A Surface Deformation Framework for 3D Shape Recovery

by

Yusuf Sahillioğlu

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science
in
Electrical and Computer Engineering

Koç University
June 2006

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Yusuf Sahillioğlu

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Assist. Prof. Yücel Yemez (Advisor)

_____

Prof. Reha Civanlar

_____

Assist. Prof. Çağatay Başdoğan

Date: _____

ABSTRACT

We present a generic surface deformation framework for the problem of 3D shape recovery. A spatially smooth and topologically plausible surface mesh representation is constructed via a surface evolution based technique, starting from an initial model. The initial mesh, representing the bounding surface, is refined or simplified where necessary during surface evolution using a set of local mesh transform operations so as to adapt it to the local properties of the object surface. The final mesh obtained at convergence can adequately represent the complex surface details such as bifurcations, protrusions and large concavities. We demonstrate the performance of our deformation framework on the problem of shape from silhouette and its fusion with shape from optical triangulation for 3D reconstruction of static objects. The framework is in fact very general and applicable to any kind of data that can be used to infer 3D geometry. Since the approach we take for surface deformation is Lagrangian, that can track changes in connectivity and geometry of the deformable mesh during surface evolution, the proposed framework can also be used to build efficient time-varying representations of dynamic scenes.

Keywords: 3D reconstruction, shape from silhouette, shape from structured light, shape from optical triangulation, surface deformation, surface evolution.

ÖZETÇE

Bu tez çalışması üç boyutlu şekil yakalama problemi için yüzey deformasyonuna dayalı bir çatı yöntem sunmaktadır. Bu yöntem, bir başlangıç modelinden yola çıkarak, yüzey evrimine dayalı bir teknikle uzamsal olarak yumuşak ve topolojisi doğru bir yüzey örgü gösterimi oluşturur. Şekli kuşatan yüzeyi temsil eden başlangıç örgü modeli, yüzey evrimi sırasında yerel örgü dönüşüm işlemleri ile gerektiği şekilde inceltilir ya da sadeleştirilir, böylelikle modelin nesne yüzeyinin yerel özelliklerine uyum göstermesi sağlanır. Yakınsama durumunda elde edilen son örgü modeli, çatallanma, çıkıntı ve kovuk gibi karmaşık yüzey detaylarını uygun bir şekilde ifade edebilir. Önerdiğimiz deformasyon yöntemini, statik (devinimsiz) nesnelerin modellenmesi amacıyla, silüetten şekil geriçatma ve bunun optik üçgenleştirme ile birleştirilmesi problemlerine uyguladık. Elde edilen deney sonuçları tezde sunulmaktadır. Önerdiğimiz yöntem aslında genel bir deformasyon çatısı tarif eder ve üç boyut bilgisi veren her türlü veriye uygulanabilir. Deformasyonu modellemek için benimsediğimiz Lagrange yaklaşımı ile örgü modelin yüzey evrimi sırasında değişebilen bağlanırlığı ve geometrisi izlenebilir, ve dolayısıyla önerdiğimiz çatı, dinamik sahnelerin zamanla değişen gösterimlerinin verimli bir şekilde oluşturulması için de kullanılabilir.

Anahtar Sözcükler: 3B geriçatma, silüetten şekil, yapısal ışıktan şekil, optik üçgenleştirme ile şekil, yüzey deformasyonu, yüzey evrimi.

ACKNOWLEDGEMENTS

I am glad to have the chance of being supervised by Assist. Prof. Yücel Yemez during my M.S. adventure. My sagacious guide has always shown me the right direction whenever I got stuck in the 3D world. Our scientific and memorable conversations have significantly contributed to this thesis. I owe him much.

I would like to thank Prof. Reha Civanlar and Assist. Prof. Çağatay Başdoğan for accepting to take part in my thesis committee. I am also grateful to Mr. Başdoğan for introducing me Open Inventor toolkit without which the graphics programming part would not be so fun.

Another academic thank is for Can James Wetherilt who has helped me to finish this work on time by sharing his test data with me.

I was lucky to work in a spacious office with a cool view that encouraged me in my bad times. Besides, nice friends in it have usually been the guys who have cared to watch my 3D demonstrations. Thanks go to ENG Z20.

Last but definitely not least, my family deserves serious thanks. My father, Edip Sahillioğlu, has always been and will always be with us although he sadly passed away in March 1998. He is my inspiration and I am sure he is proud of me for this degree which he had already received from UMIST, England. My mother, Nevin Sahillioğlu, is the best mother in the world who supported me to study Computer Engineering and who has always been with me throughout that education with her advises and delicious foods. It is the best feeling to feel her presence. My brother Erdem gets the last thank for helping me relieve.

TABLE OF CONTENTS

List of Tables viii

List of Figures ix

Nomenclature xii

Chapter 1: Introduction 1

1.1 State-of-the-Art. 2

1.1.1 3D Reconstruction Techniques 2

1.1.2 Surface Deformation 4

1.1.3 Fusion. 5

1.2 Contribution. 6

1.3 Overview and Organization. 8

Chapter 2: Surface Deformation 13

2.1 Lagrangian Deformation 13

2.2 Local Mesh Transform Operations 15

2.3 Topological Problems. 17

2.4 Surface Evolution. 19

Chapter 3: Shape from Silhouette **21**

3.1 Silhouette-based Deformation 22

3.2 Fine Tuning. 24

3.3 Algorithm. 26

3.4 Refinement. 27

3.5 Experimental Results 29

Chapter 4: Fusion - Shape From Silhouette and Optical Triangulation **39**

4.1 Shape from Optical Triangulation. 39

4.2 Fusion. 42

4.2.1 Assignment of Carvers. 44

4.2.2 Computation of External Force. 47

4.2.3 Update of Carvers 50

4.2.4 Algorithm 51

4.3. Experimental Results 52

Chapter 5: Conclusions and Future Work **65**

Bibliography **67**

Vita

LIST OF TABLES

Table 3.1: Structural mesh information and execution times. 38
Table 4.1: Number of images taken for each object. 53
Table 6.2: Structural mesh information and execution times for fusion. 64

LIST OF FIGURES

Figure 1.1: Block diagram of the overall reconstruction system. 9
Figure 1.2: The data acquisition setup. 10
Figure 1.3: Image of the object with saturated background. 11
Figure 2.1: Effect of regularization by the internal force. 15
Figure 2.2: Edge collapse, split and flip operations. 16
Figure 2.3: Illegal edge collapse. 18
Figure 2.4: Infinite loop problem. 16
Figure 2.5: Illegal edge flip. 19
Figure 2.6: Illustration of the surface deformation algorithm. 20
Figure 3.1: Vertex categorization on a single silhouette. 24
Figure 3.2: Fine-tuning. 25
Figure 3.3: Binary subdivision procedure to accurately locate the position of a vertex. . 26
Figure 3.4: Refinement with further split operation. 28
Figure 3.5: Original images of Elephant, Jumping Man, and Hand objects. 29
Figure 3.6: Reconstruction of the Elephant object at low resolution. 30
Figure 3.7: Various views from the final Elephant reconstruction at low resolution after two steps of fairing. 31
Figure 3.8: Reconstruction of the Elephant object at high resolution. 31
Figure 3.9: Various views from the final Elephant reconstruction at high resolution after two steps of fairing. 32
Figure 3.10: (Top vs. bottom) Low vs. high resolution final reconstructions. 32

Figure 3.11: Reconstruction of the Jumping Man. 33
Figure 3.12: The same deformation iterations of Fig. 3.11, as observed from a different angle. 34
Figure 3.13: Various views from the final Jumping Man reconstruction after two steps of fairing. 34
Figure 3.14: Reconstruction of the Hand object. 35
Figure 3.15: Refinement of the Hand model. 36
Figure 3.16: Views from various iterations of the deformation process after the refinement procedure. 36
Figure 3.17: Various views from the final Hand reconstruction after two steps of fairing. 37
Figure 4.1: Top view of a slice from an ellipsoidal object with two separate scanner configurations. 41
Figure 4.2: Illustration of the fusion process. 44
Figure 4.3: Carver assignment. 46
Figure 4.4: Scan line neighborhood of a triangle, used for update of carvers. 47
Figure 4.5: Illustrations of the two alternatives to compute an estimate of the distance of the deformable mesh to the object boundary. 50
Figure 4.6: Original images of Elephant, Greek, and Hand objects. 53
Figure 4.7: Initial carver assignment on silhouette models of the Elephant. 54
Figure 4.8: Triangulated range surfaces obtained from two separate scans of optical triangulations. 55
Figure 4.9: Silhouette models (top) vs. fused reconstructions (bottom) of the Elephant. 55
Figure 4.10: Fusion results of the Elephant from various views. 56
Figure 4.11: The optical triangulation surface overlaid by the silhouette model of the Greek object. 57
Figure 4.12: Initial carver assignment on silhouette models of the Greek. 58

Figure 4.13: Range surface of the Greek object, obtained by optical triangulation 58
Figure 4.14: Silhouette models (top) vs. fused reconstructions (bottom) of the Greek object. 59
Figure 4.15: Fusion results of the Greek object from various views. 59
Figure 4.16: (Top vs. bottom) Low vs. high resolution reconstructions. 60
Figure 4.17: (Left vs. right) Low vs. high resolution reconstructions in wireframe. 60
Figure 4.18: Initial carver assignment on silhouette models of the Hand. 61
Figure 4.19: Triangulated range surfaces obtained from three separate scans of optical triangulation. 62
Figure 4.20: Silhouette models (top) vs. fused reconstructions (bottom) of the Hand object. 62
Figure 4.21: Fusion results of the Hand from various views. 63
Figure 4.22: Fusion results by using two alternative ways of computing the external force as given by Equations 4.2 and 4.3 of Chapter 4. 63

NOMENCLATURE

S	Deformable mesh
P	Vertex of the deformable mesh
B	Surface boundary
$\mathbf{F}_{\text{int}}$	Internal force
$\mathbf{F}_{\text{ext}}$	External force
E	Energy
E_{int}	Internal energy
E_{ext}	External energy
v	Strength of the external force
$\mathbf{N}$	Surface normal
$\mathbf{p}$	Vector representation of the vertex
ε_{min}	Minimum edge length
ε_{max}	Maximum edge length
G	Bilinearly interpolated value of a sub-pixelic point on the discrete image raster
$f(\cdot)$	Isolevel function
$L(\cdot)$	State of a vertex – VERY-IN, IN, ON, OUT or VERY-OUT
C_i	Carver
T_i	Triangle
R_i	Range point
S_{T_i}	Triangle plane
S_{R_i}	Tangent plane at a range point

Chapter 1

INTRODUCTION

Deformation models have widely been used in various problems of 3D (three-dimensional) computer graphics and vision such as shape recovery, animation, editing, object tracking and segmentation [1]. They in general yield smooth and robust representations that can successfully capture and preserve semantics of the data with well established mathematical foundations. This thesis considers the use of deformation models for surface extraction in 3D object reconstruction schemes.

3D reconstruction of real objects has been an active research field of computer vision and graphics during the past two decades. There are many application areas concerned with 3D reconstruction among which virtual reality, digital preservation of cultural heritage, machine vision, medical imaging are the most common. In general, techniques employed for 3D digitization can be collected under two groups: active and passive [9]. Active methods make use of calibrated light sources such as lasers or coded light most typical example of which is the shape from structured light method. Passive methods, on the other hand, rely solely on 2D photographs of the scene to extract surface information. Among the most common that fall into this category are the techniques known as shape from silhouette, shape from stereo, and shape from shading.

Although the techniques proposed for 3D reconstruction are numerous and the relevant literature on each individual technique is abundant, there is relatively very little work reported in the literature on the fusion of different reconstruction techniques. Since each reconstruction technique has certain benefits as well as some drawbacks as compared to

other methods, combining different methods can significantly improve accuracy and robustness.

The goal of most 3D reconstruction schemes is to achieve a polygonal (e.g. triangular) mesh model that represents the object surface geometry. The surface extraction technique utilized in these schemes vary with the type of the reconstruction method (active or passive) as well as with the type of the information used to infer 3D geometry (silhouette, texture, shade, range images, etc.). Most of the techniques existing in the literature do not always produce topologically correct triangulations. Holes and non-manifold triangulations are common to most surface reconstruction results. The marching cubes algorithm [25] for instance, which is commonly used for surface triangulation in both active and passive reconstruction schemes, can guarantee topologically correct triangulations only locally.

