direction is coincident with a camera viewing direction. The view-dependent vertex weight is set to zero, $b_{imf} = 0$, if any of the vertices are not visible in the camera view. The vertex weights are finally normalised to sum to one across the two texture views to give $\hat{b}_{imf}$.

The vertex weights $\hat{b}_{im}$ provide a view-dependent vertex colour $\underline{I}_i$ for the model that varies smoothly according to the virtual viewpoint in rendering. The triangle weights $\hat{b}_{imf}$ provide a smooth transition across each triangle between two camera images for view-dependent texturing.

6.2.3 Multi-pass rendering

The model is rendered in multiple passes to a virtual view with view-dependent weighting of the texture from the two adjacent camera images. Some triangles on the model will not necessarily be visible in either image and will therefore have no texture in rendering. In an initial render pass the model is first coloured with the view-dependent colour derived at the mesh vertices in order to fill the missing areas of texture. The surface colour is then replaced where possible by the detailed texture from the camera images in subsequent passes.

Input: camera images, $\underline{I}_m$
image correspondence, $\underline{u}_{im}$
virtual camera parameters, $\mathbf{P}$

Output: rendered image

Procedure: *view-dependent_texture*

1. set (*vertex weights* $\hat{b}_{im}$)
2. set (*triangle vertex weights* $\hat{b}_{imf}$)
3. set (*weighted vertex colour* $\underline{I}_i$)
4. set (*unassigned vertex colours*)
5. set (*interpolation operation to "smooth"*)
6. set (*depth test operation to "less"*)
7. set (*perspective view for rendering with* $\mathbf{P}$)
8. for (*each triangle facet* f)
9. set (*vertex colours from* $\underline{I}_i$)
10. render (*triangle*)
11. set (*depth test operation to "less than or equal"*)
12. for (*each closest camera image* m)
13. load (*camera image* $\underline{I}_m$ *as texture*)
14. for (*each triangle facet* f)
15. set (*texture coordinates from* $\underline{u}_{im}$)
16. set (*texture modulated by vertex weights* $\hat{b}_{imf}$)
17. set (*vertex locations from* $\underline{x}_i$)
18. if (*triangle not textured previously*)
19. set (*texture operation to "replace"*)
20. else set (*texture operation to "add"*)
21. render (*triangle*)

The algorithm for view-dependent rendering is outlined in *view_dependent_texture*. In the first render pass (*steps 8-10*) the vertex colours are smoothly interpolated across each surface triangle. Depth testing is enabled (*step 6*) in rendering to eliminate the hidden surfaces in the rendered view. Depth testing is then switched to the operation *less than or equal* (*step 11*) allowing the same surface to be rendered again. In a second render pass the triangles are then textured from the closest camera view and in a final pass, textured from the second closest view. When a triangle is first textured the texturing operation is set to *replace* in order to replace the coloured surface (*step 19*), and in the second instance that a triangle is textured the operation is set to *add* in order to add the weighted textures from the two camera views (*step 20*). The triangle

texture is modulated by the view-dependent weights at the triangle vertices $\hat{b}_{imf}$ to give a weighted blend between the two camera textures (*step 16*).

6.3 Evaluation

Two approaches have been proposed to define the surface appearance for a reconstructed model. In Section 6.1 a view-independent texture map is derived and in Section 6.2 an image-based representation is used to provide a view-dependent appearance. These techniques are now evaluated subjectively and compared with the appearance in the original camera images for three real data-sets. The reconstructed shape models for these cases were shown previously in Figures 5.16(b), 5.17(b), and 5.18(b).

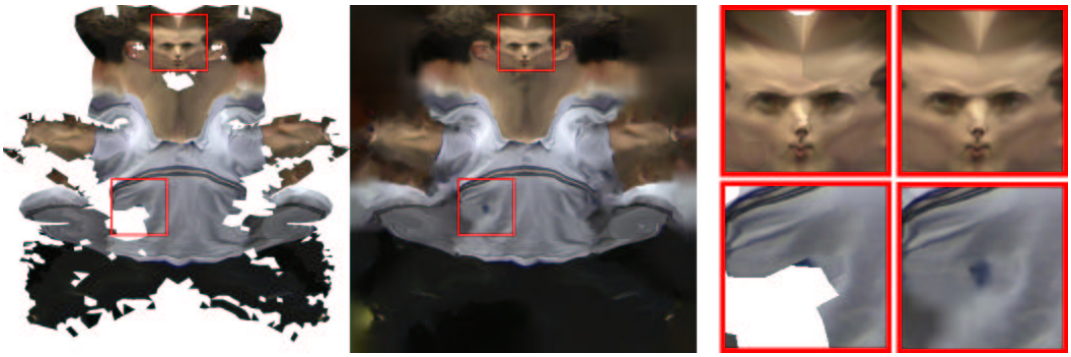
6.3.1 View-independent model texture

The view-independent texture maps derived for the three subjects captured in the studio are shown in Figure 6.8 together with a rendered view for each model. Subjectively these texture maps provide a visually realistic surface appearance for the reconstructed models. The shape of the models has been optimised to obtain sub-pixel accurate correspondence between views as described in Chapter 5. This minimises the blurring and misalignment of features that would otherwise occur in recovering texture with inaccurate registration in multiple views.

The texture image for a model is recovered through a process of target view assignment (Figure 6.2), camera image resampling (Figure 6.3), texture filling (Figure 6.5), and texture blending. The final blended image is now shown in Figure 6.9(b) for the case presented in Figures 6.2, 6.3, and 6.5. The texture resampled from the assigned camera images shown in Figure 6.9(a) provides the highest resolution appearance that is possible from the single frame of camera images. Texture filling provides the missing sections of colour where the surface is not visible in any camera images. Texture blending then provides a smooth transition across the boundaries between the texture resampled from different cameras. The process provides a single complete texture image for a model.

(a) Texture-maps (b) Textured Model (c) Textured Model (d) Textured Model

Figure 6.8: Texture-mapped models for three subjects captured from 9 camera views. The reconstructed model shape is shown in Figures 5.16, 5.17, and 5.18.



(a) Resampled texture (b) Filled and blended texture (c) Highlighted texture regions

Figure 6.9: Resampling, filling and blending texture showing (a) the camera images resampled to the texture map, (b) the filled and blended texture, and (c) two highlighted texture regions demonstrating blending and filling.

Filling and blending in a texture image is highlighted in Figure 6.9(c). Blending removes the boundary between different camera views that can be seen in Figure 6.9(a). A multiple resolution technique is used to blend the texture without any loss of detail in the appearance as demonstrated in 6.9(c). In regions were there is no resampled surface colour the texture map is filled using a "push-pull" interpolation technique. This provides the missing colour from adjacent regions on the model surface. In Figure 6.9(a) the texture is missing at the sides of the torso, the inner surface of the legs and under the chin due to self-occlusion in the camera images. Image interpolation fills these areas to provide a complete description of the surface appearance. It should be noted that detailed appearance such as a clothing pattern cannot be synthesised using this technique for the occluded surface regions. The technique will also reproduce any artifacts in the surrounding colour as highlighted in Figure 6.9(c) where some of the background image colour has been resampled to the texture map and interpolated in the missing texture region.

The texture maps derived from multiple camera views provide a "photo-realistic" appearance for a reconstructed model of a person. These models can then be animated and rendered in a standard graphics pipeline. In Chapter 8.1 the technique is applied to different subjects in a multiple camera studio and model animation is demonstrated.

6.3.2 View-dependent rendering

The rendered appearance of the three reconstructed models is now compared to the original camera images. Figures 6.10, 6.11, and 6.12 show five different virtual views rendered for each model first using the view-independent model texture and then using view-dependent rendering from the original camera images. Subjectively view-dependent rendering appears to provide a slight improvement in the rendered images. View-dependent rendering removes the intermediate step of resampling the camera images to a texture map and so can maintain the resolution of the original camera images more faithfully. View-dependent rendering also reproduces the captured view-dependent lighting in the camera images. This will however make the technique sensitive to variations in the lighting conditions between cameras.

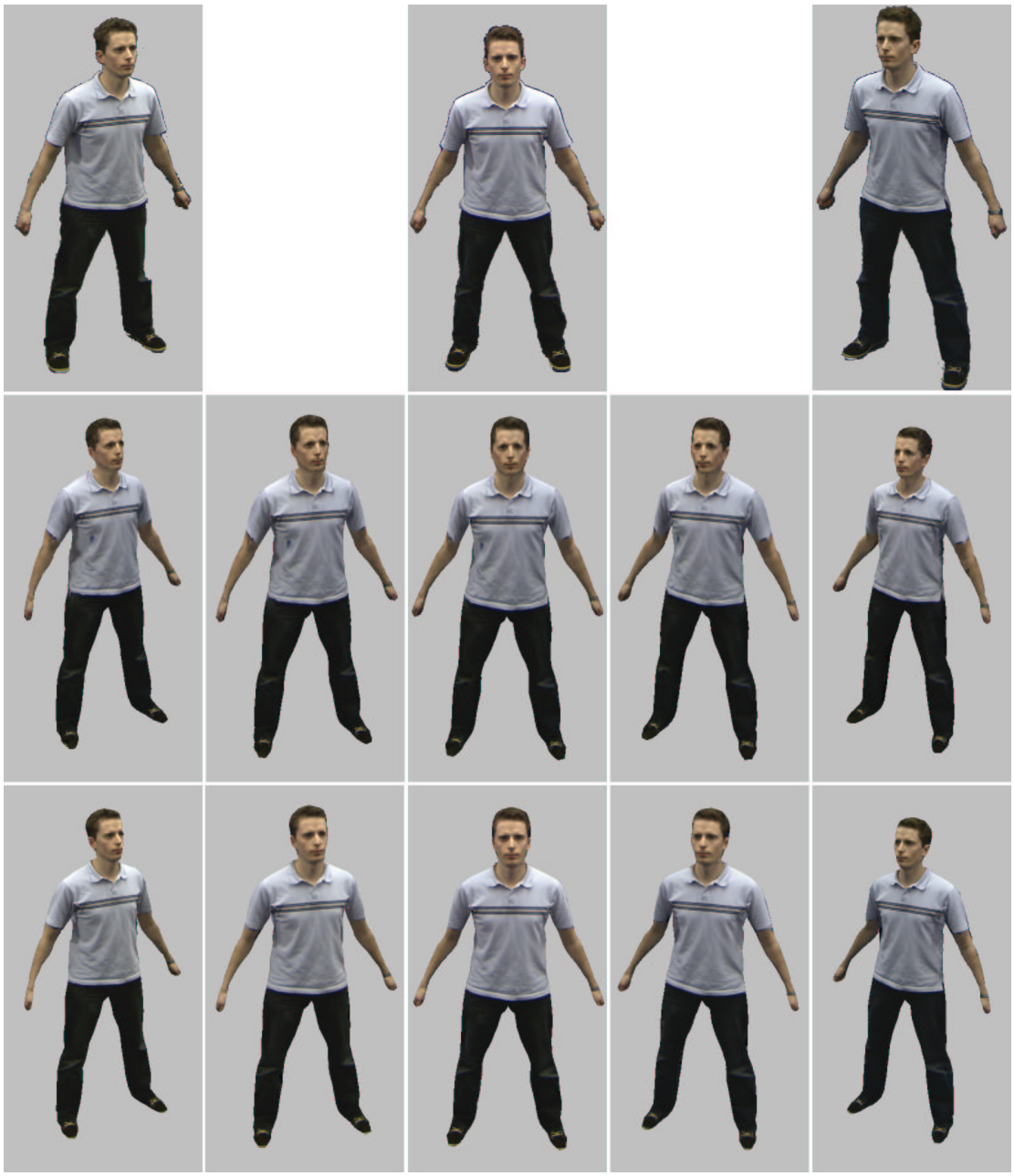


Figure 6.10: Five different virtual views of a reconstruct reconstructed model, comparing the original camera images (top row), the view-independent model texture (middle row) and view-dependent rendering (bottom row).