In this thesis work, we develop a generic framework based on surface deformation, that can be used to build accurate, robust and topologically correct 3D models of complex real objects. Although the proposed surface extraction framework is applicable to any type of reconstruction method, we specifically address the shape from silhouette technique and its fusion with shape from optical triangulation.

1.1 State-of-the-Art

1.1.1 3D Reconstruction Techniques

Shape from silhouette technique, as a passive reconstruction method, basically draws the shape information by first back-projecting multiple silhouettes into world space as conical volumes and then intersecting these cones to generate a volumetric visual hull. Early examples of this technique were presented by Chien and Aggarwal [6], [7] and later much improvement has been established concerning efficiency and space constraint matters

[34], [8], [2], [3], [4]. In general, the technique's strength lies in its simplicity, efficiency and accuracy especially when applied to convex shapes. The drawback of this method is that it fails to capture hidden concavities.

Methods of shape from stereo rely on finding correspondences between multiple images of the same scene (or object) using texture or color information [16]. This feature makes such techniques very sensitive to lighting conditions and renders them less effective as stand-alone methods. On the other hand, shape from shading methods are based on the diffusing properties of Lambertian surfaces [17]. They require controlled environments where the illumination of the object space and object reflectance must be known. Textured objects pose great problems and the proposed techniques are found to be mathematically unstable.

Shape from structured light, as an active triangulation technique, produces accurate point clouds. Due to inherent camera and light occlusions, it poses problems to create complete and watertight reconstructions. Park et al. [13] explain how to integrate several laser projectors to reduce light occlusions, Davis and Chen [15] propose an acquisition system to minimize calibration complexity and cost, and Liska and Sablatnig [14] review next view planning techniques to optimize the surface coverage. Levoy et al. [12] have used large-scale enhanced acquisition systems to overcome the occlusion problem presenting very accurate and successful results. The problem of integrating aligned images while producing hole-free reconstructions is addressed by both Rusinkiewicz et al. [11] and Curless and Levoy [10]. Rusinkiewicz's work is based on the design of a real-time acquisition system allowing to scan objects faster and with greater ease than conventional model acquisition pipelines. Although successful, Curless and Levoy [10] point out the need for added carving especially where the range data are either occluded or scarce. Parts of the reconstructed object surface may sometimes be visible by the sensor only at sharp angles, making such reconstructed sections of the model unreliable.

Most of the active scene capture technologies become inapplicable in the dynamic case. The shape from optical triangulation, which is the most accurate active capturing technique, can not for example be used when the object is in motion [10]. A plausible alternative might be the use of passive techniques for time-varying scene capture [19], [24],[28]. Such multicamera systems infer the 3D shape from its silhouettes (in the case of object reconstruction) and/or from multistereo texture and color consistency. However, as pointed out before, silhouette-based techniques are not capable of capturing hidden concavities of the object surface whereas stereo-based techniques suffer from accuracy problems. Yet, when the object to be captured is not very complicated in shape, passive techniques may yield robust, hole-free and complete reconstructions of an object in motion.

1.1.2 Surface Deformation

Surface deformation models existing in the computer vision literature for 3D shape recovery can be grouped under two different categories: 1) Level sets (the Eulerian approach) and 2) Active contours (the Lagrangian approach). The active contour models, or so called "snakes", were first developed by Kass et al. [22] for detection of salient features in 2D image analysis and then extended by Terzopoulos et al. [29] to 3D for the surface recovery problem. In the Lagrangian approach, an initial parametric contour or surface is made to evolve towards the boundary of the object to be detected under the guidance of some application-specific external and internal forces that try to minimize an overall energy. The original snake model was not designed to handle possible topological changes that might occur during surface/contour evolution, nor was it capable to represent protrusions and bifurcations of complex shapes. It was nevertheless improved by many successors and has found applications in various domains of computer vision [30].

The level set technique on the other hand was first proposed by Malladi et al. [23] as an alternative to the classical snake approach in order to overcome its drawbacks mentioned above. This technique favors the Eulerian formulation with which the object shape is implicitly embedded into a higher dimensional space as the level set solution of a time varying shape function. The level-set technique is computationally very expensive and inevitably necessitates a parallel implementation especially for the 3D surface recovery problem [19]. Although it can implicitly handle topological changes in geometry, the topology control is often very difficult to achieve. Moreover, with the level set approach, the explicit connectivity information of the initial shape model is lost through the iterations between the initial state and its convergence. Thus the level set technique becomes inapplicable to track objects in motion and to build efficient time-varying representations.

The active contour (snake) approach was successfully used before to address the 3D object reconstruction and modeling problem. In [21] for instance, the fusion of shape from silhouette and shape from stereo techniques is considered. The initial surface model obtained by shape from silhouette is carved further by surface deformation to amend missing concavities. In [31], a generic PDE-based methodology is proposed for surface recovery, that can handle multiple types of data such as 3D point clouds, 2D images and volumetric data.

1.1.3 Fusion

The robust but cavity-insensitive output of the shape from silhouette technique constitutes a solid initial foundation for further volume carving or mesh deformations, depending on the type of the representation employed. This observation has led to several attempts to combine shape from silhouette with several other techniques which alone do not produce complete or fully reliable model reconstructions. The common strategy for

integrating silhouette information with others is to start off with an initial estimate of the object shape in the form of a convex hull obtained from its silhouettes and gradually recover the cavities with a cavity-sensitive method.

Several workers as in [20], [27] fuse shape from silhouette and shape from stereo in a volumetric fashion while others as in [21] adhere to deformation models for further enhancing description of the object mesh initially obtained from silhouettes. Savarese et al. [26] integrate shape from shadow and shape from silhouette techniques in a volumetric carving fashion, but the results show that the obtained reconstructions can not compete with the accuracy of active methods.

Tosovic et al. [5] have attempted to combine shape from silhouette and shape from optical triangulation fully favouring an octree-based fusion that builds a single model without the need to convert the data structure of one method to that of the other. However, it lacks many specifications on crucial points such as how the volume triangulation is conducted and fails to produce clear results. The only method in the literature that underpins the feasibility of combining shape from silhouette with shape from optical triangulation as a successful method is the one proposed by Yemez et al. in [32]. This technique carries out the fusion volumetrically by carving the superfluous parts of the visual hull (from the silhouettes) with the range points (from the optical triangulation) and yields very satisfactory results as long as the resolution of the carved volumetric octree matches with that of the sampled range points.

1.2 Contribution

There are two main contributions of this thesis work to the 3D object reconstruction literature:

- **i.** We develop a generic snake-based surface deformation framework, that can be used to build topologically correct 3D mesh representations of complex real objects. The surface deformation framework is compatible with any reconstruction method, active or passive, so that it can handle any type of data to infer 3D geometry such as silhouettes, texture images and range data.
- **ii.** Based on the developed surface deformation framework, we propose a technique to combine two reconstruction techniques, namely shape from silhouette and shape from optical triangulation, for a more accurate and robust object reconstruction.

The surface deformation framework that we propose has much in common with the two closely related works [21] and [31]. The work in [21] addresses the fusion of two passive methods, shape from silhouette and shape from stereo, by deforming the initial silhouette surface towards its hidden concavities incorporating stereo information. In [31], the authors use a PDE-driven surface evolution methodology to develop a generic surface deformation technique that can handle multiple types of data to infer 3D geometry. Both techniques rely on snake-based deformation of mesh models, but do not take into account topological problems that may arise during surface evolution. We incorporate to the well known snake-based deformation framework, the use of local mesh operators to control the topology so that the deformable mesh is guaranteed to remain free of degenerate and non-manifold triangles.

So far there have been only two attempts to combine the passive shape from silhouette technique with the active shape from optical triangulation. The work presented in [5]

addresses only the object volume reconstruction problem. The acquired range images are first converted to binary images so that the white pixels belong "possibly" to the object volume and the black pixels correspond to the background. The binary images are then directly fused with the silhouette information under an octree. This yields a fast algorithm for volumetric fusion, which is however liable to various topological problems. The paper does not address these problems and the presented method seems to be an approximate and ad-hoc technique that can not be used for reconstruction of complex objects.

In [32], Yemez et al. present a hybrid surface reconstruction method which is based on volume carving followed by isosurface merging. A volumetric octree representation is first obtained from the silhouette images and then carved by range points to amend the missing cavity information. An isolevel value on each surface cube of the carved octree structure is accumulated using local surface triangulations obtained separately from range data and silhouettes. The marching cubes algorithm is then applied for triangulation of the volumetric representation. With this technique, it is possible to obtain satisfactory surface reconstructions of complex objects. However, the volume carving process is sensitive to sensor noise and to deficient range data, and thus does not permit high resolution reconstructions. The technique that we develop in this thesis overcomes this limitation by addressing the fusion problem in a surface deformation framework.

1.3 Overview and Organization

The block diagram of our overall reconstruction system is given in Fig. 1.1 The 5 basic tasks are data acquisition, silhouette extraction, deformation for shape from silhouette, optical triangulation and deformation for fusion.

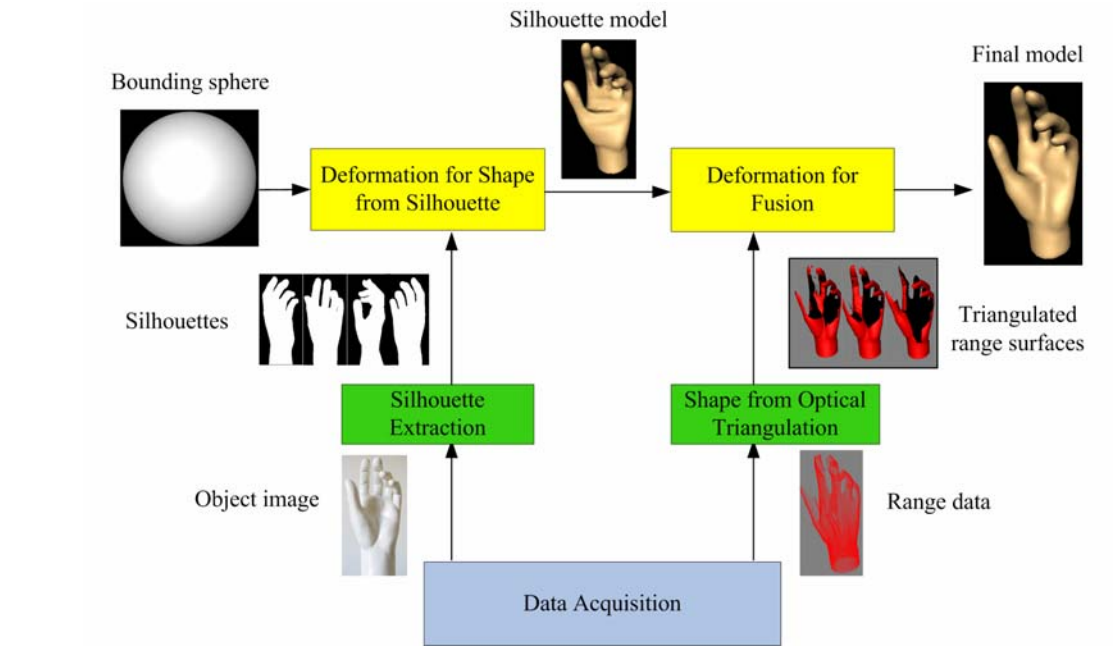


Figure 1.1: Block diagram of the overall reconstruction system.

The *data acquisition* task involves collecting silhouette and range data to be used for object reconstruction. We employ the same acquisition system as Yemez et al. [32], where the setup is composed of a computer, a camera, single-stripe laser projectors, and a turntable (see Fig. 1.2). The object to be reconstructed is placed on the turntable and the images of the object are taken at fixed intervals while the turntable rotates around a vertical axis. The laser projectors are pointed towards the object, and together with the camera, remain stationary throughout the acquisition. The different components that make up the acquisition system, the laser projectors, the camera, and the turntable are calibrated with respect to a common reference frame. After the calibration, the object to be reconstructed is placed on the turntable and image acquisition is carried out to obtain the silhouettes and the range data of each laser projector.

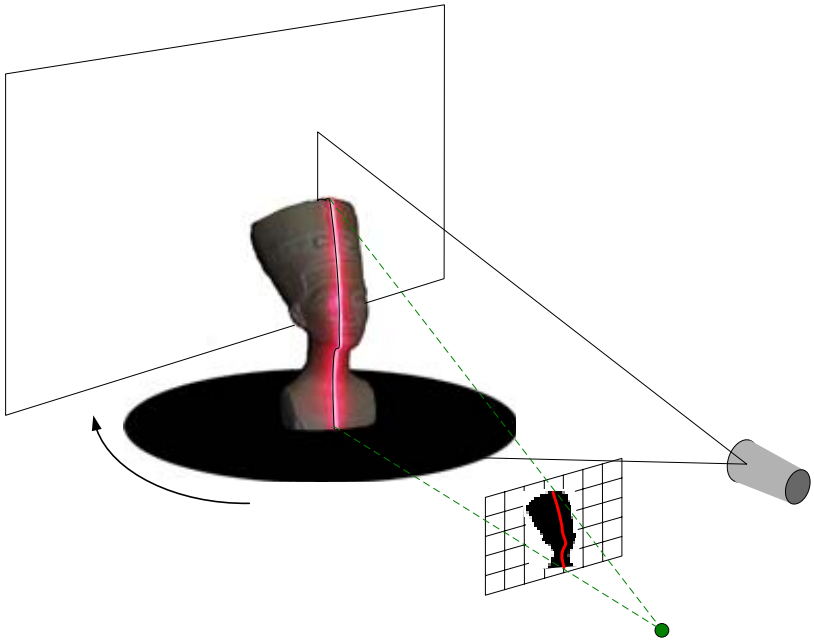


Figure 1.2: The data acquisition setup.

Accurate *silhouette extraction* is crucial for the quality of the reconstructed object. Our object extraction method depends on the use of a sharp contrast that must be maintained between the background and the object for precise results [2]. In this scheme the backstage is saturated with light while the object is left in the dark creating a natural silhouette of the object. To obtain a successful extraction, the strength of the light sources and the camera settings have to be fine-tuned. The background saturation method, although problematic with very shiny surfaces, in general produces very clean and accurate results and the radiosity and color confusion problems are circumvented (see Figure 1.3).

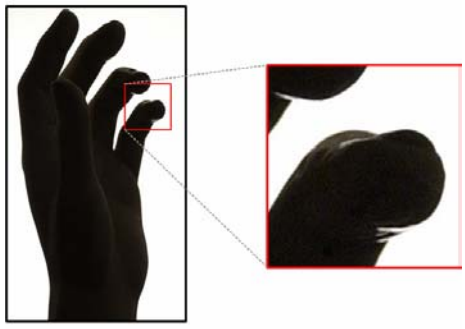


Figure 1.3: Image of an object with saturated background. The zoomed section reveals the difficulty encountered sometimes in discerning the object from the background.

The *deformation for shape from silhouette* is initiated with extraction of the silhouettes from the obtained input images. We take the Lagrangian approach for deformation and construct a mesh representation via surface evolution, starting from an initial model that represents the surface that bounds the object (the bounding sphere for instance). The initial model evolves towards the visual hull of the object surface under the guidance of the external forces driven by the silhouettes. The deformable model is refined or simplified where necessary during surface evolution using a set of local mesh transform operations and eventually adapts itself to local properties of the object surface.

The *shape from optical triangulation* first obtains a sequence of 2D points from the input laser images and then computes their 3D coordinates in the common reference frame using the calibration parameters. Finally, the cloud of 3D range points is triangulated into a mesh representing the object surface.

The *deformation for fusion* starts off with the surface mesh obtained by silhouette-based deformation, which correctly represents the visual hull, but lacks most of the concavities of the object surface. This surface mesh is further deformed by employing the

same deformation model and evolves towards the object surface under the guidance of the external forces which are driven by the triangulated range data.

The organization of the thesis is as follows. In Chapter 2 we describe our generic surface deformation framework that can be used to build surface models from any type of data. We explain our deformation-based technique for shape from silhouette and its fusion with shape from optical triangulation in Chapter 3 and Chapter 4, respectively. The experimental results regarding the shape from silhouette and the fusion are separately presented at the end of each relevant chapter. Finally in Chapter 5 we provide concluding remarks and discuss possible future work.

Chapter 2

SURFACE DEFORMATION

In this chapter we describe a surface deformation algorithm that starts off with an initial surface and evolves in a smooth manner towards the target surface boundary. We assume that the deformable model is represented in the form of a topologically correct triangle mesh that initially bounds the surface to recover. The algorithm is required to preserve the correct topology of the initial mesh during surface evolution. We denote the deformable surface with S and the target boundary to be recovered with B.

2.1 Lagrangian Deformation

Our deformation method is based on Lagrangian flow and seeks for an optimal surface S^* that minimizes a global energy term E:

$$E(S,B) = E_{\text{int}}(S) + E_{\text{ext}}(S,B) \tag{2.1}$$

where the internal energy component E_{int} controls the smoothness of the surface and the external energy component E_{ext} measures the match between the surface S and the object boundary B. This energy term can be minimized by solving the following first-order partial differential equation:

$$\frac{\partial S}{\partial t} = \mathbf{F}_{\text{int}}(S) + \mathbf{F}_{\text{ext}}(S,B) \tag{2.2}$$

where the internal and external forces, $\mathbf{F}_{\text{int}}$ and $\mathbf{F}_{\text{ext}}$, guide the initial surface in a smooth manner towards the object boundary. The discrete form of this differential equation can be solved by surface evolution via the following iteration:

$$S^k = S^{k-1} + \Delta t(\mathbf{F}_{\text{int}}(S^{k-1}) + \mathbf{F}_{\text{ext}}(S^{k-1}, B)) \tag{2.3}$$

The time step Δt, which is a scale factor, can simply be set to 1 when the strengths of the external and internal forces are adjusted accordingly.

By iterating the above equation, the surface S^k converges to its optimum S^* at the equilibrium condition when all the forces cancel out to 0. The external force component, $\mathbf{F}_{\text{ext}}$, is application-specific; its value depends on the current position of the surface with respect to the targeted boundary. The external force is commonly set to be in the direction of the surface normal:

$$\mathbf{F}_{\text{ext}}(P, B) = v(P, B) \cdot \mathbf{N}(P) \tag{2.4}$$

where $\mathbf{N}(P)$ is the normal vector and $v(P,B)$ is the force strength at vertex P of the deformable mesh. The force strength at each vertex P of the mesh and at each iteration of the surface evolution is based on how far and in which direction the vertex P is with respect to the targeted boundary. Thus the strength v may take negative values as well.

The internal force component, $\mathbf{F}_{\text{int}}$, controls the smoothness of the mesh as the surface evolves towards the object boundary under the guidance of the external force (see Fig. 2.1). At each vertex of the mesh, first the external force is applied as specified and then the internal force tries to regularize its effect by moving the vertex to the centroid of its neighbors, that is the Laplacian operator:

$$\mathbf{F}_{\text{int}}(P) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i - \mathbf{p} \tag{2.5}$$

where **p** stands for the vector form of the position of the vertex P and $\mathbf{p}_i$, $i = 0,1,\ldots, N$, are the vertices adjacent to **p**.

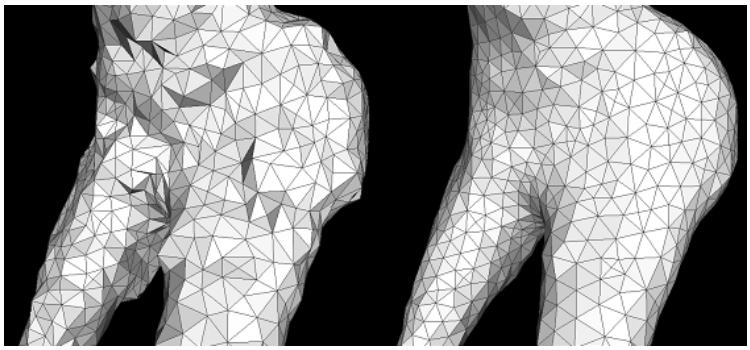


Figure 2.1: Effect of regularization by the internal force.

2.2 Local Mesh Transform Operations

If the deformation method is directly applied as described above without any further considerations, problems may arise during surface evolution, regarding the regularity and topology of the deformable model. These can be itemized as follows: 1) Non-manifold triangles may appear, 2) Degenerate edges may show up, 3) Irregular vertices with high valence values may occur, 4) The deformable mesh may fail to capture some fine details on the object surface.

Non-manifold triangles on the mesh structure may appear as the positions of the vertices are updated by the external forces. To prevent the occurrence of such topological problems, we constrain the strength of the external force with the finest detail on the mesh,

i.e., $|v_{max}| < \varepsilon_{min}/2$, where ε_{min} is the minimum edge length appearing on the mesh. To handle the three other problems, we incorporate three special procedures [35], [36], namely edge collapse, edge split and edge flip, to the surface deformation process (see Fig. 2.2).

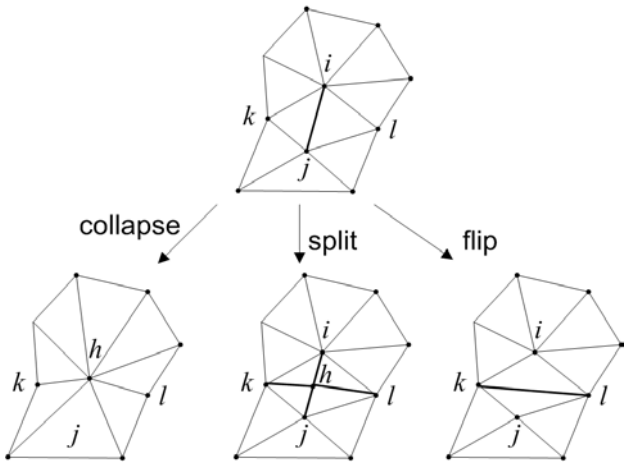


Figure 2.2: Edge collapse, split and flip operations.

- *Edge collapse*: As the vertices are pulled towards the object boundary by the external force, neighboring vertices may get too closer and cause degenerate edges. Thus we collapse an edge by merging their endpoints to the midpoint whenever its length falls below the threshold ε_{min}. The merging point can actually be optimized according to the needs of the application; it can be for instance one of the endpoints, whichever is appropriate, or the optimal position on the edge if it is possible to define one.

- *Edge split*: Similarly, as the deformable surface evolves, neighboring vertices may get further from each other and cause very long edges that degrade the regularity of the mesh. Moreover, parts of the deformable mesh where such long edges accumulate can not capture the details of the object shape. Thus whenever the length of an edge exceeds a certain threshold ε_{max}, an additional vertex is inserted on the middle position of such an edge and the data structure is updated accordingly. For the split operation to be compatible with the collapse operation, the threshold ε_{max} has to be chosen such that $\varepsilon_{max} \leq 2\ \varepsilon_{min}$ since otherwise the split operation would create edges with length larger than ε_{min}.
- *Edge flip*: Edge collapse and split operations inevitably change the valence distribution of the mesh structure, that yield irregular vertices. To prevent this, during surface evolution, the common edge of any two neighboring triangles is swapped with the one joining the unshared vertices of the triangles, as long as this operation favors the existence of the vertices of valence close to 6.

2.3 Topological Problems

The local mesh operations themselves may cause topological problems such as fold-overs and non-manifold triangulations if no particular care is taken. These problems and how we deal with them are explained in the sequel:

1) The edge collapse operation may occasionally cause a mesh triangle to fold over another and may create a non-manifold triangulation created as depicted in Fig. 2.3. If the collapse of an edge is detected to be an illegal move in this sense as defined in [36], the collapse operation is avoided for that edge. Illegal collapses are rarely encountered and do not significantly degrade the mesh regularity.

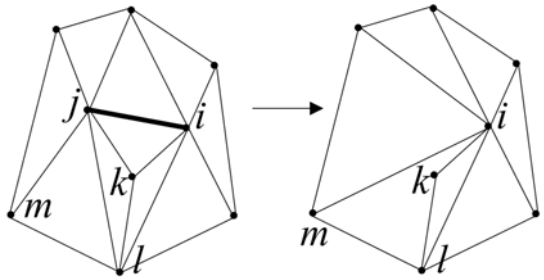


Figure 2.3: Illegal edge collapse: The collapse of the edge *ij* onto the vertex *i* causes the triangle *ikl* to fold over the triangle *iml*, which results in a non-manifold triangulation.