Figure 6.11: Five different virtual views of a reconstruct reconstructed model, comparing the original camera images (top row), the view-independent model texture (middle row) and view-dependent rendering (bottom row).

Figure 6.12: Five different virtual views of a reconstruct reconstructed model, comparing the original camera images (top row), the view-independent model texture (middle row) and view-dependent rendering (bottom row).

Both view-independent texture and view-dependent rendering provide a visual quality approaching the captured image resolution. In fact the most noticeable flaw in in Figures 6.10, 6.11, and 6.12 is not the surface appearance for either technique, rather it is the incorrect model shape. The model-based technique relies on the shape in the generic humanoid model to constrain the reconstruction process. The models therefore cannot reproduce the complex shape of the hair or hands in the camera images that is not present in the generic model. A more complex model could be considered for reconstruction to provide a higher degree of freedom in the surface shape. However this in turn could increase the reconstruction errors arising from noisy data. View-dependent rendering can potentially overcome this problem by providing a view-dependent visual cue to the missing geometry with an approximate model.

View-dependent rendering can only be applied for the fixed pose in which a person is captured in the multiple camera views. For an animated model pose the relationship between the vertex location and an original camera view is no longer valid and the view-dependent blend weight b cannot be calculated. While view-dependent rendering subjectively improves the visual-realism in rendering a reconstructed model, it cannot be applied to an animated model for the creation of new 3D content. In Chapter 8.1, view-dependent rendering is applied to render virtual views for the fixed geometry captured in multiple view video sequences.

Conclusion:

1. **The visual appearance derived with a view-independent texture map and with view-dependent rendering approach the resolution of the original camera images;**

2. **A view-independent texture map allows a reconstructed model to be animated and rendered in a standard computer graphics pipeline; and**

3. **View-dependent rendering can subjectively improve the fidelity in rendered images.**

6.4 Summary

In this chapter the surface colour for a model is recovered from a set of camera views given the image plane correspondence for the model vertices in the images. The shape-constrained deformable model outlined in Chapter 5 is used to reconstruct the shape of a person and obtain good correspondence between views to simplify the problem of deriving model appearance from multiple views. Two approaches are evaluated to specify model appearance through either a view-independent or a view-dependent representation. A view-independent approach using texture mapping gives a realistic appearance with a visual quality approaching the captured image resolution. The reconstruction of model geometry and a texture map allows the model to be animated and rendered in a conventional graphics pipeline. This approach does not however reproduce the changes in appearance of a person with viewing direction, resulting in a potential loss of visual realism. View-dependent texturing with an image-based representation reproduces the captured changes in the appearance of a person with viewpoint. Subjectively this produces a slight increase in the visual fidelity of the rendered images. However, view-dependent texturing can only be applied to render the static shape of the model captured in the images without model animation.

Chapter 7

Application

A model-based framework has been introduced to reconstruct the shape and appearance of a person from multiple camera views. The technique has been evaluated in Chapters 3, 4, 5 and 6 using synthetic data for a quantitative analysis of performance and real test cases for a subjective assessment of the results. The overall objective of this research work was to recover visually realistic models of people that can be controlled to synthesise new content for 3D production. In this chapter the application of the technique is demonstrated. In Section 7.1 the reconstruction of a person in different poses is presented. This demonstrates that the shape and appearance of a person can be recovered for an arbitrary pose with camera images taken from arbitrary positions in a studio. In Section 7.2 the generation of new 3D content is demonstrated for different models reconstructed in a studio. The model-based approach to reconstruction provides the control structure that allows the recovered models to be animated. This enables new dynamic scenes to be generated from the camera images and provides the freedom to render the models in different environments with visualisation from arbitrary 3D viewpoints. Finally in Section 7.3 the reconstruction frame-work is applied to multiple view video sequences of people. Model-based reconstruction requires a manual process to register a model with each frame of a video sequence. This can become an unfeasible task even for relatively short sequences. The framework is therefore applied as an object-centred approach to reconstruction without a prior model and compared to previous work for rendering virtual views from multiple view video.

Figure 7.1: Thirteen camera views captured in the multiple camera studio.

7.1 Capturing Dynamic Pose

A prerequisite for the model-based reconstruction technique was the requirement to reconstruct the shape and appearance of a person in an arbitrary body pose. Previous work on whole-body modelling is restricted to the reconstruction of a person in a fixed pose for orthogonal camera images [79]. In the multiple camera studio we wish to be able to recover the shape and appearance of a moving person from multiple arbitrary camera positions. Model reconstruction is now demonstrated in Figure 7.2 for a subject in a range of different poses captured in 13 camera views as shown in Figure 7.1. The model-based technique enables the reconstruction of the shape and appearance for a person from multiple views for different body poses in the presence of complex self-occlusions in the camera images.

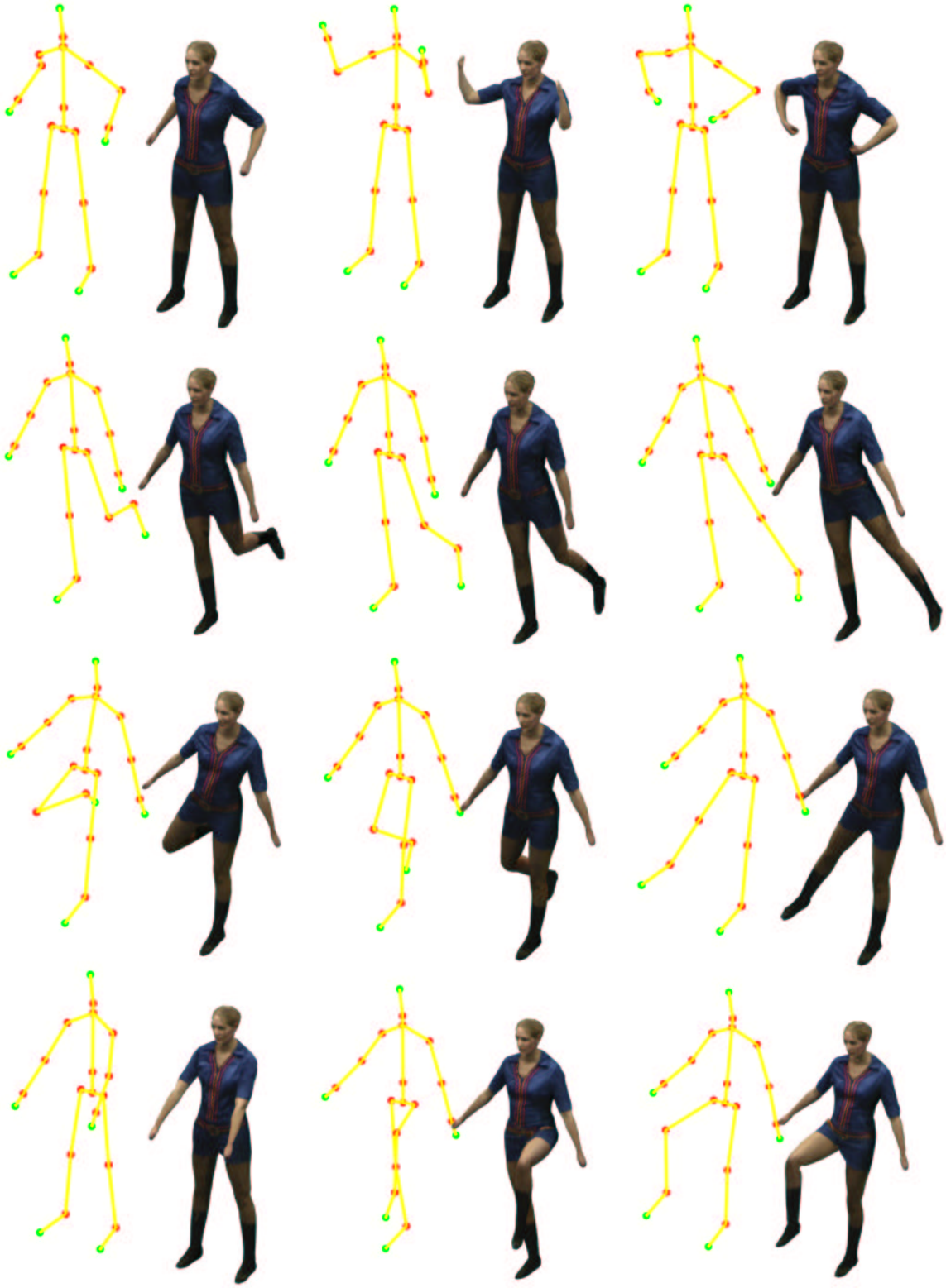


Figure 7.2: Models reconstructed for a range of different poses.

7.2 3D Content Production

The reconstruction and animation of the models for different subjects is now demonstrated. Six different subjects were reconstructed from from 13 camera views as illustrated in Figure 7.1. A view-independent texture map is derived for each model and the reconstructed shape and appearance of each subject is shown in Figure 7.3. Reconstruction with a prior model provides the control structure necessary to animate the models. This enables the creation of new dynamic content from the original camera images with the freedom to control the viewpoint in 3D visualisation as shown in Figures 7.4, 7.5, 7.6, 7.7, 7.8, and 7.9 for the six different models. Subjectively, the animated models provide a visual appearance that matches the captured images of each person. The drawback of the technique lies in the static nature of the model geometry and appearance. Human motion produces a complex range of movement in clothing and hair for example that is not reproduced in animating the fixed geometry of the model. These movements also generate a complex view-dependent appearance for a person arising from the spatially varying surface reflectance properties. The texture map derived for the reconstructed models can only provide a fixed view-independent surface colour.

7.3 Multiple View Video Sequences

The model-based approach to reconstruction is based on a prior model of human geometry. In Chapter 5 it was demonstrated that improved shape reconstruction can be obtained using this geometry to constrain the reconstruction process in comparison with current non model-based techniques. The disadvantage of this approach is that it limits the feasible space of the reconstructed shape. The generic model used in this work cannot for example model dresses or long hair. The model must also be registered to match the pose of a person in the multiple views, a process that currently requires the manual labelling of features in different images.

Non model-based approaches have been used previously to reconstruct dynamic scenes from multiple views [87, 113, 175]. These techniques provide an automatic method to

Figure 7.3: Six different models generated from 13 camera views in a studio.

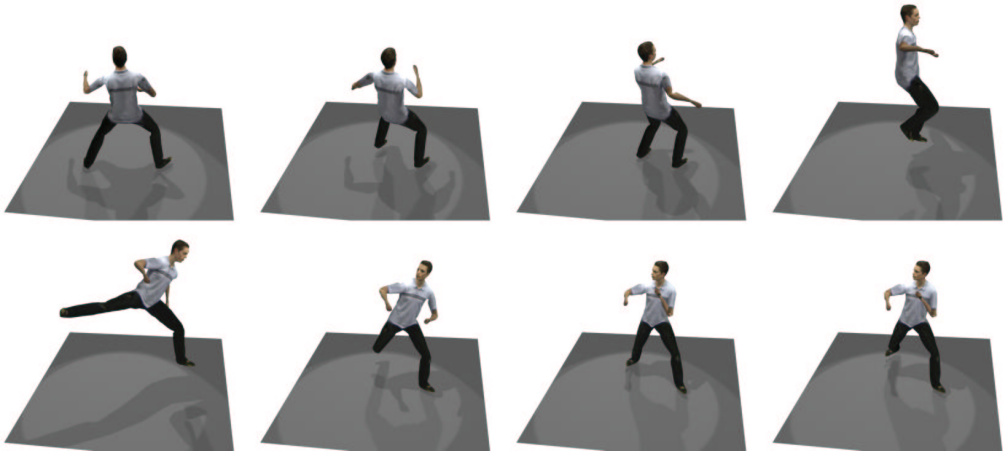


Figure 7.4: An animated sequence for a reconstructed model.

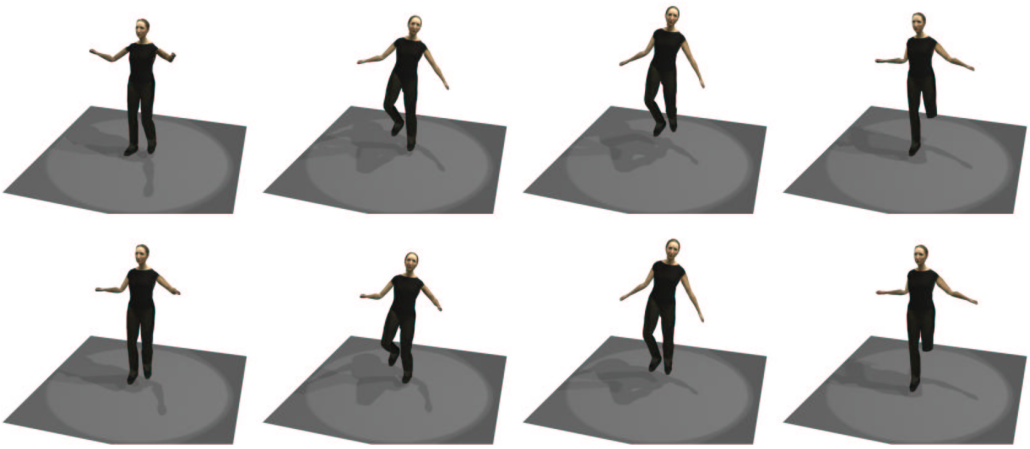


Figure 7.5: An animated sequence for a reconstructed model.

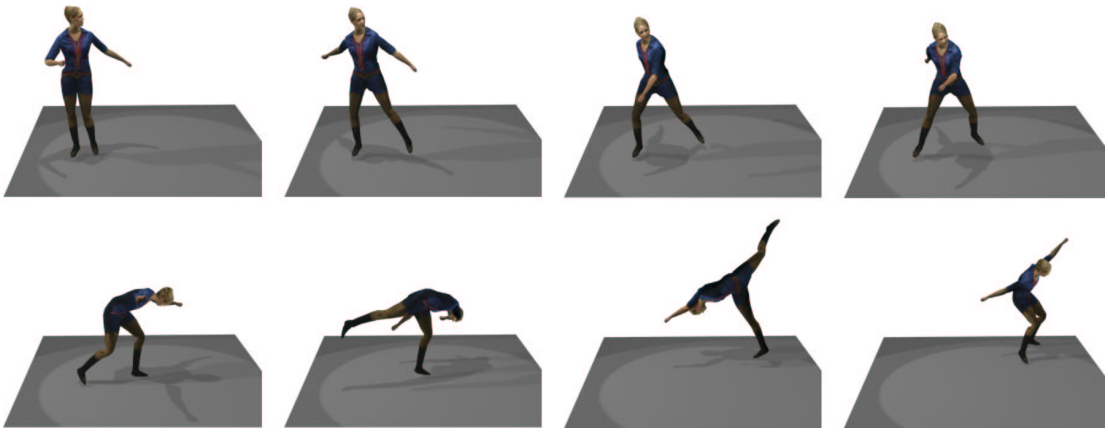


Figure 7.6: An animated sequence for a reconstructed model.

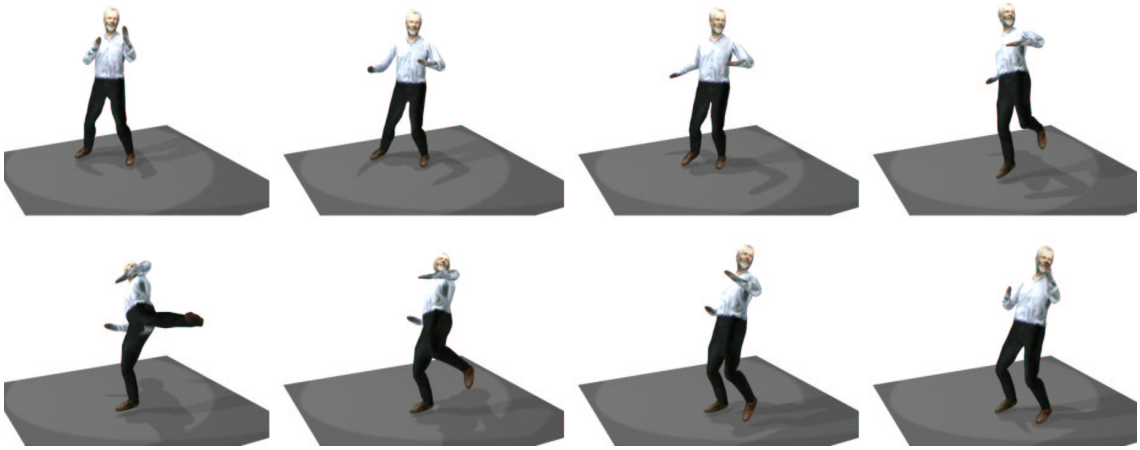


Figure 7.7: An animated sequence for a reconstructed model.

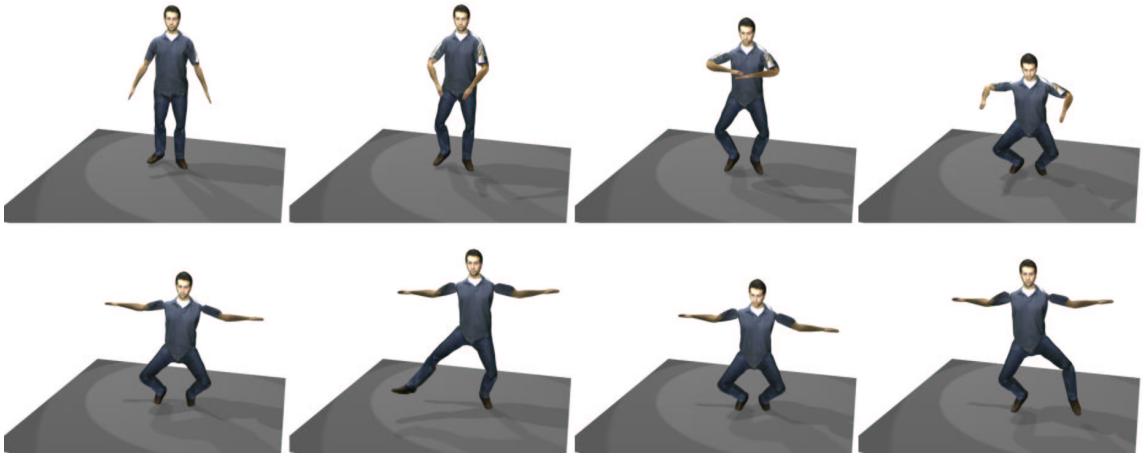


Figure 7.8: An animated sequence for a reconstructed model.

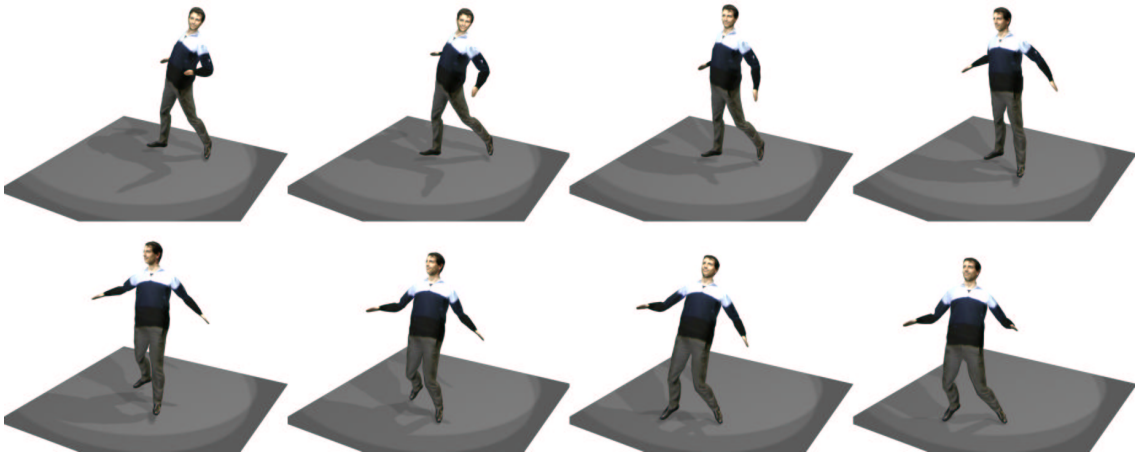


Figure 7.9: An animated sequence for a reconstructed model.

reconstruct geometry without any restriction on the dynamic content. Highly realistic results have been demonstrated using view-dependent rendering to provide visual cues to the detailed shape that may be missing in the underlying scene reconstruction [175]. However, non-model based techniques suffer from visual ambiguities in reconstruction.

In this section the framework for multiple view appearance reconstruction is applied as an object-centred approach to reconstruction. An initial estimate of the scene geometry is derived using the visual-hull. The shape of the visual-hull is then optimised to recover the correspondence between camera images for view-dependent rendering. A triangulated surface model is derived for the visual-hull using the *Marching Cubes* algorithm [106]. The model-based algorithm outlined in Chapter 5 is then applied to deform the shape of the surface mesh to match both stereo and silhouette data. A smooth regularisation constraint is obtained in the shape constrained model by first smoothing the discrete initial surface of the visual-hull. The model-based framework refines the model shape and provides sub-pixel accurate image correspondence for the view-dependent rendering algorithm outlined in Chapter 6.