2) Split operations are always legal and never result in topological problems. However, there is an infinite loop problem regarding edge split operation, which can be avoided by splitting the edges in an appropriate order: All the edges of the mesh, that need to be split, are first arranged in descending order with respect to their lengths and then split in that order (see Fig. 2.4).

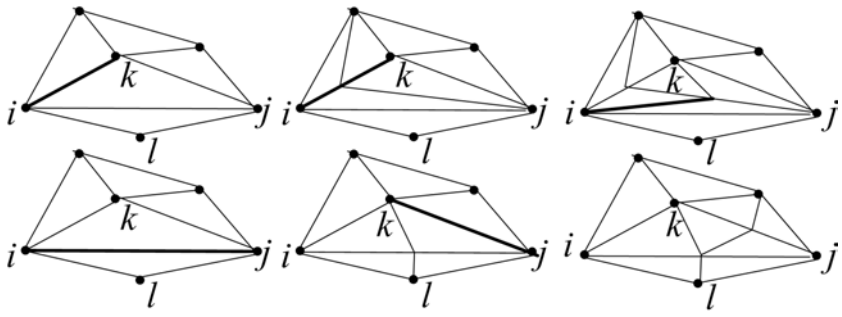


Figure 2.4: Infinite loop problem. (Top row) Splitting the edge *ik* before the longer edge *ij* causes an infinite loop. (Bottom row) Splitting first the longer edges avoids this problem and yields a more plausible triangulation.

3) An edge flip is legal if and only if the edge is adjacent to two triangles whose union is a convex quadrilateral. Figure 2.5 illustrates an illegal edge split operation that results in a non-manifold triangulation.

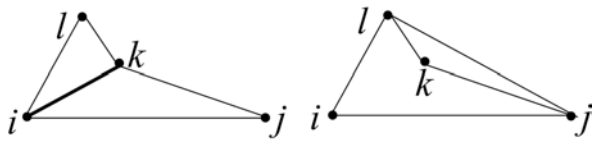


Figure 2.5: Illegal edge flip: Flipping the edge *ik* with *lj* creates a non-manifold triangulation.

2.4 Surface Evolution

The deformation algorithm incorporates the local mesh transform operations to the surface evolution process in an adequate order and where necessary, and continues to iterate until convergence. The overall surface evolution algorithm in its most generic form is as follows (see also Fig. 2.6):

Iterate

- Move each vertex P with $v(P, B)$ in the direction of its normal $\mathbf{N}(P)$,
- Regularize the mesh using Equation 2.5,
- Collapse edges with length smaller than ε_{min},
- Split edges with length exceeding ε_{max},
- Flip edges where necessary, favoring the vertices with valence close to 6,

Till convergence

There are various alternatives to set the convergence criterion. Ideally, the algorithm converges when the vertices of the deformable mesh no longer move, that is, at the equilibrium state where the external and internal forces all cancel out to zero. In practice, the algorithm can be terminated if the number of iterations exceeds a given threshold or when the error between the object boundary and the deformable mesh falls below a certain value. Another possibility, that gives a significant computational efficiency, is to freeze and inactive those vertices of the deformable mesh that are detected to get already attached to the boundary.

The final model at convergence can capture the details that can be represented with the resolution imposed by the minimum edge length ε_{min}. By adjusting this parameter, the boundary surface of an object can be represented at various resolutions or at different levels of detail as desired.

Thanks to local mesh transform operations incorporated to the surface deformation process, the deformable mesh always remains topologically correct and preserves an optimal configuration for connectivity. One limitation of the presented algorithm is that the initial topology of the deformable mesh can not be altered, however this limitation can be overcome by employing special procedures to detect possible splitting and merging as proposed in [31].

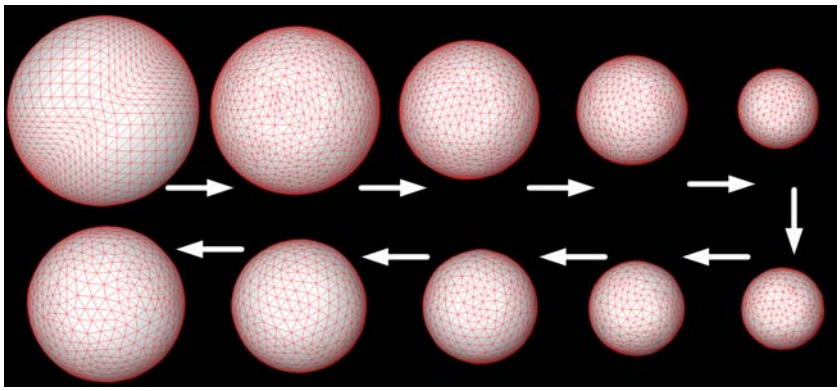


Figure 2.6: Illustration of the surface deformation algorithm. (Top row) 5 out of 17 iterations by applying a constant negative external force $-\varepsilon_{min}/2$ to the initial sphere model. (Bottom row) Same as the first row, but this time a constant force in the reverse direction is applied.

Chapter 3

SHAPE FROM SILHOUETTE

The shape from silhouette technique has recently regained interest since its robust output forms a solid initial model for further tasks of computer vision. Silhouette models can not capture hidden surface concavities, but can further be carved or deformed so as to achieve a more accurate object representation by incorporating stereo or optical triangulation information [21], [32]. Also, the shape from silhouette technique, as a passive reconstruction method, can successfully be used to model time-varying scenes and objects [19], [24],[28].

The shape from silhouette techniques predict the 3D shape of an object using silhouettes, which are projections of the object to 2D image planes. Using the calibration parameters for an image plane, it is possible to back-project an object contour to obtain a 3D silhouette cone that bounds the object volume. The intersection of the silhouette cones generated in this way by a set of image planes from different views provides an approximation to the 3D *visual hull* [37], a superset of the object shape. The common strategy to compute this intersection is first to construct an intermediate volumetric representation and then to triangulate it via marching cubes algorithm [25] so as to obtain a surface model. The marching cubes algorithm however does not always produce topologically correct triangulations. Mesh models that contain non-manifold triangles pose problems when manipulated, and thus can not be used for further deformation.

The surface deformation framework that we have presented in Chapter 2, when applied to shape from silhouette problem, produces topologically correct mesh representations that are eligible for further deformation. These silhouette-based models can easily be manipulated for instance to incorporate stereo or optical triangulation information or to track time-varying geometry.

3.1. Silhouette-based Deformation

We reconstruct the mesh representation of the object surface from its multi-view silhouettes by deforming the 3D bounding sphere that encloses the shape. The bounding sphere is represented as a mesh with topology of genus 0.

The external force component, $\mathbf{F}_{\text{ext}}$, is based on silhouette information, though it is also possible to incorporate texture or color consistency information [21]. We set the direction of the external force so as to be perpendicular to the deformable surface as before. Recalling Eq. 2.4, the strength of the external force $v(P)$ at each vertex P of the mesh and at each iteration of the surface evolution is based on how far and in which direction (inside or outside) the vertex P is with respect to the silhouettes. Thus the force strength v, which may take negative values as well, is computed by projecting P onto the image planes and thereby estimating an isolevel value $f(P)$ via bilinear interpolation:

$$v(P) = \varepsilon_{\min} f(P) = \varepsilon_{\min} \min_n \left\{ G\left[\text{Proj}_{I_n}(P) \right] - 0.5 \right\}, \tag{3.1}$$

where $\text{Proj}_{I_n}(P)$ is the projection of the point $P(x, y, z)$ to I_n, the n'th binary image (0 for outside, 1 for inside) in the sequence, and

$$\begin{aligned} G(x', y') = (1-\alpha)((1-\beta) I(\lfloor x' \rfloor, \lfloor y' \rfloor) + \beta I(\lfloor x' \rfloor, \lfloor y' \rfloor + 1)) \\ + \alpha((1-\beta) I(\lfloor x' \rfloor + 1, \lfloor y' \rfloor) + \beta I(\lfloor x' \rfloor + 1, \lfloor y' \rfloor + 1)) \end{aligned} \tag{3.2}$$

where $(\lfloor x' \rfloor, \lfloor y' \rfloor)$ denotes the integer part and (α, β) is the fractional part of the coordinate (x', y') in the binary discrete image *I*. The function *G*, taking values between 0 and 1, is the bilinear interpolation of the sub-pixelic projection (x', y') of the vertex *P*. Thus, the isolevel function $f(P)$ takes on values between -0.5 and 0.5**,** and the zero crossing of this function reveals the isosurface. The isovalue of the vertex *P* is provided by the image of the silhouette that is farthest away from the point, or in other words, where the interpolation function *G* assumes its minimum value.

We distinguish the vertices of the deformable mesh under five categories with respect to their isovalues: VERY-IN, IN, ON, OUT and VERY-OUT. Let *L*(*P*) denote the state of a vertex *P* with isovalue *f*(*P*) at a given iteration. We define *L*(*P*) as follows:

$$L(P) = \begin{cases} \text{VERY-OUT} & \text{if } f(P) = -0.5 \\ \text{OUT} & \text{if } -0.5 < f(P) < 0 \\ \text{ON} & \text{if } f(P) = 0 \\ \text{IN} & \text{if } 0 < f(P) < 0.5 \\ \text{VERY-IN} & \text{if } f(P) = 0.5 \end{cases} \tag{3.3}$$

According to this definition, OUT and IN vertices are those positioned within a narrow band around the boundary surface. This band has a thickness of approximately two pixels when projected onto the image planes. By Eq. 3.1, the external force strength at each OUT and IN vertex varies within the interval $(-\varepsilon_{\min}/2, 0)$ or $(\varepsilon_{\min}/2, 0)$, respectively. The vertices which are out of this band are labeled as VERY-IN and VERY-OUT depending on whether they are located inside or outside the silhouettes and have fixed force strength $\varepsilon_{\min}/2$ or $-\varepsilon_{\min}/2$. ON vertices are assumed to be located exactly on the shape boundary and do not move. Figure 3.1 illustrates categorization of vertices when projected onto a single silhouette image.

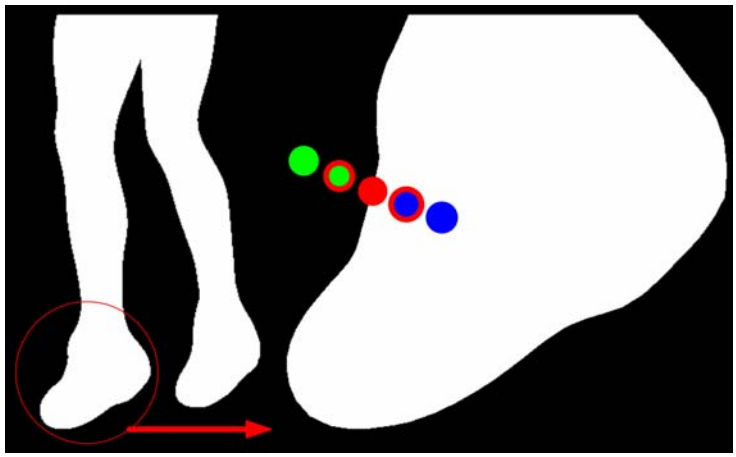


Figure 3.1. Vertex categorization on a single silhouette. The red point is the projection of an ON vertex that falls exactly on the silhouette boundary. The green and green-red points correspond to VERY-OUT and OUT vertices according to their isovalue. Similarly, the blue and blue-red points correspond to VERY- IN and IN vertices, respectively.

3.2. Fine Tuning

When the surface deformation algorithm presented in Chapter 2 is iterated starting with the initial bounding sphere and using the external force given by Eq. 3.1, the vertices of the deformable mesh, which are all initially VERY-OUT, are gradually pulled towards the object boundary. The external force strength at a vertex remains fixed at its maximum default value, $\varepsilon_{\min}/2$, as long as the vertex remains VERY-OUT. When the vertex enters into the boundary band, its speed decreases in proportion to its distance to the boundary.

During surface evolution, the state of a vertex can switch between any two of all five categories. A vertex moves not only due to the external force, but also due to the regularization effect of the internal force that alters its positioning. Depending on the

magnitude of the external force, which is bounded above by $\varepsilon_{\min}/2$, the state of a vertex can even switch from VERY-OUT to VERY-IN, or vice versa, at one single iteration.

The vertices of the deformable mesh, when they get close to the boundary, usually oscillate between IN and OUT states until convergence, that is, until they no longer move. Some vertices remain as VERY-OUT or VERY-IN even at convergence. To improve accuracy and to speed up convergence, we incorporate a fine-tuning procedure to the surface deformation process. We detect the instance that a vertex changes its state for the first time from outside to inside, or vice versa, and then locate the point where it crosses the boundary. The vertex is frozen at that point and does not move any more (see Fig. 3.2).

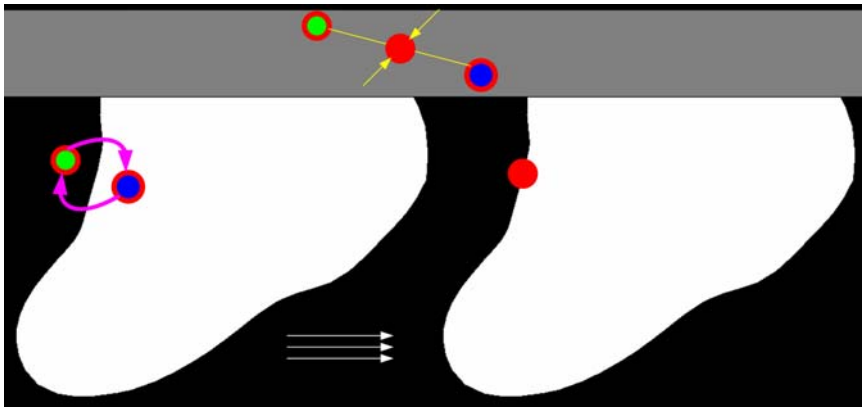


Figure 3.2. Fine-tuning. The exact location of a vertex, that changes its state from outside to inside, or vice versa, is sought on the silhouette boundary via binary subdivision.

Suppose that, at iteration k, the state of a vertex P changes from VERY-IN or IN to VERY-OUT or OUT. Let P^{k-1} and P^k denote the positions of the vertex at iterations k-1 and k, respectively. To locate the point where the vertex is to be frozen, a dichotomic subdivision is carried out to search for the point where the isolevel function f is close to 0 on the line segment joining P^{k-1} and P^k. A sufficiently small threshold value ξ, where

$-\xi < f(P) < \xi$, is used to determine how close the isolevel of a given point must be to assume it as being exactly on the boundary surface (see Figure 3.3).

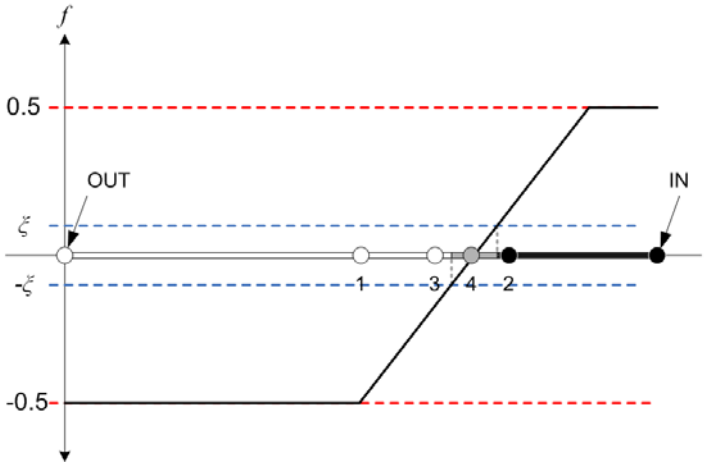


Figure 3.3. Binary subdivision procedure to accurately locate the position of a vertex. One endpoint of the line segment is OUT (assigned a negative isovalue) and the other is IN (assigned a positive isovalue). The boundary surface is assumed to pass where the isolevel function f takes on a value between ξ and - ξ. For this example, the binary search takes 4 steps to locate the boundary point.

3.3. Algorithm

The shape from silhouette algorithm differs in two ways from the deformation algorithm that we have presented in Chapter 2. First, a fine-tuning procedure is incorporated to the algorithm. Second, external and internal forces are applied only to active vertices, that is, to those vertices that are not yet fine-tuned and frozen. These two modifications improve accuracy and speed up the convergence. The algorithm continues to iterate until all vertices get attached to the boundary and become inactive:

Iterate

- Move each active vertex P in the direction of its normal $\mathbf{N}(P)$, using Eq. 3.1
- Regularize the active vertices of the mesh using Equation 2.5,
- Fine-tune vertices that move from outside to inside, or vice versa,
- Collapse active edges with length smaller than ε_{min},
- Split active edges with length exceeding ε_{max},
- Flip active edges where necessary, favoring the vertices with valence close to 6,
- Inactivate vertices that have been fine-tuned

Until all vertices are inactivated

Note that only active edges of the deformable mesh, that is, edges with at least one active vertex, may need to be collapsed, split or flipped. Thus as iterations proceed and as more and more vertices become inactive, the time spent at each iteration significantly reduces, yielding on overall a computationally efficient algorithm.

3.4. Refinement

The choice of the minimum edge length, $\varepsilon_{\min}$, is critical. Ideally it should match the resolution of the silhouette images, that is, the finest detail that can be captured. The smaller the value of $\varepsilon_{\min}$, the higher is the resolution of the deformable mesh. The choice of $\varepsilon_{\min}$ also scales and puts an upper bound to the magnitude of the external force. Thus small values of $\varepsilon_{\min}$ slow down the algorithm. To speed up the algorithm and to obtain representations at lower resolutions, one may choose to set $\varepsilon_{\min}$ to relatively large values, which may however pose problems in recovering the shape of objects with deep and

narrow, though visible, concavities. For such objects, the deformable mesh may fail to proceed into concavities, resulting in an unsatisfactory representation. This is depicted in Fig. 3.4, where two vertices of an edge, which are located exactly on the boundary, become inactive, though the mesh needs to proceed further to be able to represent the shape satisfactorily.

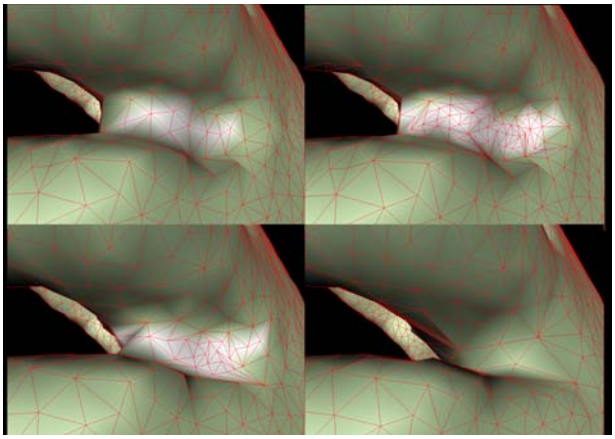


Figure 3.4. Refinement with further slit operation: The endpoints of some edges between two fingers are frozen (ON) although their midpoints are still VERY-OUT; a situation which triggers further split operations.

Once the algorithm converges to its optimal surface for a given $\varepsilon_{\min}$, our deformation scheme avails us with the possibility of further increasing the resolution at parts of the deformable mesh that fail to represent the shape adequately. After convergence, each edge having a VERY-IN or VERY-OUT midpoint is split at that point. The deformation algorithm is rerun, activating only the newly inserted vertices and keeping the same value for $\varepsilon_{\min}$. This process can be repeated until no vertices with VERY-IN or VERY-OUT midpoints remain on the deformable mesh.

3.5. Experimental Results

We have tested our silhouette-based deformation technique on both synthetic and real-world objects. The original images of these objects, which are Hand (made of plastic), Elephant (made of wood) and the synthetic Jumping Man are displayed in Fig. 3.5.

Figure 3.5: Original images of Elephant, Jumping Man, and Hand objects.

The resolution of the images acquired during the experiments is 2000 by 1310 pixels which is sufficient to support the high level of detail demanded by the reconstructions. The number of silhouettes used for each reconstruction is 72.

Figures 3.6 - 3.18 illustrate the steps/results of silhouette-based reconstructions for the three objects. For all intermediate step demonstrations, green, white, blue, magenta and yellow colors represent ON, Very OUT, Very IN, OUT and IN vertices, respectively. The initial mesh is chosen to be the bounding sphere for all cases, which is automatically determined from silhouette images.

Figures 3.6-3.10 illustrate the results of the Elephant reconstruction at two different resolutions, obtained by properly adjusting the value of ε_{min}. In Fig. 3.6 we provide views from the deformable model at various iterations of the surface deformation process, as the initial bounding sphere evolves towards the object boundary. Fig. 3.7 displays views from various angles of the final Elephant model obtained at low resolution. Figures 3.8 and 3.9 provide intermediate and final results for the Elephant reconstruction at a higher resolution. The final models obtained at low and high resolutions are compared in Fig. 3.10. As we observe from figures, the final silhouette models lack cavity information, especially on the ear and trunk of the Elephant.

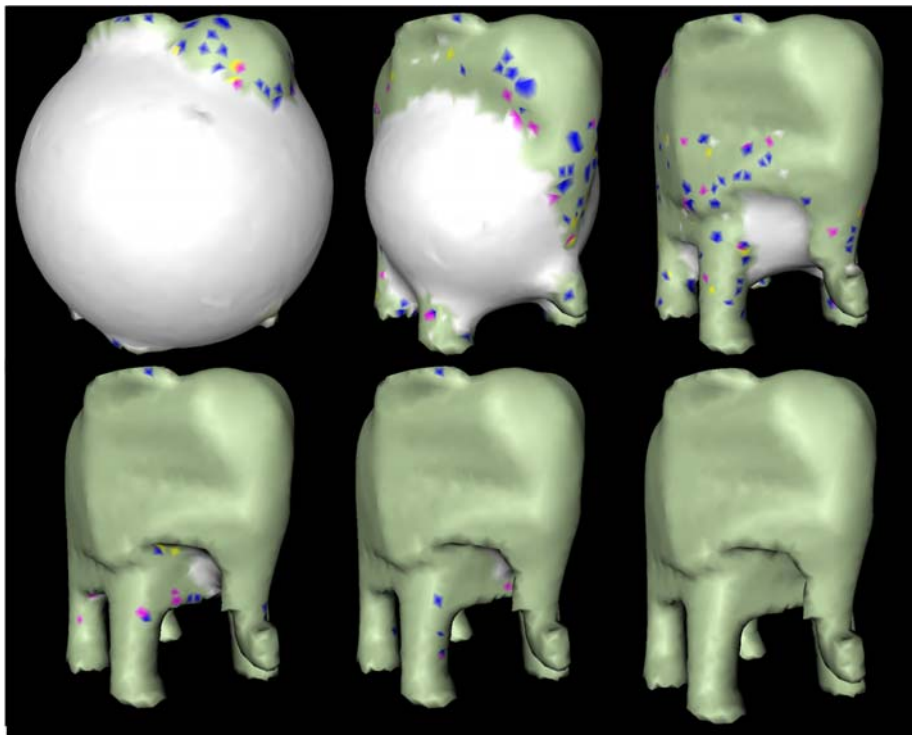


Figure 3.6: Reconstruction of the Elephant object at low resolution. Views from various iterations of the deformation process, as the initial bounding sphere evolves towards the object boundary.