The application of the technique for the synthesis of novel views from multiple view video sequences is presented in Figures 7.10 and 7.12. This is compared to view dependent rendering with the visual-hull alone [113] in Figures 7.11 and 7.13. These results demonstrate that the resolution can be improved in rendering virtual views by recovering the correspondence between the images for rendering. Rendering with the visual-hull alone provides a blurred appearance due to incorrect correspondence between the camera images used in view-dependent texturing.

7.4 Summary

The model-based framework for the reconstruction of the shape and appearance of a person has been demonstrated for different subjects in a studio and for a variety of different poses. The view-independent texture derived for the models subjectively provides a highly realistic appearance that approaches the camera resolution used in model reconstruction. The model-based approach to reconstruction provides a control structure to animate the models and synthesise new dynamic 3D content. The principal

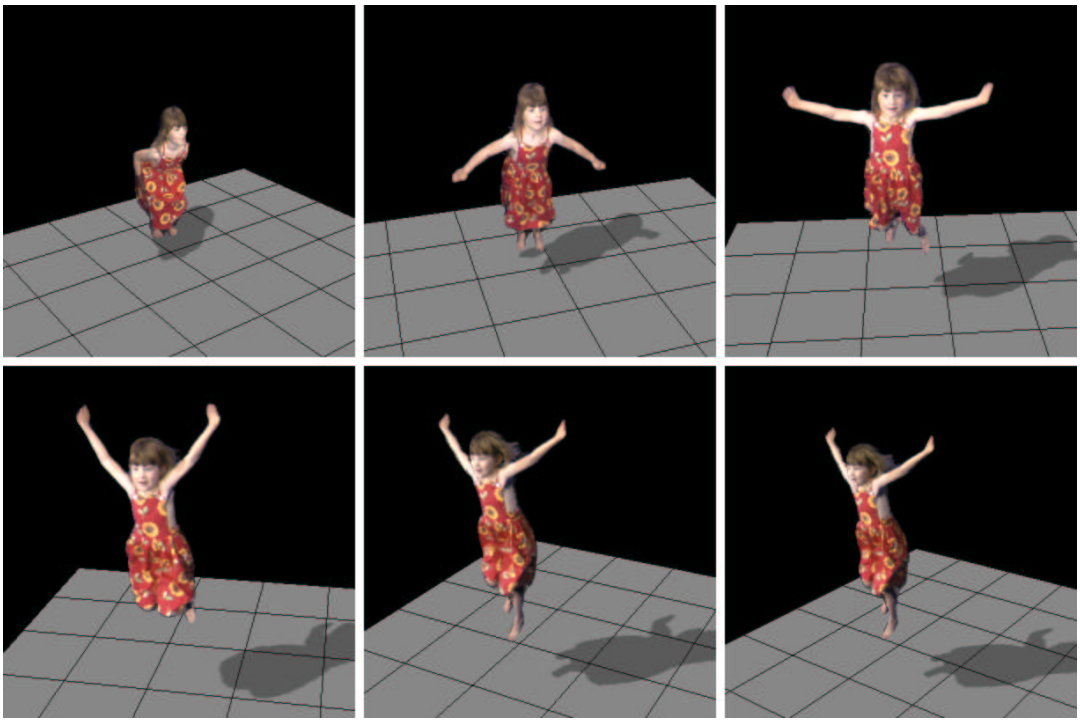


Figure 7.10: Virtual views with view-dependent rendering of the visual-hull optimised using the model-based framework in this thesis.

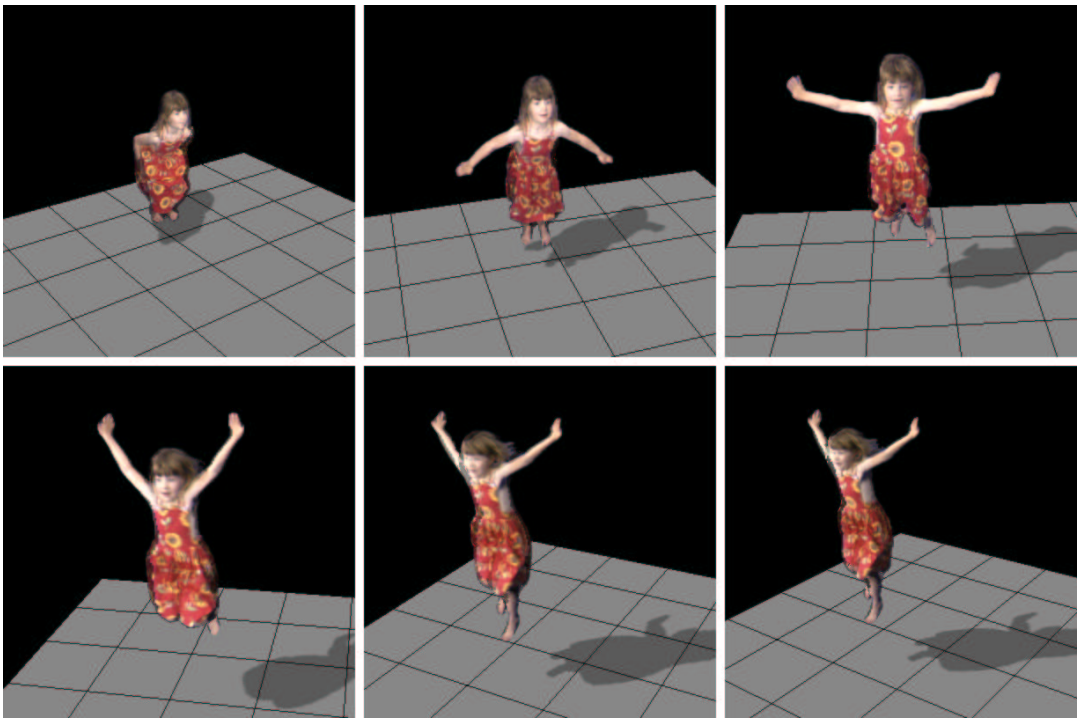


Figure 7.11: Virtual views with view dependent rendering of the visual-hull.

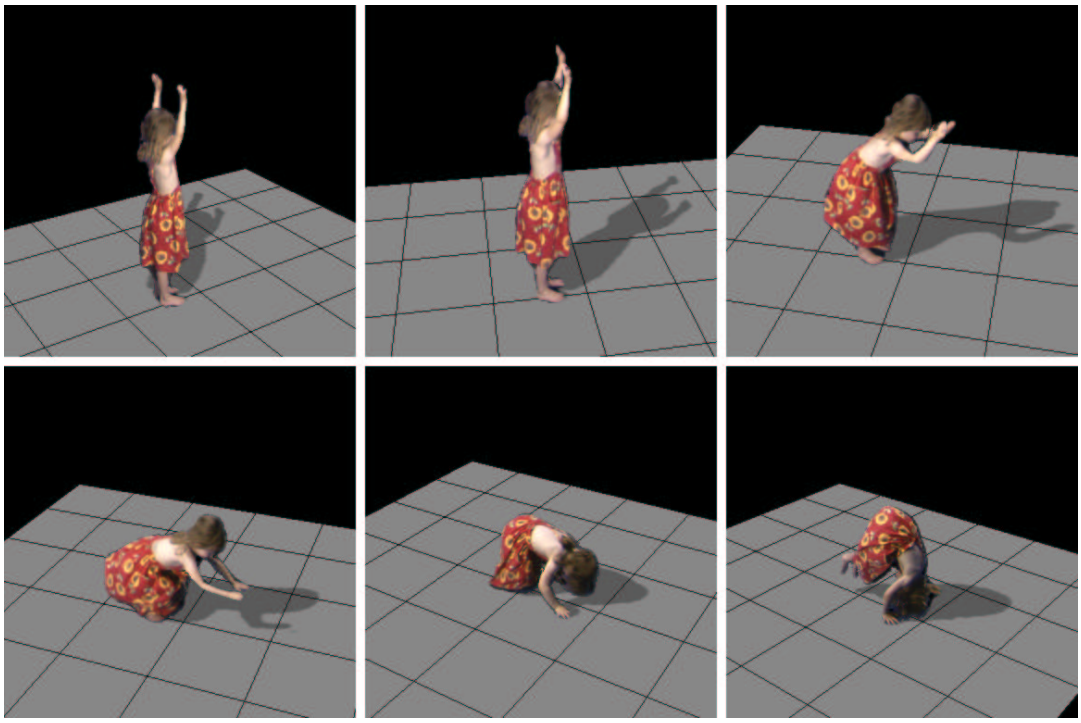


Figure 7.12: Virtual views with view-dependent rendering of the visual-hull optimised using the model-based framework in this thesis.

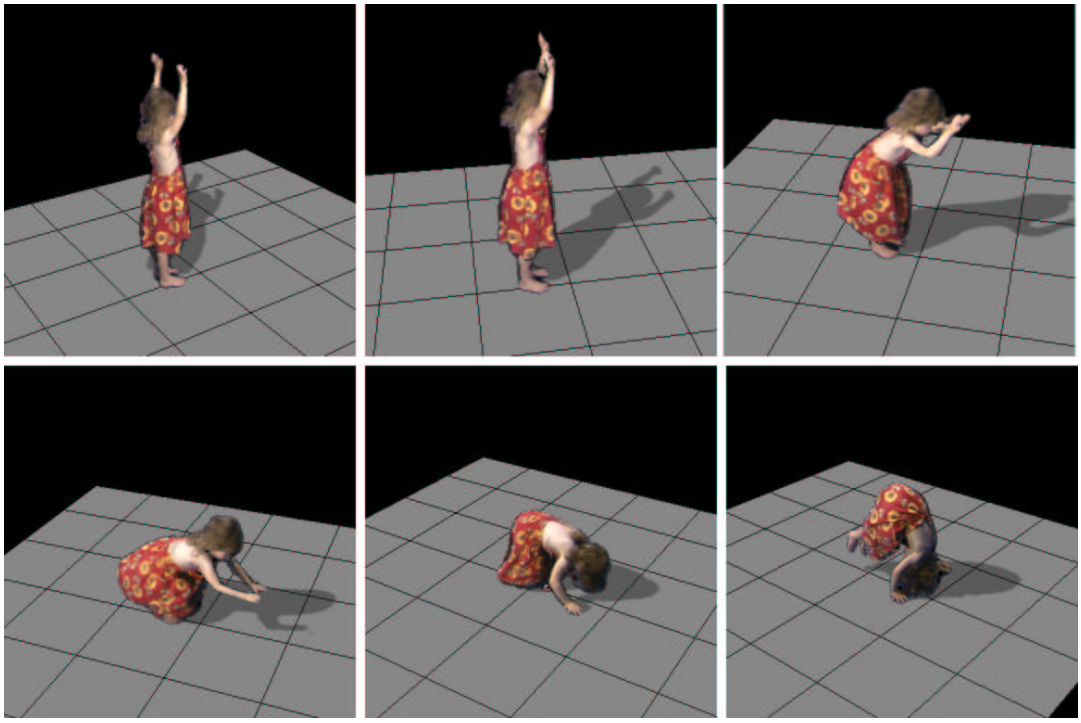


Figure 7.13: Virtual views with view dependent rendering of the visual-hull.

limitations in the visual realism of the models lie in the restricted shape of the model and the fixed appearance. The technique cannot reconstruct complex shapes in clothing or hair. The animated models do not reproduce the complex changes in geometry with clothing movement and view-dependent appearance with spatially varying surface reflectance properties. The advantage of a model-free approach to reconstruction is that it makes no prior assumptions on the content of a captured scene and the complex geometry of clothes and hair can be represented. This can be combined with view-dependent rendering to synthesise highly realistic novel views with dynamic changes in shape and appearance. The disadvantage of a non-model based approach is that it does not provide a consistent model structure necessary to edit or synthesise new 3D content from a captured video sequence. View-dependent rendering also cannot be applied with animation of the scene geometry.