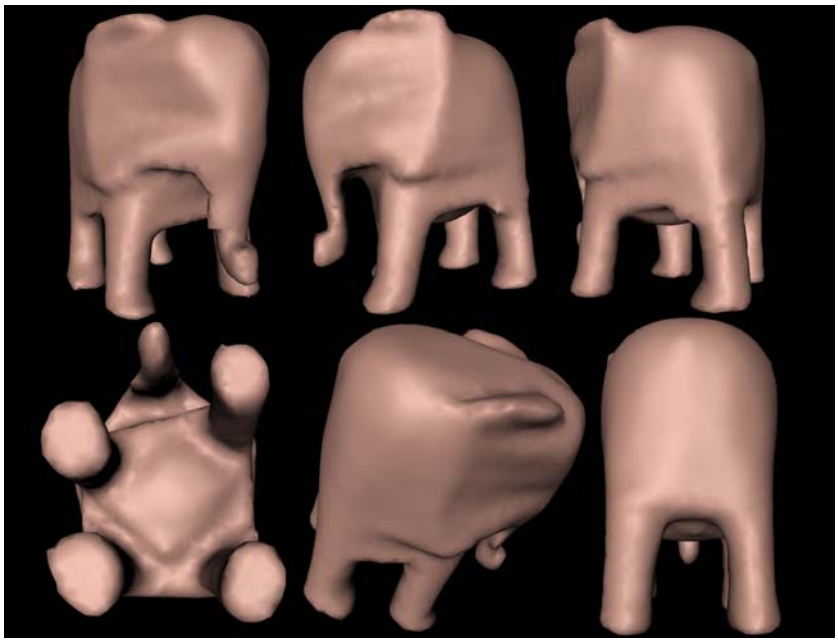


Figure 3.7: Various views from the final Elephant reconstruction at low resolution after two steps of fairing.

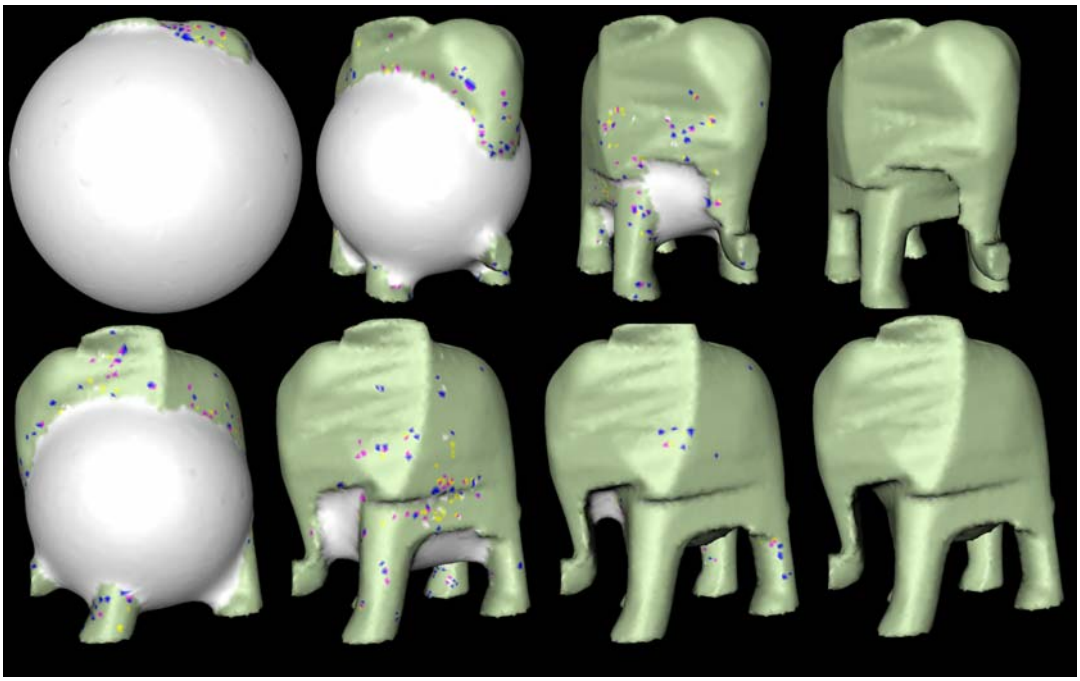


Figure 3.8: Reconstruction of the Elephant object at high resolution. Views from various iterations of the deformation process.

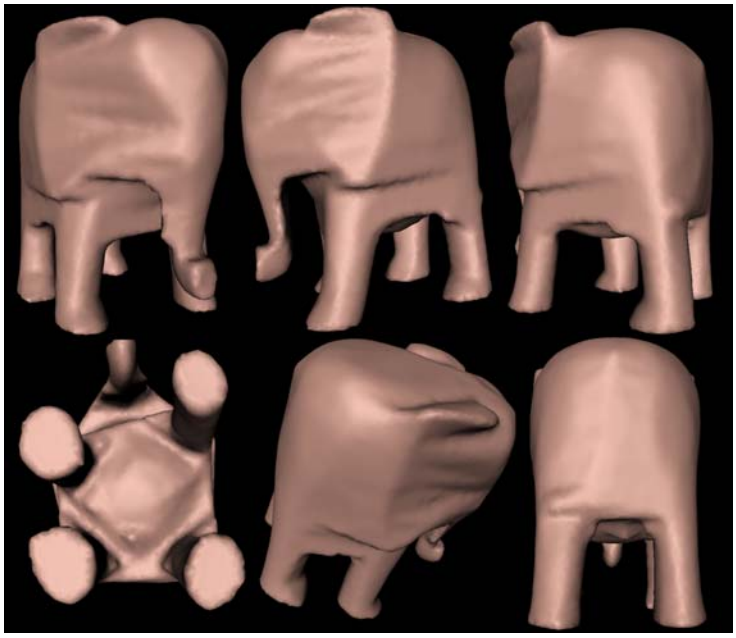


Figure 3.9: Various views from the final Elephant reconstruction at high resolution after two steps of fairing.

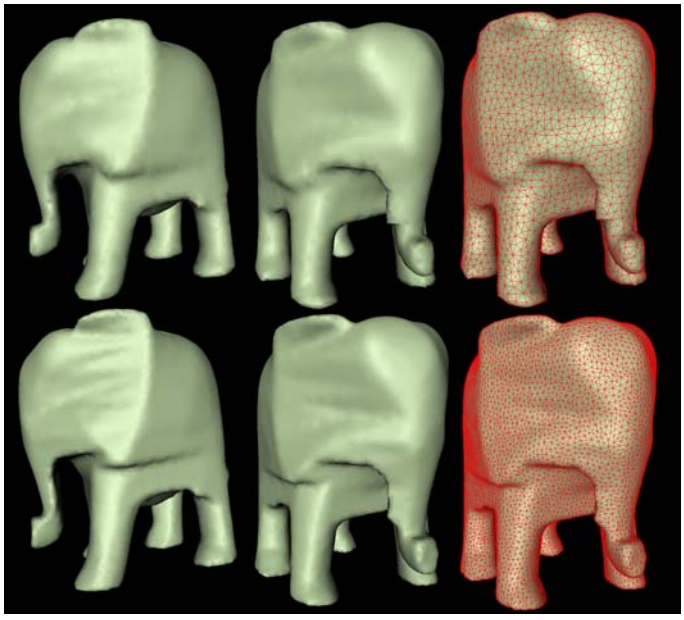


Figure 3.10: (Top vs. bottom) Low vs. high resolution final reconstructions.

Figures 3.11 to 3.13 illustrate the results of the synthetic Jumping Man reconstruction. In Figures 3.10 and 3.12, we provide views from the deformable model at various iterations of the surface deformation process, as observed from two different angles, respectively. Fig. 3.13 displays views from various angles of the final model. We observe that the Jumping Man does not contain severe hidden concavities and the final reconstructions are smooth and very satisfactory, that can adequately represent the large visible cavities, such as those between the legs of the Jumping Man.

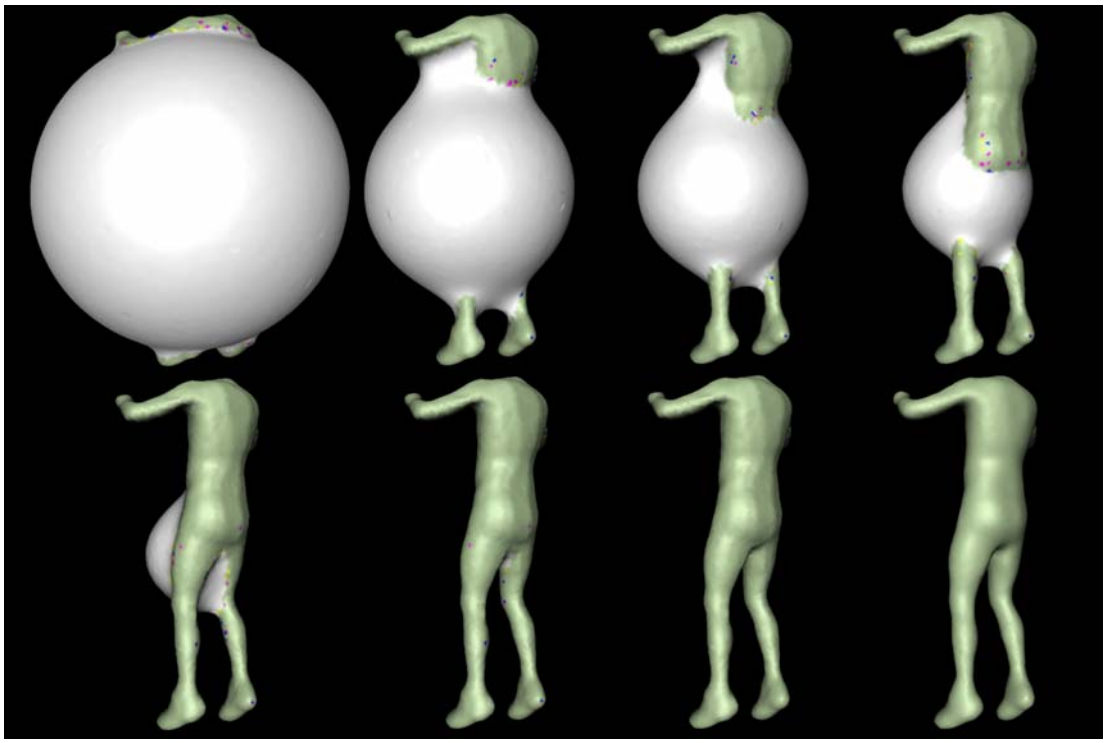


Figure 3.11: Reconstruction of the Jumping Man. Views from various iterations of the deformation process. The last (right bottom) image is the two-step faired version of its left adjacent.

Figure 3.12: The same deformation iterations of Fig. 3.11, as observed from a different angle.

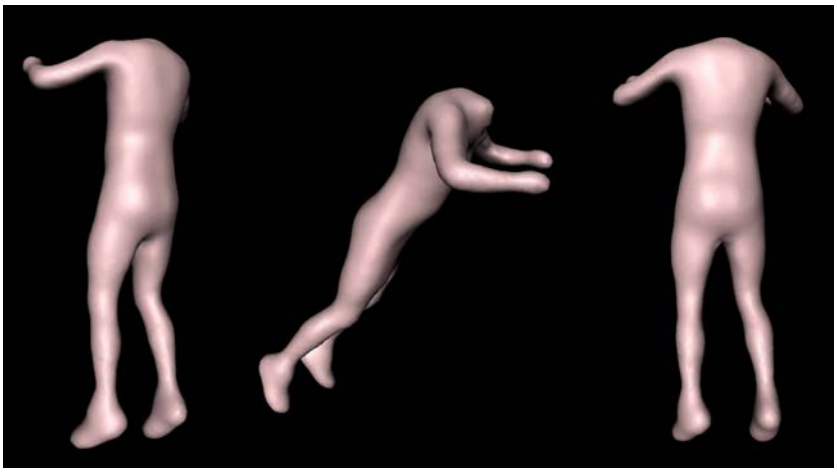


Figure 3.13: Various views from the final Jumping Man reconstruction after two steps of fairing.

Figures 3.14 to 3.17 illustrate the results of the Hand object reconstruction. Figure 3.14 displays views from the deformable model at various iterations of the surface deformation process. The chosen resolution causes the evolving mesh to get stuck as we carve the gap between the bird and ring finger (and also the ring and middle finger). As described in Section 3.4, the mesh model obtained after convergence is further refined using the edge split operation and the deformation algorithm is rerun to obtain more accurate results. As observed in Figures 3.15 and 3.16, at the end of the second run of the algorithm, the fingers are successfully recovered. The cavity of the palm is however missing in the final reconstruction (Fig. 3.17).