Chapter 8

Conclusions and Further Work

The creation of realistic human models in 3D content production is currently a high cost process requiring the skills of experienced computer graphics artists and animators. Humans play a central role in most forms of visual media and these models have found widespread use throughout many different industries such as computer games, film, and advertising. The research in this thesis has focussed on the problem of deriving an animated human model from multiple camera views of a real person. This provides the potential for the rapid creation of highly realistic models that would allow the synthesis of new 3D content from the original camera images and allow freedom of viewpoint in 3D visualisation.

8.1 Achievements

The reconstruction of 3D shape from camera images forms a central problem in Computer Vision and techniques have been developed previously to construct models from multiple views [87, 113, 175]. In recent years the problem of generating visually realistic models from images has been addressed. Image-based representations for appearance have been combined with geometric scene reconstruction to give highly realistic 3D models with the visual quality of camera images. These non model-based approaches to visual reconstruction can fail to accurately reconstruct shape and appearance in the

presence of visual ambiguities. The techniques also provide no structure to edit or reuse the captured content in computer animation.

A model-based computer vision framework has been introduced in this thesis to derive the shape and appearance of a person from multiple camera views in the presence of visual ambiguities. The philosophy of the approach is to use a generic humanoid model for reconstruction following the functional modelling paradigm introduced by Terzopoulos et al. [162]. The generic model is instrumented with a skeletal control structure that allows the recovered models to be animated for the synthesis of 3D content. A number of publications have resulted from this work [132, 80, 152, 149, 153, 151, 150]

Model-based reconstruction of whole-body models has been presented previously by Hilton et al. [79]. In Chapter 3 this technique was extended to recover the shape of a person in an arbitrary pose viewed from an arbitrary camera position for application in the multiple camera studio [152]. This work is based on matching the projected shape of a generic model to the silhouette of a person in a camera view. Evaluation of the technique lead to the important conclusion that establishing the model correspondence in 2D is ill-posed with inexact model shape and pose as well as self-occlusions in the images.

In Chapter 4 a technique was presented for model-based shape from multiple view silhouettes [153, 80]. The technique integrates the shape from image silhouettes by optimising a model to match the shape of the visual-hull. This avoids inconsistent matches found in establishing the correspondence of a model with silhouettes in 2D. A multiple point matching scheme is proposed to obtain robust matches in the presence of visual ambiguities and avoid local minima in the matching the model to the surface of the visual-hull [153, 149]. A shape constraint has been introduced to regularise the deformation of the model in optimisation [153, 149]. The constraint is formulated for a triangulated surface model to preserve the relative position of the vertices defining the surface shape. This ensures that the prior correspondence between the vertices and the animation structure of the model remains valid. The model-based reconstruction algorithm enables an approximate shape model to be recovered from multiple view

image silhouettes.

Image silhouettes provide only a bounding approximation on the shape of a person. Evaluation of the model-based technique for shape from silhouette demonstrated that with an approximate shape the correspondence between camera images is not correct for the reconstruction of appearance. In Chapter 5 a technique was introduced to optimise a generic model to recover both the shape and the correspondence that matches appearance in the images. The technique introduces a model-based approach to integrate multiple visual cues for shape recovery [151, 150]. Multiple-view stereo is used to refine a model shape to match appearance between images. Shape from silhouette is used to provide a robust shape constraint in regions of uniform appearance where stereo matching fails. Finally, image features are incorporated to provide a sparse set of constraints on shape where stereo matching fails in PAL resolution whole-body images. Evaluation of the technique demonstrated improved shape reconstruction in comparison with current techniques for multiple reconstruction such as *Voxel Coloring* [140] and multiple view stereo [123] in the presence of visual ambiguities.

The model-based framework for shape from stereo, silhouette and feature reconstructs the shape of person from multiple views and provides sub-pixel accurate image correspondence for the reconstruction of model appearance. In Chapter 6 techniques have been described to recover either a view-independent texture map for a model or to render a model with a view-dependent appearance. With a view-independent texture map the reconstructed models can be animated and displayed in a standard computer graphics pipeline. With view-dependent rendering subtle changes in the appearance of a person with viewing direction can be reproduced. Evaluation demonstrated that a realistic appearance can be derived for a model through both techniques with a visual quality approaching the captured image resolution. The application of the reconstructed appearance models was shown in Chapter . The synthesis of new dynamic 3D content with the freedom to control the viewpoint in visualisation has been demonstrated.

8.2 Further Work

There are two principal limitations in the model-based framework for shape and appearance reconstruction. The first is the requirement for a set of manually defined feature points to register a generic humanoid model with the camera images and to define the sparse feature constraints defining the shape of the face. Manual labelling of a limited set of features in multiple view images is not necessarily a lengthy task. This has enabled the reconstruction of models from static frames for the synthesis of new content and can provide the means to analyse the dynamic shape and appearance of a person in different poses. However, labelling multiple view video sequences that can extend for thousands of frames is unfeasible. The technique is therefore restricted to a limited set of frames. Further work is required to automate the process of pose estimation from multiple views of a person and to derive facial feature correspondences in the images.

The second limitation in the framework is the restriction on the reconstructed shape imposed by the geometry of the generic model. The generic model used in this work failed to reproduce the complex shape of the hair and hands for the test subjects. The model will also fail to reconstruct clothing that does not follow the shape of the body such as dresses or coats. This represents a fundamental problem for model-based multiple view shape and appearance reconstruction. The prior model is required to constrain reconstruction in the presence of visual ambiguities and a greater degree of freedom in the model shape may simply lead to greater reconstruction errors rather than a more accurate shape representation. Further work is required to investigate the influence of the model chosen for reconstruction against a range of subjects wearing different clothing.

The model-based framework provides the means to reconstruct the shape and appearance of a static scene of a person with an animation structure to synthesise new content. The challenge remains to extend this work to a model-based representation of complex dynamic sequences of a person. There are many exciting avenues of future research for this work. The use of marker free visual motion capture from multiple views provides the potential for an automated system to capture both the dynamic pose of a person

and reconstruct the dynamic shape. The view-dependent appearance captured in the multiple view video provides the potential to then define the dynamic changes in model appearance. Calibration of the lighting conditions would also provide the potential to relight the models in new environments. Model based reconstruction enables the capture of an appearance model for a person that provides control over the model dynamics and viewpoint in visualisation. Future research will enable a model to reproduce the dynamic changes in shape and appearance of person to give a high degree of visual realism to a computer graphics model.

Appendix A

Camera image projection and reconstruction

Video sequences are recorded from multiple cameras in a dedicated studio. Sony DXC-9100P 3-CCD colour cameras are used, providing PAL-resolution progressive scan images at 25Hz. The cameras are synchronised by an external trigger and the RGB analogue output is converted to a digital SDI stream using Miranda ASD-111i analogue to digital converters. The SDI streams are time stamped using a Miranda TCP-101i time code generator and stored to disk using DVS SDStationBoard framegrabbers on a PC network. The system provides 8 channels of synchronised video with an additional 5 channels available for non-synchronous capture. The studio is equipped with a lighting grid to provide controlled lighting conditions and a blue curtain for background segmentation. All cameras are colour calibrated by white-balancing the RGB output with a white reference object. The studio set-up enables broadcast standard multiple view digital video capture.

A.1 Pin-hole camera model

The geometric projection of a three-dimensional (3D) point in space to the two-dimesional (2D) plane of a camera image is defined by the extrinsic and intrinsic parameters of a camera model. The extrinsic camera parameters $(\mathbf{R}, \underline{t})$ define the orientation of the

camera reference frame with respect to the world reference frame. The relationship between the coordinates of a point in world coordinates $\underline{x}$ and a point in the camera frame $\underline{x}_m$ for image m is defined as follows.

$$\underline{x}_m = \mathbf{R}_m \underline{x} + \underline{t}_m \tag{A.1}$$

In the pin-hole camera model geometric distortions introduced by the camera optics are neglected. The image plane projection is defined by the intrinsic parameters, the focal length f, the relative pixel size s, and the central point for the image (o_u, o_v). The relationship between the image plane coordinates $\underline{u}_m$ and a point in the camera frame $\underline{x}_m$ is then defined as follows.

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{bmatrix} f & 0 & o_u \\ 0 & sf & o_v \\ 0 & 0 & 1 \end{bmatrix} \underline{x}_m \tag{A.2}$$

$$\underline{u}_m = \begin{pmatrix} u/w \\ v/w \end{pmatrix} \tag{A.3}$$

A.2 Three-dimensional reconstruction

A point in a camera image corresponds to a ray in space along which the imaged 3D point will lie. The relationship between an image point $\underline{u}_m$ and the 3D point $\underline{x}$ is given by the inverse of the projective transformation with a parameter a_m defining the unknown position of $\underline{x}$ along the ray as follows.

$$\underline{x} = a_m \mathbf{R}^T \begin{bmatrix} 1/f & 0 & -o_u/f \\ 0 & 1/sf & -o_v/sf \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \underline{u}_m \\ 1 \end{pmatrix} - \mathbf{R}^T \underline{t} \tag{A.4}$$

$$\underline{x} = a_m \underline{n}_m + \underline{o}_m \tag{A.5}$$

For the case of two cameras the position of a point $\underline{x}$ can be reconstructed at the intersection of two rays. In practise rays will not necessarily intersect due to inexact camera calibration and a point is located at the minimum distance from each ray. The closest point on the two rays is derived by minimising the distance between them with respect to the positions a_m. The following criterion is minimised.

$$\mathcal{D} = \|x_1 - x_2\|^2 \tag{A.6}$$

A closed form solution can be derived for a_1, a_2. A triangulated point $\underline{x}$ is then defined at the mid-point between these closest ray positions, giving the point in space with minimum distance from both rays [173].

$$\frac{d\mathcal{D}}{da_m} = \underline{n}_m^T \left((a_1 \underline{n}_1 + \underline{o}_1) - (a_2 \underline{n}_2 + \underline{o}_2) \right) = 0 \tag{A.7}$$

$$a_1 = \frac{(\underline{n}_1^T \underline{n}_2 \underline{n}_2^T - \underline{n}_2^T \underline{n}_2 \underline{n}_1^T)(\underline{o}_2 - \underline{o}_1)}{(\underline{n}_1^T \underline{n}_2 \underline{n}_1^T \underline{n}_2 - \underline{n}_1^T \underline{n}_1 \underline{n}_2^T \underline{n}_2)} \tag{A.8}$$

$$a_2 = \frac{(\underline{n}_1^T \underline{n}_2 \underline{n}_1^T - \underline{n}_1^T \underline{n}_1 \underline{n}_2^T)(\underline{o}_2 - \underline{o}_1)}{(\underline{n}_2^T \underline{n}_2 \underline{n}_1^T \underline{n}_1 - \underline{n}_1^T \underline{n}_2 \underline{n}_1^T \underline{n}_2)} \tag{A.9}$$

Triangulation from two views provides a closed-form solution that minimises the distance to the inverse projection of the image points in 3D. For multiple view reconstruction a non-linear method is used to derive the point $\underline{x}$ that minimises the reprojection error to the image plane positions $\underline{u}_m$ [56]. The following criterion is minimised, where $\hat{\underline{u}}_m$ defines the projected image plane coordinates of the estimated position $\hat{\underline{x}}$.

$$\sum_{m=1}^{N_m} \|\hat{\underline{u}}_m - \underline{u}_m\|^2 \tag{A.10}$$

A closed form solution that minimises the 2D image plane error does not exist and an iterative non-linear minimisation technique is required [56]. In this work an initial estimate is obtained by triangulation from two camera views and the Guass-Newton method is used for minimisation.