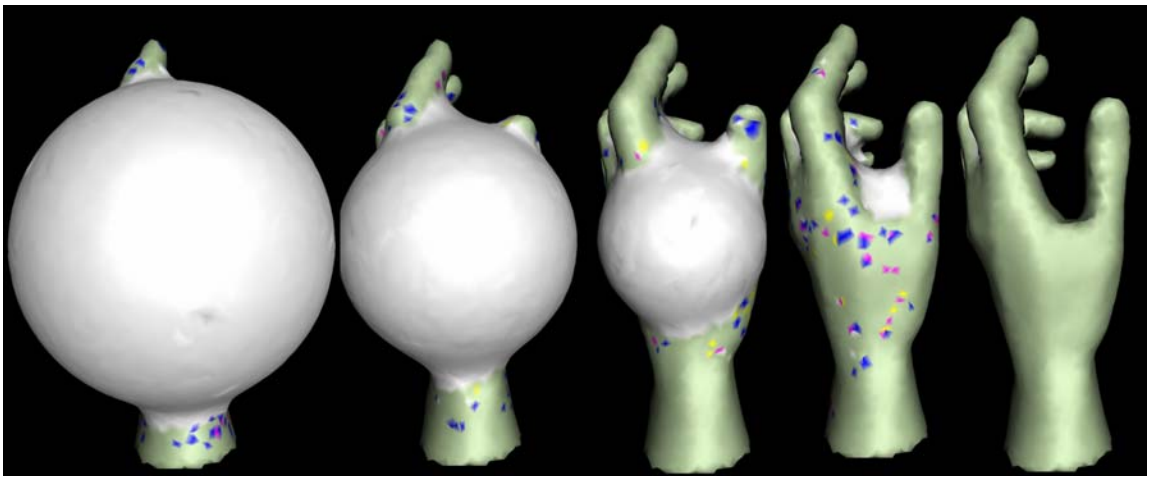


Figure 3.14: Reconstruction of the Hand object. Views from various iterations of the deformation process.

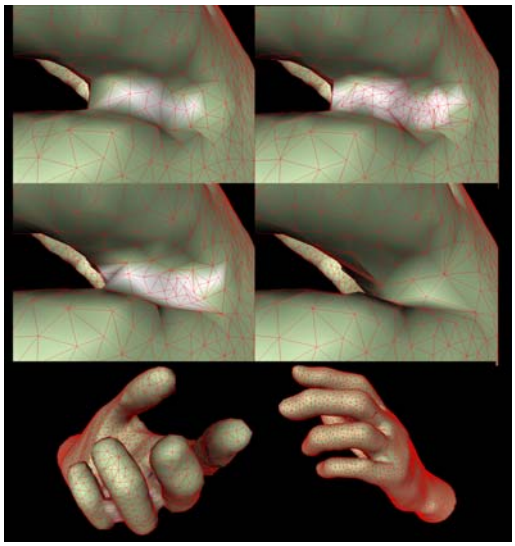


Figure 3.15: Refinement of the Hand model: (First two rows) The deformable mesh can not penetrate into the thin gap between the bird and the ring fingers due to insufficient resolution. The refinement procedure with further split operations increases the mesh resolution in these regions, which eventually leads to perfect fingers. (Last row) Zoomed-out versions of the first (top-left) and the last (bottom-right) images displayed in the first two rows.

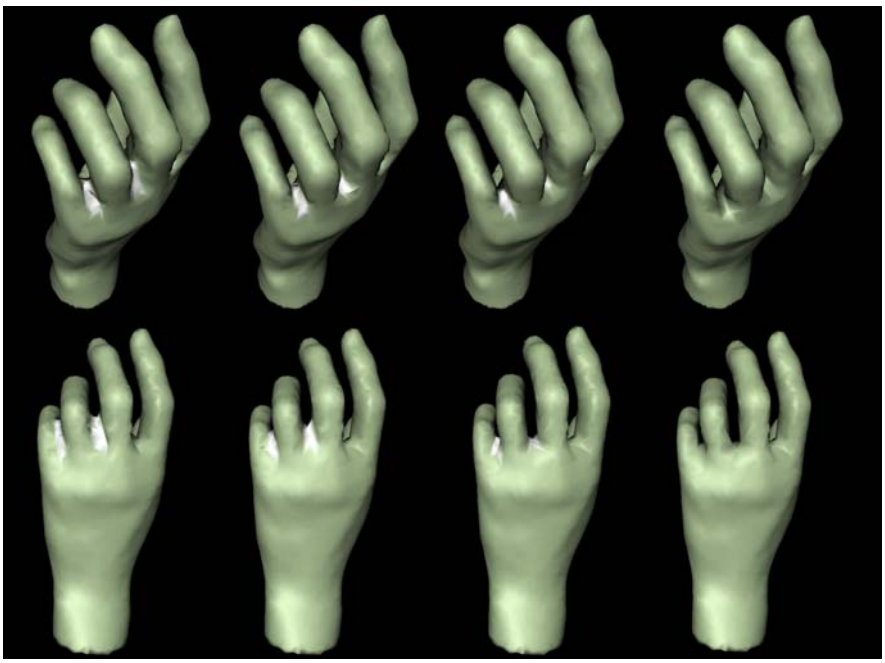


Figure 3.16: Views from various iterations of the deformation process after the refinement procedure. Note that the fingers are correctly recovered.

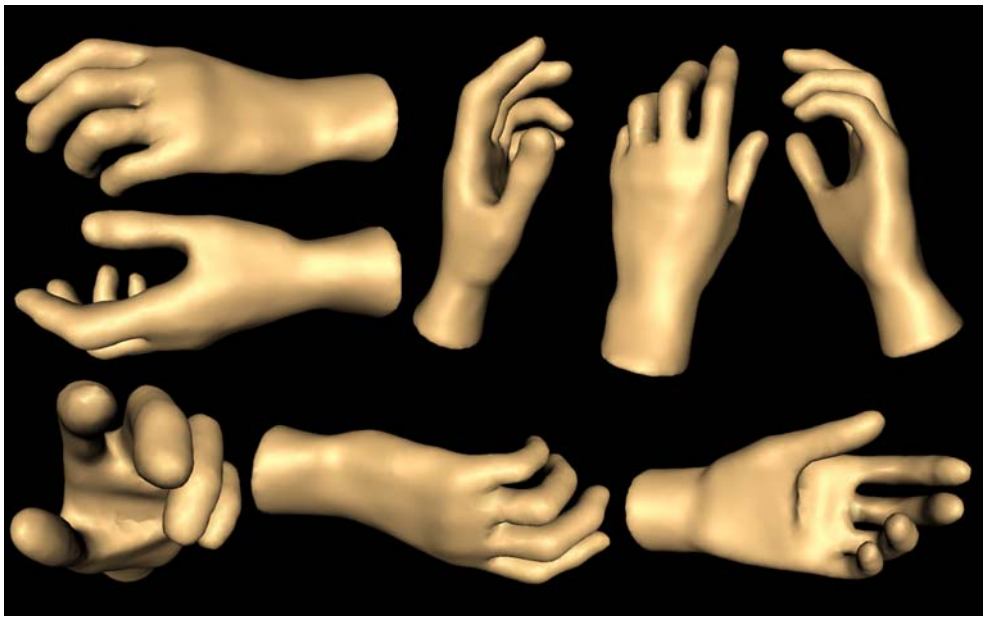


Figure 3.17: Various views from the final Hand reconstruction after two steps of fairing.

Finally, Figure 3.18 visualizes the quality of the meshes produced whereas Table 3.1 reveals the structural mesh information and the compact execution times of the algorithm measured on an Intel Pentium-4M 3GHz PC with 512MB of RAM.

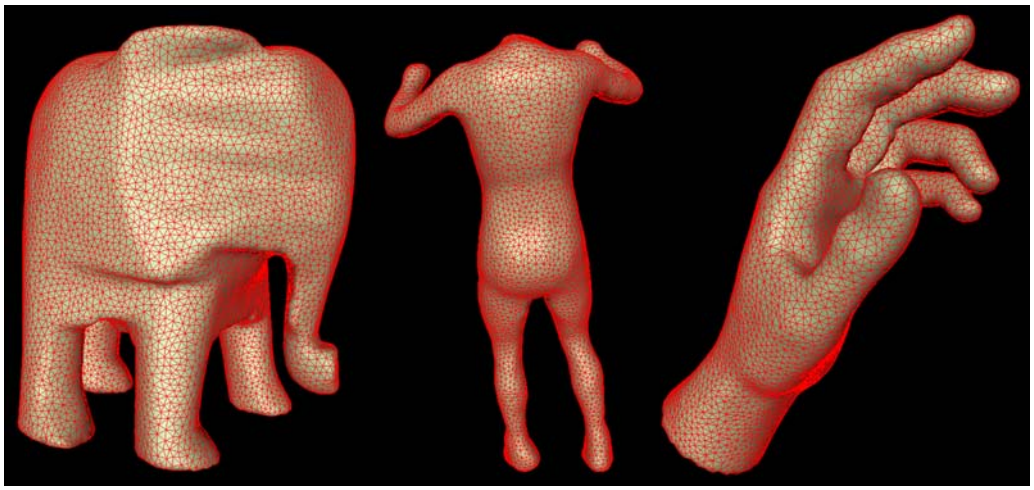


Figure 3.18: Final reconstructions in wireframe-overlay.

Table 3.1: Structural mesh information and execution times.

Object	**Vertices (#)**	**Faces (#)**	**Resolution**	$\varepsilon_{\min}$	**Time (secs)**
Elephant	4246	8488	Low	5/204 = .025	54
Elephant	11761	23518	High	3/204 = .015	198
Jumping Man	6073	12142	High	14/1850 = .007	225
Hand	5680	11356	High	2.3/176 = .013	228

The resolution of reconstruction is determined by the chosen value $\varepsilon_{\min}$, which is computed as the minimum edge length in proportion the radius of the bounding sphere. The value of $\varepsilon_{\min}$ also gives a clue about the corresponding execution time; the larger the value of $\varepsilon_{\min}$, the larger the speed of each vertex during deformation and the fewer the number of faces to deal with.

Chapter 4

FUSION - SHAPE FROM SILHOUETTE AND OPTICAL TRIANGULATION

In this chapter we describe an algorithm that combines the shape from silhouette and shape from optical triangulation techniques. The algorithm is based upon the surface deformation framework presented in Chapter 2. A silhouette-based mesh representation is first obtained as described in Chapter 3. This initial model is then further deformed under the guidance of the external forces exerted by the range data obtained from optical triangulation. The aim is to achieve a cavity-sensitive, hole-free surface model of the object by fusing the geometry information obtained from silhouettes and range data in the smoothest and the most accurate possible manner. We first discuss the general shape from optical triangulation problem and describe our 3D scanning system. Then we address the fusion problem and provide experimental results.

4.1. Shape from Optical Triangulation

Shape from optical triangulation techniques employ range scanners, whose basic components are sensors and light projectors. In order to generate a more complete 3D model of the scanned objects, most range scanning systems also integrate instruments such as turntables to move and cover the object from multiple views. In general, optical triangulation scanners produce range images which are a collection of points with regular

spacing. This is achieved by first casting a pattern of light (usually a planar stripe) onto the object of interest, while the sensor (the camera) observes the reflected light.

The structure of the scanner setup directly affects the acquisition process and the quality of the optical triangulation reconstruction. Therefore, it is important to distinguish the different types of available scanners [38]. The simplest and most common of all the structured light systems is the single-camera, single-stripe optical triangulation system. Since a single stripe covers only a small portion of the object, the scanner must be swept along the object to obtain a full range image. This is most commonly implemented with a circular sweep using a turntable on which the object is placed. Some systems also incorporate translational movement either to the scanner or to the object. A single sweep produces parallel stripes of sampled range points from which a patch of the object surface is triangulated. This system introduces the added computation of translational calibration and the problem of aligning overlapping parts of the multiple patches obtained from different views. Another alternative is the use of multi-stripe projectors that bear the requirement of distinguishing the stripes one from another [11].

The current setup that we employ fits into the single-camera, single-stripe scanner category, supporting only a rotational movement of the object (see Fig. 1.2). The alignment of patches is not required, but in turn the surface coverage is limited. However it is possible to make multiple scans with the projectors positioned differently, to cover more of the object surface. Since the object is rotated with respect to the projector, the projected laser planes intersect in space, yielding irregular sampling of the surface. This may cause deficient range data on some parts of the surface for some objects, as depicted in Fig. 4.1.