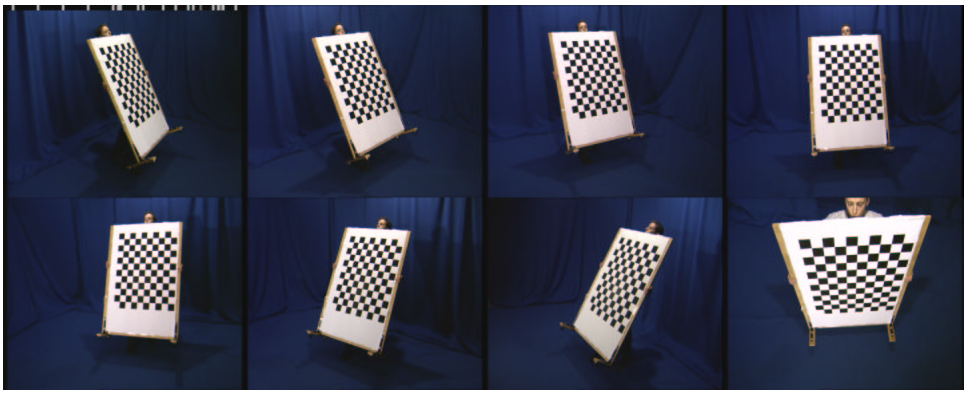


Figure A.1: Chart object for multiple view calibration.

A.3 Camera calibration

The intrinsic and extrinsic parameters for the studio cameras are calibrated using the Camera Calibration Toolbox for Matlab from MRL-Intel [3] with a planar calibration chart illustrated in Figure A.1. The source code for the implementation of the toolbox is available in the Open Source Computer Vision library distributed by Intel [10].

The reconstruction accuracy is assessed for an eight camera configuration using a sequence of recorded chart positions not used in calibration. Sub-pixel accurate corners are derived for 99 points in each chart image using the Calibration Toolbox [3]. The 3D location of each point is then reconstructed from the correspondence across multiple views as described in Section A.2. The error of the reconstructed positions is assessed using the image plane distance of the reprojected 3D postion from the original point correspondences. The mean reprojection error and the range for each camera is presented in Table A.1. The average reprojection error in all camera views is in the order of 1 pixel. This corresponds to a reconstruction error in the order of $5mm$ for PAL resolution whole body images where the depth to a subject in the scene is approximately $3m$.

Camera	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5
1	0.40 (±0.50)	1.22 (±0.90)	1.05 (±0.40)	1.28 (±0.48)	0.47 (±0.56)
2	0.28 (±0.44)	0.53 (±0.60)	0.91 (±0.50)	0.41 (±0.30)	0.37 (±0.32)
3	0.23 (±0.24)	0.36 (±0.50)	0.48 (±0.34)	0.23 (±0.30)	0.20 (±0.30)
4	0.20 (±0.26)	0.37 (±0.60)	0.24 (±0.26)	0.50 (±0.28)	0.18 (±0.32)
5	0.54 (±0.56)	0.45 (±0.40)	0.63 (±0.48)	0.57 (±0.46)	0.44 (±0.32)
6	0.69 (±0.50)	0.61 (±0.60)	1.20 (±0.52)	0.91 (±0.36)	0.86 (±0.48)
7	0.52 (±0.70)	1.53 (±1.08)	1.04 (±0.82)	2.01 (±0.72)	0.60 (±0.36)
8	0.76 (±0.84)	2.18 (±1.22)	2.50 (±1.48)	2.66 (±0.84)	1.05 (±1.00)

Table A.1: Mean reprojection error (± 2 standard deviations) in each camera image for 99 test points reconstructed from the image correspondence in all views.

Appendix B

Registration of a humanoid model to match multiple views

B.1 Generic humanoid model

The generic humanoid model used in this work consists of a single seamless mesh defining the surface shape of the body, attached to an underlying skeleton structure for animation as shown in Fig. (B.1). The model is animated using a control skeleton with 16 articulated joints to provide the gross pose of the human body.

The skeleton structure is animated as a rigid set of bone segments. The surface is animated from the skeleton using a standard vertex weighting scheme widely used in current commercial software packages. This scheme has been termed Skeletal-Subspace Deformation [108, 103]. Each bone in the skeleton is associated with a set of mesh vertices with a corresponding set of weights. The deformation of the vertices of the surface mesh is then defined as follows.

$$\underline{x}_i = \sum_b w_{ib} \mathbf{T}_b \underline{x}_i^0 \tag{B.1}$$

The animation scheme describes the deformation of the default vertex locations for a model, $\underline{x}_i^0$, in terms of the homogeneous transformation matrix for each bone $\mathbf{T}_b$ with

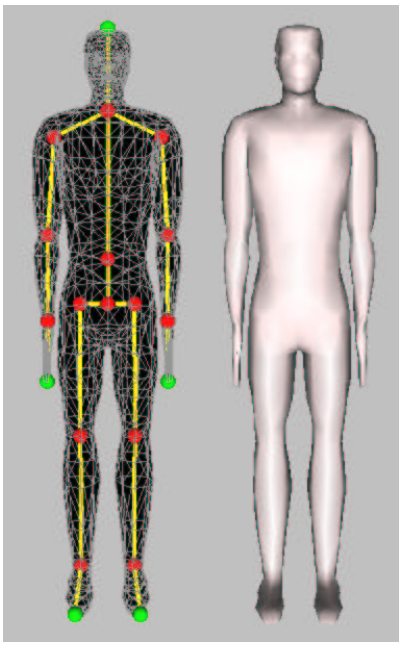


Figure B.1: The generic humanoid control model.

a weight w_{ib} associating each vertex i with each bone b. In rigid body animation only one bone affects each vertex with a corresponding weight $w_{ib} = 1$. Non-rigid surface deformation is obtained by association a vertex to multiple bones.

B.2 Model registration

The pose of the generic model is defined by the joint rotations, limb lengths and global translation of the control skeleton. The articulation of the control skeleton is defined by a 3 degree of freedom (DOF) root rotation, 3DOF rotations at the vertebrae, hips and shoulder joints, 2DOF at the clavicles and wrists, and 1DOF at the elbows, knees and ankles [72]. The dimensions of the skeleton are defined by 9DOF for the lengths of the spine, head, clavicles, upper-arm, forearm, hands, thigh, shank and foot, with left and right segments constrained to be symmetric.

The surface deformation scheme is formulated in terms of the set of joint angles $\underline{\theta}$, segment lengths $\underline{l}$, and the global translation $\underline{t}_{root}$ as follows.

$$\underline{x}_i = \sum_b w_{ib} \mathbf{T}'_b (\underline{x}_i^0 - \underline{o}_b^0) \tag{B.2}$$

$$\mathbf{T}'_b = \begin{bmatrix} exp(\underline{\mathbf{w}}_\mathbf{0}) & \underline{t}_{root} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} exp(\underline{\mathbf{w}}_\mathbf{1}) & \hat{\underline{n}}_1^0 l_1 \\ 0 & 1 \end{bmatrix} \dots \begin{bmatrix} exp(\underline{\mathbf{w}}_\mathbf{n}) & \hat{\underline{n}}_n^0 l_n \\ 0 & 1 \end{bmatrix} \tag{B.3}$$

The deformation of a vertex is now defined in terms of the rigid body transformation of a local coordinate frame centred at each bone, $\mathbf{T}'_b$. This transformation can be expressed as a concatenation of the local transformations at each joint in the skeleton in terms of the joint rotations, bone lengths and the global translation. The default joint locations are expressed as $\underline{o}_b^0$. The length of a bone is expressed as l_i and the unit direction defining the offset of a joint from a parent joint in the hierarchy is expressed as $\hat{\vec{n}}_i^0$. The rotation of the i^{th} joint in the hierarchy is expressed in an axis-angle representation as $\exp(\underline{\omega}_i)$ [70].

Bibliography

[1] *2D3: Boujou.* www.2d3.com.

[2] *Alias—Wavefront: Maya.* www.aliaswavefront.com.

[3] *Camera Calibration Toolbox.* www.vision.caltech.edu/bouguetj/calib-doc.

[4] *Canon: 3D Software Object Modeller.* www.canon.com/technology/software.

[5] *Cyberware: WB4.* www.cyberware.com.

[6] *Discreet: 3D Studio Max.* www.discreet.com.

[7] *EOS Systems: PhotoModeler.* www.photomodeler.com.

[8] *Hamamatsu: Body Line Scanner.* usa.hamamatsu.com/sys-industrial/blscanner.

[9] *NewTek: Lightwave 3D.* www.lightwave3d.com.

[10] *Open Source Computer Vision Library.* www.intel.com/research/mrl/research/opencv/.

[11] *RealViz: ImageModeler.* www.realviz.com.

[12] *SoftImage: XSI.* www.softimage.com.

[13] *Vitronic: Vitus.* www.vitus.de/english.

[14] *Wicks and Wilson: TriForm.* www.wwl.co.uk.

[15] E.H. Adelson and J.R. Bergen. The Plenoptic Function and the Elements of Early Vision. In Landy, M. and Movshon, J.A. (Eds.). *Computational Models of Visual Processing*, pages 3–20, MIT Press, Cambridge, Mass. 1991.

[16] J.K Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999.

[17] T. Akimoto, Y. Suenaga, and R.S. Wallace. Automatic creation of 3D facial models. *IEEE Computer Graphics and Applications*, 13:16–22, 1993.

[18] E. Bardinet, L. Cohen, and N. Ayache. A parametric deformable model to fit unstructured 3D data. *Computer Vision and Image Understanding*, 71(1):39–54, 1998.

[19] A. Baumberg. Reliable feature matching across widely separated views. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, I:774–781, 2000.

[20] P. Besl. Triangles as a primary representation. *in Object Representation in Computer Vision (eds. Hebert M., Ponce J., Boult T., and Gross A.), Lecture Notes in Computer Science 994*, pages 191–206, 1994.

[21] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[22] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press, Cambridge, Massachusetts, USA, 1987.

[23] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. *Proceedings of ACM SIGGRAPH*, pages 187–194, 1999.

[24] J. Blinn. Models of light reflection for computer synthesized pictures. *Computer Graphics Annual Conference Series*, pages 192–198, 1977.

[25] J. Bonet and P. Viola. Roxels: Responsibility weighted 3D volume reconstruction. *Proceedings of the ICCV Workshop, Vision Algorithms: Theory and Practice*, pages 100–115, 1999.

[26] A Broadhurst, T.W. Drummond, and R. Cipolla. A probabilistic framework for space carving. *IEEE International Conference on Computer Vision*, I:388–393, 2001.

[27] P.J. Burt and E.H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[28] B. Buxton, L. Dekker, I. Douros, and T. Vassilev. Reconstruction and interpretation of 3D whole body surface images. *Scanning*, 2000.

[29] J. Chadwick, D. Haumann, and R. Parent. Layered construction for deformable animated characters. *Computer Graphics Annual Conference Series*, 23(3):243–252, 1989.

[30] S.E. Chen. Quicktime vr - an image-based approach to virtual environment navigation. *Proceedings of ACM SIGGRAPH*, pages 29–38, 1995.

[31] S.E. Chen and L. Williams. View interpolation for image synthesis. *Proceedings of ACM SIGGRAPH*, 27:279–288, 1993.

[32] W. Chen, R. Grzeszczuk, and J. Bouguet. Light field mapping: Efficient representation and hardware rendering of surface light fields. *Proceedings of ACM SIGGRAPH*, 2002.

[33] D. Chetverikov and J. Matas. Periodic textures as distinguished regions for wide-baseline stereo correspondence. *2nd International Workshop on Texture Analysis and Synthesis*, pages 25–30, 2002.

[34] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 44–51, 2000.

[35] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, pages 114–141, 2003.

[36] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998.

[37] L.D. Cohen and I. Cohen. Finite element methods for active contour models and balloons for 2d and 3D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.

[38] G. Collins and A. Hilton. Models for character animation. *Software Focus*, 2:44–51, 2.

[39] R. Cook and K. Torrance. A reflection model for computer graphics. *ACM Transactions on Graphics*, 1:7–24, 1982.

[40] T.H. Cootes, A. Hill, C. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):355–366, 1994.

[41] I. Cox, S. Hingorani, S. Rao, and B. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, 1996.

[42] W.B. Culbertson, T. Malzbender, and G. Slabaugh. Generalized voxel coloring. *Proceedings of the ICCV Workshop, Vision Algorithms: Theory and Practice*, pages 100–115, 2000.

[43] B. Curless. From range scans to 3D models. *Computer Graphics*, 33(4), 1999.

[44] K. Dana. BRDF/BTF measurement device. *IEEE International Conference on Computer Vision*, pages 460–466, 2001.

[45] P. Debevec. Pursuing reality with image-based modeling, rendering, and lighting. *3D Structure from Images, SMILE 2000*, pages 1–14, 2000.

[46] P. Debevec, T. Hawkins, C. Tchou, H. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. *Proceedings of ACM SIGGRAPH*, pages 145–156, 2000.

[47] P. Debevec, Y. Yu, and G. Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. *9th Eurographics Rendering Workshop*, pages 105–116, 1998.

[48] P.E. Debevec. Image-based modeling and lighting. *Computer Graphics*, 33(4):46–50, 1999.

[49] P.E. Debevec and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Proceedings of ACM SIGGRAPH*, pages 11–20, 1996.

[50] H. Delingette, M. Hebert, and K. Ikeuchi. Shape representation and image segmentation using deformable surfaces. *Image and Vision Computing*, 10(3):132–144, 1992.

[51] U.R. Dhond and J.K. Aggarwal. Structure from stereo - a review. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1489–1510, 1989.

[52] M. Dooley. Anthropometric modeling programs - a survey. *IEEE Computer Graphics and Applications*, 2(9):17–25, 1982.

[53] C. Dyer. Volumetric Scene Reconstruction from Multiple Views. In Davis, L.S (ed.). *Computational Models of Visual Processing*, pages 469–489, Kluwer, Boston. 2001.

[54] P. Eisert, E. Steinbach, and B. Girod. Multi-hypothesis, volumetric reconstruction of 3-d objects from multiple calibrated camera views. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 3509–3512, 1999.

[55] R. Enciso, J. Li, D. Fidaleo, T-Y. Kim, J-Y. Noh, and U. Neumann. Synthesis of 3D faces. *International Workshop on Digital and Computational Video*, 1999.

[56] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, Massachusetts, USA, 1996.

[57] O. Faugeras, B. Hotz, H. Mathieu, T. Vieville, Z. Zhang, P. Fua, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy. Real-time correlation-based stereo: algorithm, implementation and applications. Technical Report 2013, INRIA, 1993.

[58] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. Technical Report 3021, INRIA, 1996.

[59] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6:35–49, 1993.

[60] P. Fua. Using model-driven bundle-adjustment to model heads from raw video sequences. *IEEE International Conference on Computer Vision*, pages 46–53, 1999.

[61] P. Fua and Y. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Journal of Computer Vision*, 16:35–56, 1995.

[62] P. Fua, R. Plankers, C. Miccio, and D. Thalmann. *From Image Synthesis to Image Analysis: Using Human Animation Models to Guide Feature Extraction.* EPFL-LIG Computer Graphics Lab, Switzerland, 1998.

[63] A. Fusiello, V. Roberto, and E. Trucco. Symmetric stereo with multiple windowing. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(8):1053–1066, 2000.

[64] A. Fusiello, E. Trucco, and A. Verri. Rectification with unconstrained stereo geometry. *British Machine Vision Conference*, pages 400–409, 1997.

[65] O. Grau. G. Thomas. 3D image sequence acquisition for tv & film production. *1st International Symposium on 3D Data Processing Visualization and Transmission*, pages 320–326, 2002.

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

[67] S. Gibbs, C. Arapis, C. Breiteneder, V. Lalioti, S. Mostafawy, and J. Speier. Virtual studios: The state of the art. *Eurographics State of the Art Reports*, pages 63–86, 1996.

[68] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. The lumigraph. *Proceedings of ACM SIGGRAPH*, pages 43–54, 1996.

[69] V. Gouet, P. Montesinos, and D. Pel. A fast matching mehotd for color uncalibrated images using differential invariants. *British Machine Vision Conference*, pages 367–376, 1998.

[70] F.S. Grassia. Practical parameterization of rotations using the exponential map. *The Journal of Graphics Tools*, 3(3), 1998.

[71] N. Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11):21–29, 1986.

[72] M.R. Grosso, R. Quach, E. Otani, J. Zhao, S. Wei, P.H. Ho, J. Lu, and N.I. Badler. Anthropometry for computer graphics human figures. Technical Report MS-CIS-89-71, University of Pennsylvania, Dept. of Computer and Information Science, 1989.

[73] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. *Proceedings of ACM SIGGRAPH*, pages 55–66, 1998.

[74] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey88*, pages 147–152, 1988.

[75] R. Hartley and A. Zisserman. *Multiple-View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[76] X. He, K. Torrance, F. Sillon, and D. Greenberg. A comprehensive physical model for light reflection. *Computer Graphics Annual Conference Series*, pages 175–186, 1991.

[77] P.S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications*, pages 56–67, 1986.

[78] A. Hilton, D. Beresford, T. Gentils, R. Smith, and W. Sun. Virtual people: Capturing human models to populate virtual worlds. *IEEE International Conference on Computer Animation*, pages 174–185, 1999.

[79] A. Hilton, D. Beresford, T. Gentils, R. Smith, W. Sun, and J. Illingworth. Whole-body modelling of people from multiview images to populate virtual worlds. *The Visual Computer*, 16(7):411–436, 2000.

[80] A. Hilton, J. Starck, and G. Collins. From 3D shape capture to animated models. *1st International Symposium on 3D Data Processing Visualization and Transmission*, pages 246–255, June 2002.

[81] B.K.P Horn. *Robot Vision*. MIT Press, Cambridge, Massachusetts, 1986.

[82] X. Ju and J.P Siebert. Conforming generic animatable models to 3D scanned data. *Scanning*, 2001.

[83] I. Kakadiaris, D. Metaxas, and R. Bajcsy. Inferring 2d object structure from the deformation of apparent contours. *Computer Vision and Image Understanding*, pages 129–147, 1997.

[84] I.A. Kakadiaris and D. Metaxas. 3d human body model acquisition from multiple views. *IEEE International Conference on Computer Vision*, pages 618–623, 1995.

[85] I.A. Kakadiaris and D. Metaxas. Three-dimensional human body model acquisition from multiple views. *International Journal of Computer Vision*, 30(3):191–218, 1998.

[86] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, 1994.

[87] T. Kanade, P.W. Rander, and P.J. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*, 4(1):34–47, 1997.

[88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[89] J. Konrad and Z-D. Lan. Dense disparity estimation from feature correspondences. *IS&T/SPIE Symposium on Electronic Imaging Stereoscopic Displays and Virtual Reality Systems*, 2000.

[90] P.G. Kry, D.L. James, and D.K. Pai. Eigenskin: Real time large deformation character skinning in hardware. *Proceedings of ACM SIGGRAPH Symposium on Computer Animation*, 2002.

[91] T. Kurihara and K. Arai. A transformation method for modeling and animation of the human face from photographs. *Computer Animation. Magnenat-Thalmann N and Thalmann D (Eds.)*, Springer-Verlag, Berlin. 1991.

[92] K. Kutulakos and S. Seitz. A theory of shape by space carving. Technical Report 692, University of Rochester, 1998.

[93] K.N. Kutulakos. Approximate n-view stereo. *Proceedings of the European Conference on Computer Vision*, pages 67–83, 2000.

[94] J-O Lachaud and A. Montanvert. Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction. *Medical Image Analysis*, 3(2):187–207, 1999.

[95] A. Laurentini. The visual hull concept for silhouette based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[96] S. Laveau and O. Faugeras. 3d scene representation as a collection of images and fundamental matrices. Technical Report 2205, INRIA, 1994.

[97] W-S. Lee, M. Escher, G. Sannier, and N. Magnenat-Thalmann. MPEG-4 compatible faces from orthogonal photos. *Computer Animation*, pages 186–194, 1999.

[98] W-S. Lee, J. Gu, and N. Magnenat-Thalmann. Generating animatable 3D virtual humans from photographs. 19(3), 2000.

[99] Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. *Proceedings of ACM SIGGRAPH*, pages 55–62, 1995.

[100] M. Levoy and P. Hanrahan. Light field rendering. *Proceedings of ACM SIGGRAPH*, pages 31–42, 1996.

[101] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project. *Proceedings of ACM SIGGRAPH*, pages 131–144, 2000.

[102] B. Levy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *Proceedings of ACM SIGGRAPH*, 2002.

[103] J.P. Lewis, M. Cordner, and N. Fong. Pose space deformations: A unified approach to shape interpolation a nd skeleton-driven deformation. *Proceedings of ACM SIGGRAPH*, 2000.

[104] A. Lippman. Movie-maps: An application of the optical videodisc to computer graphics. *Computer Graphics Annual Conference Series*, 1980.

[105] S.A. Lloyd. A dynammic programming algorithm for binocular vision. *GEC Journal of Research*, 3(1):18–24, 1985.

[106] W.E. Lorensen and H.E. Cline. Marching cubes: a high resolution 3D surface reconstruction algorithm. *SIGGRAPH Conference Proceedings*, 21(4):163–169, 1987.

[107] C. Lurig, L. Kobbelt, and T. Ertl. Deformable surfaces for feature based indirect volume rendering. *Computer Graphics International*, pages 752–760, 1998.

[108] N. Magnenat-Thalmann, R. Laperriere, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. *Proceedings of Graphics Interface*, pages 26–33, 1988.

[109] R. Malladi, J. Sethian, and B. Vemuri. Shape modelling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–174, 1995.

[110] S. Marschner, E. Lafortune, S. Westin, K. Torrance, and D. Greenberg. Image-based brdf measurement. *Applied Optics*, 39(16):2592–2600, 2000.

[111] W.N. Martin and J.K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, 1983.

[112] W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. *Proceedings of Eurographics Workshop on Rendering.*

[113] W. Matusik, C. Buehler, R. Raskar, S.J. Gortler, and L. McMillan. Image-based visual hulls. *Proceedings of ACM SIGGRAPH*, pages 369–374, 2000.

[114] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan. Image-based 3D photography using opacity hulls. *Proceedings of ACM SIGGRAPH*, pages 427–437, 2002.

[115] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[116] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Proceedings of ACM SIGGRAPH*, pages 39–46, 1995.

[117] T. Moeslund and E Granum. A survey of computer vision based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

[118] S. Moezzi, A. Katkere, D. Kuramura, and R. Jain. Reality modeling and visualization from multiple video sequences. *IEEE Computer Graphics and Applications*, 16(6):58–63, 1996.

[119] S. Moezzi, L.C. Tai, and P. Gerard. Virtual view generation for 3D digital video. *IEEE Multimedia*, 4(1):18–25, 1997.

[120] J. Montagnat and H. Delingette. Volumetric medical images segmentation using shape constrained deformable models. pages 13–22, 1997.

[121] J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing*, 19:1023–1040, 2001.

[122] J. Montagnat and O. Faugeras. Spatial and temporal shape constrained deformable surfaces for 3D and 4d medical image segmentation. Technical Report 4078, INRIA, 2000.

[123] P.J. Narayanan, P. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. *IEEE International Conference on Computer Vision*, pages 3–10, 1998.

[124] W. Niem. Robust and fast modelling of 3D natural objects from multiple views. *SPIE Proceedings Image and Video Processing II*, 2182:388–397, 1994.

[125] W. Niem and H. Broszio. Mapping texture from multiple camera views onto 3d-object models for computer animation. *Proceedings of the International Workshop on Stereoscopic and Three Dimensional Imaging*, 1995.

[126] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, 1993.

[127] F. Parke. Computer generated animation of faces. *ACM National Conference*, 1972.

[128] F. Parke. *A Parametric Model of Human Faces*. PhD thesis, University of Utah, Salt Lake City, 1974.

[129] F. Pighin, J. Hecker, D. Lischinski, and R. Szeliski. Synthesizing realistic facial expressions from photographs. *Proceedings of ACM SIGGRAPH*, pages 75–84, 1998.

[130] D. Piponi and G. Borshukov. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. *Proceedings of ACM SIGGRAPH*, pages 471–477, 2000.

[131] R Plankers and P. Fua. Articulated soft objects for video-based body modeling. *IEEE International Conference on Computer Vision*, pages 394–401, 2001.

[132] M. Price, J. Chandaria, O. Grau, G.A. Thomas, D. Chatting, J. Thorne, G. Milnthorpe, P. Woodward, L. Bull, E-J. Ong, A. Hilton, J. Mitchelson, and J. Starck. Real-time production and delivery of 3D media. *Proceedings of the International Broadcasting Convention*, 2002.

[133] P. Pritchett and A. Zisserman. Wide baseline stereo matching. *IEEE International Conference on Computer Vision*, pages 754–760, 1998.

[134] K. Pulli, M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro, and W. Stuetzle. View-based rendering: Visualizing real objects from scanned range and color data. *Eurographics workshop on Rendering*, pages 23–34, 1997.

[135] A. Rangarajan, H. Chui, and F. Bookstein. The softassign procrustes matching algorithm. *Information Processing in Medical Imaging*, pages 29–42, 1997.

[136] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86(11):2210–2239, 1998.

[137] Y. Sato, M. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. *Proceedings of ACM SIGGRAPH*, pages 379–387, 1997.

[138] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7–42, 2002.

[139] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. *Computer Graphics Annual Conference Series*, 20(4):151–160, 1986.

[140] C.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1067–1073, 1997.

[141] C.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):1–23, 1999.

[142] S.M. Seitz and C.R. Dyer. View morphing. *Proceedings of ACM SIGGRAPH*, pages 21–30, 1996.

[143] H. Seo and N. Magnenat-Thalmann. An automatic modeling of human bodies from sizing parameters. *ACM SIGGRAPH Symposium on Interactive 3D Graphics*, pages 19–26, 2003.

[144] J.W. Shade, S.J. Gortler, L-W. He, and R. Szeliski. Layered depth images. *Proceedings of ACM SIGGRAPH*, pages 231–242, 1998.

[145] S.A. Shafer. Using color to separate reflection components. *Color Research Applications*, 10(4):210–218, 1985.

[146] J. Shen and D. Thalmann. Interactive shape design using metaballs and splines. *Proceedings of Implicit Surfaces*, pages 187–196, 1995.

[147] H-Y Shum and R. Szeliski. Panoramic image mosaics. Technical Report MSR-TR-97-23, Microsoft Research, 1997.

[148] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer. A survey of methods for volumetric scene reconstruction from photographs. *Proceedings of the Joint IEEE TCVG and Eurographics Workshop*, pages 81–100, 2001.

[149] J. Starck, G. Collins, R. Smith, A. Hilton, and J. Illingworth. Animated statues. *Machine Vision and Applications, Special Issue on Human Modeling, Analysis, and Synthesis*, 2002.

[150] J. Starck and A. Hilton. Model-based multiple view reconstruction of people. *IEEE International Conference on Computer Vision*, 2003.

[151] J. Starck and A. Hilton. Towards a 3D virtual studio for human appearance capture. *Vision, Video, and Graphics*, 2003.

[152] J. Starck, A. Hilton, and J. Illingworth. Human shape estimation in a multi-camera studio. *British Machine Vision Conference*, 2:573–582, 2001.

[153] J. Starck, A. Hilton, and J. Illingworth. Reconstruction of animated models from images using constrained deformable surfaces. *10th International Conference on Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science*, 2301:382–391, 2002.

[154] E. Steinbach, B. Girod, P. Eisert, and A. Betz. 3d object reconstruction using spatially extended voxels and multi-hypothesis voxel coloring. *Proceedings of the International Conference on Pattern Recognition*, 2000.

[155] C. Sun. Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. *International Journal of Computer Vision*, 47(1/2/3):99–117, 2002.

[156] R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics and Image Processing: Image Understanding*, 58(1):23–32, 1993.

[157] R. Szeliski. Stereo algorithms and representations for image-based rendering. *British Machine Vision Conference*, pages 314–328, 1999.

[158] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32(1):45–62, 1999.

[159] R. Szeliski and D. Tonnesen. Surface modeling with orientated particle systems. *Computer Graphics*, 26(2):185–194, 1992.

[160] D. Tell and S. Carlsson. Wide baseline point matching using affine invariants computed from intensity profiles. *Proceedings of the European Conference on Computer Vision*, pages 814–828, 2000.

[161] D. Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):417–438, 1988.

[162] D. Terzopoulos. From physics-based representation to functional modeling of highly complex objects. *NSF-ARPA Workshop on Object Representation in Computer Vision*, pages 347–359, 1994.

[163] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:703–714, 1991.

[164] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics Annual Conference Series*, 21(4):205–214, 1987.

[165] D. Terzopoulos and K. Waters. Physically-based facial modeling, analysis, and animation. *Visualization and Computer Animation*, 1:73–80, 1990.

[166] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models for 3D object reconstruction. *International Journal of Computer Vision*, 1:211–221, 1987.

[167] D. Thalmann. Human modelling and animation. *Eurographics State of the Art Reports*, 1993.

[168] D. Thalmann, J. Shen, and E. Chauvineau. Fast human body deformations for animation and vr applications. *Proceedings of Computer Graphics International*, pages 166–174, 1996.

[169] A. Thompson, J. Brown, J. Kay, and D. Titterington. A study of methods of choosing the smoothing parameter in image restoration by regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):326–339, 1991.

[170] A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Winston, Washington, DC, 1977.

[171] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. *Vision Algorithms: Theory and Practise. Triggs W., Zisserman A, and Szeliski, R. (Eds.)*, pages 298–375, 2000.

[172] E. Trucco, A. Fusiello, and V. Roberto. Robust motion and correspondence of noisy 3-D point sets with missing data. *Pattern Recognition Letters*, pages 889–898, 1999.

[173] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, New Jersey, USA, 1998.

[174] G. Turk and J.F. O'Brien. Shape transformation using variational implicit functions. *Proceedings of ACM SIGGRAPH*, pages 335–342, 1999.

[175] S. Vedula, S. Baker, and T. Kanade. Spatio-temporal view interpolation. *Eurographics Workshop on Rendering*, pages 1–11, 2002.

[176] S. Vedula, P. Rander, H. Saito, and T. Kanade. Modeling, combining, and rendering dynamic real-world events from image sequences. *Proceedings of Virtual Systems and Multimedia*, pages 323–344, 1998.

[177] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003.

[178] G. Ward. Measuring and modeling anisotropic reflectance. *Computer Graphics Annual Conference Series*, pages 265–273, 1992.

[179] R. Whitaker. Volumetric deformable models: Active blobs. *Visualization in Biomedical Computing*, pages 122–134, 1994.

[180] J. Wingbermuhle, S. Weik, and A. Kopernik. Highly realistic modeling of persons for 3D videoconferencing systems. *IEEE Workshop on Multimedia Signal Processing*, pages 286–291, 1997.

[181] G. Wolberg. Image morphing: A survey. *The Visual Computer*, 14(8/9):360–372, 1998.

[182] D. Wood, D. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle. Surface light fields for 3D photography. *Proceedings of ACM SIGGRAPH*, pages 287–296, 2000.

[183] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. *Proceedings of ACM SIGGRAPH*, pages 215–214, 1999.