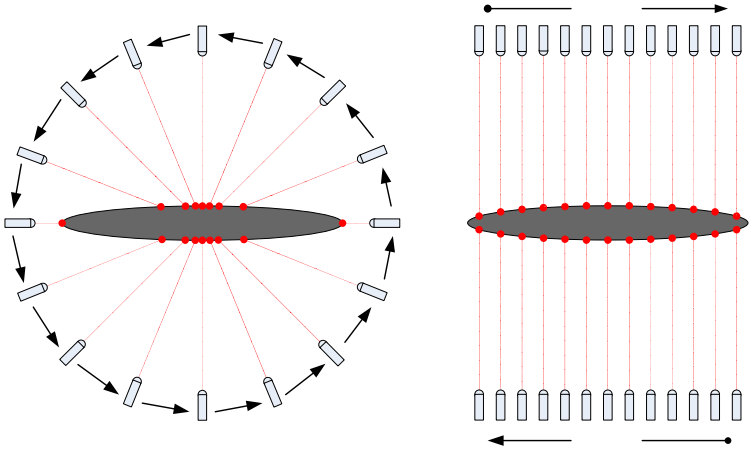


Figure 4.1: Top view of a slice from an ellipsoidal object with two separate scanner configurations. (Left) A rotational single-stripe projector takes samples from the object surface. The samples are denser towards the centre of the object. This irregularity in the distribution causes some parts of the object to be under-sampled for the extraction of proper surface information. (Right) A translational projector is swept twice over the object. The obtained samples are more uniformly spread out which results in a smoother and more complete surface reconstruction than that of the rotational scheme.

The input to our shape from optical triangulation process is a series of laser images of the object in full rotation on the turntable. The routine starts with processing the laser images to draw 2D sample points from the laser stripes. A depth profile is computed from the sample points, creating a cloud of range points. Connecting nearest neighbors with triangles is a common strategy for systems that produce lattices of regular samples, such as in [10]. The triangulation method that we use undertakes a similar path by weaving a web of faces across range points of adjacent stripes.

The fusion technique that will be presented in Section 4.2 does not in fact explicitly use the connectivity information of a polygon mesh generated from the range points. The triangulation of range points rather serve us to estimate a surface normal for each range point. Note also that the oriented range points resulting from different scans will be

integrated separately in the fusion process, thus they are not yet merged into a single surface representation.

4.2. Fusion

The most prominent problem with range scanning devices is that range points can only be collected from the observed portions of the object surface. The sensor may not be able to access the obstructed sections of the object (camera occlusion) or there may be parts of the object surface that cut off the projected light and prevent other parts from getting illuminated (laser occlusion), which eventually, cause the surface reconstruction to be incomplete. Although some scanners perform better than others in generating more complete surfaces, final surface reconstructions for some objects will always contain holes, no matter what kind of scanning is implemented or how many scans are run. Depending on the material of the object, the projected light may get scattered or reflected from off the object surface, which may increase further the proliferation of holes. Use of hole filling algorithms and multiple laser sources and cameras help improve the reconstructions but some portions of the object surface such as the inner walls of hollow parts are at best unreliably reconstructed. This main flaw in range scanning systems is our motivation to fuse optical triangulation with shape from silhouette.

Figure 4.2 demonstrates our motivation and the basic idea of our fusion technique on an example. Neither the shape from optical triangulation nor the shape from silhouette technique alone can satisfactorily recover the shape of the Hand object displayed in Fig. 4.2a. The silhouette model (Fig. 4.2b) lacks concavities of the palm, but the fingers are successfully recovered. On the other hand, the optical triangulation model (Fig. 4.2c) exhibits holes due to severe occlusions. The fingers not only occlude each other, but also obstruct the palm on which the cavity information can only partially be recovered. The

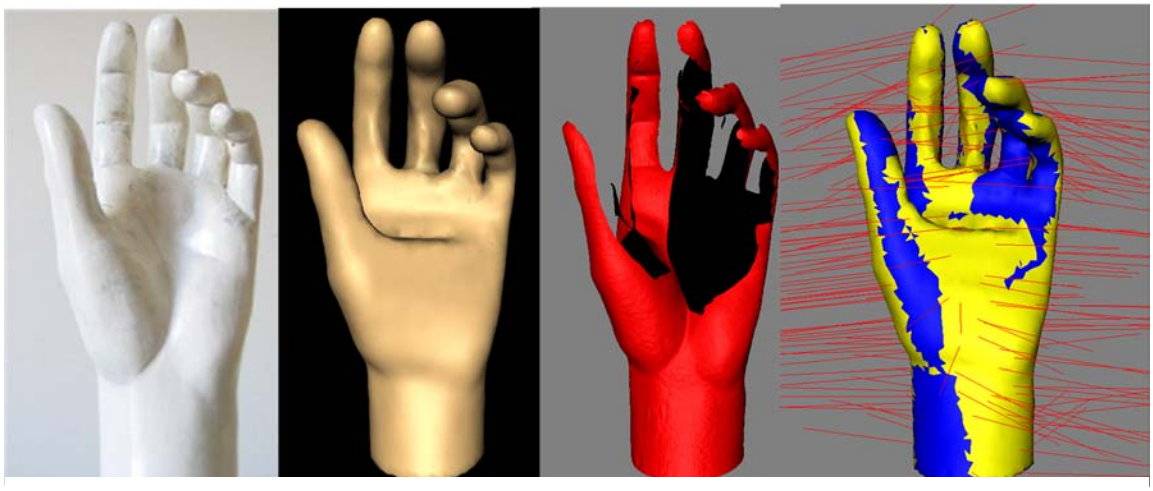
fusion process should take advantage of the benefits of one technique to compensate for the drawbacks of the other. For this specific example, the fingers can be recovered using the silhouette information, probably with some additional cavity information whenever it is made available by scanned range data. The cavity information of the palm on the other hand is provided by optical triangulation and the holes are patched up wherever this information is missing.

The acquired range data actually contains more information about the object shape than the one represented as a triangulated surface. The triangle mesh obtained by shape from optical triangulation is the part of the true object surface that can be sensed by the scanner without any obstruction. The range data also contains information about where the object surface can not lie. This is depicted in Fig. 4.2d. Each range point sampled on the object, that appears as a vertex on the optical triangulation model of Fig. 4.2c, defines a scan line segment that joins the range point and its projection on the camera screen. No part of the object surface or volume normally intervenes these line segments. We observe in Fig. 4.2d that the line segments intersect the silhouette model at parts where the surface is to be carved. Even on some parts of the palm where no range point was sampled, there are line segments intersecting the surface. These line segments actually target at different locations on the surface, but the way they trace the space provides additional information about the cavities of the palm where normally no range point was sampled.

The goal of the fusion process is to combine all the available information to infer 3D geometry provided by the silhouette model and the range data in the smoothest possible manner. This goal can be achieved by using our surface deformation framework. We start off with the initial silhouette model which is then deformed under the guidance of the external forces exerted by the scan lines of the range data. The surface deformation algorithm is basically the same as the one that we have described in Chapter 2. The initial silhouette model can be obtained by using the deformation-based technique that we have

described in Chapter 3, or by any other technique that would produce a topologically correct shape model that is eligible to further deformation.

The most critical part of the algorithm is to determine the appropriate external forces that will deform the initial silhouette model towards its final shape as faithfully as possible to the true object surface. This requires, prior to the deformation process, identifying the scan lines that intersect the initial silhouette model



(a) **(b)** **(c)** **(d)**

Figure 4.2. Illustration of the fusion process. (a) Original Hand image, (b) The silhouette model, (c) Triangulated range surface, (d) Range scan lines: Triangles (in yellow) intersected by scan lines need to be carved further whereas triangles (in blue) have no associated range information (scan lines are actually denser than those displayed in the figure).

4.2.1 Assignment of Carvers

A scan line segment that joins a range point and its projection on the camera screen ideally does not intersect the true object surface except for the point itself. However since the true object surface lies inside the visual hull, some of these line segments, that we will

refer to as *carvers*, may intersect the initial silhouette model. Finding these carvers and their intersections with the model gives an indication of where, how much and in which direction the surface is to be deformed.

A scan line can intersect the silhouette model at more than one triangles, but normally targets only one. We refer to a scan line which targets a triangle, as a *primary carver* of that triangle. If this scan line also intersects some other triangle, then it is called a *secondary carver* of that triangle. Some triangles may have more than one primary and secondary carvers while others may have none. Triangles with no carver at all correspond to those parts of the shape where we can rely only on silhouette information. Triangles having only secondary carvers are parts of the model that need to be carved, where we do not however have reliable information for deformation.

The procedure for finding initial carvers searches and assigns a primary carver to each triangle if it has one. Otherwise, a secondary carver is sought and assigned if exists. In Fig. 4.3, we demonstrate the initial assignment of carvers, where we observe on the silhouette model of the Hand object, the triangles with primary and secondary carvers and those with no carver, in different colors. Finding carvers involves computing intersections of the scan line segments with the surface triangles for which there exists in the literature various efficient algorithms [39].

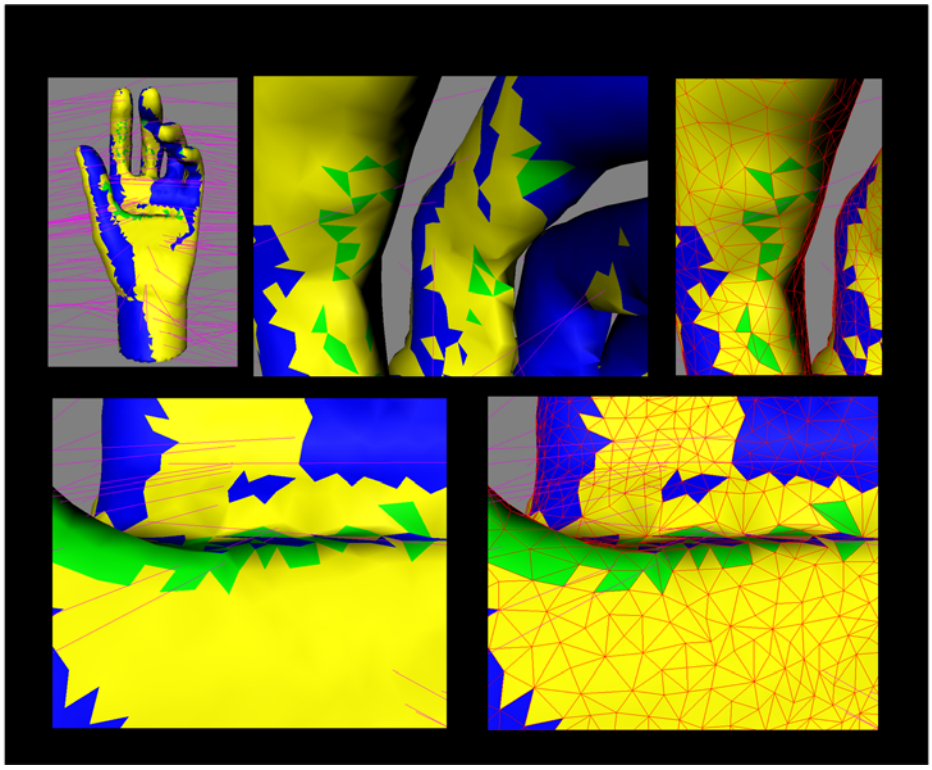


Figure 4.3. Carver assignment. Triangles with secondary, primary, and no carvers are displayed in green, yellow, and blue, respectively.

The carver assignment is initially performed on a relatively coarse silhouette model as compared to the resolution of the range data. This reduces the risk of missing some triangles and leaving them without any carver during the assignment process due to possible deficiencies in the range data or due to a resolution mismatch. After this initial assignment at low resolution, the silhouette model is refined so as to match the resolution of the range data by properly adjusting the minimum edge length $\varepsilon_{\min}$ and restructuring the deformable mesh via local mesh transform operations. The operations, edge collapse, edge split and edge flip, are applied only to those vertices having carvers. For each new triangle created by an edge split operation, a new carver is sought within the scan line

neighborhood of the parent triangle as demonstrated in Fig. 4.4. If no carver is found, the carver of the parent triangle, primary or secondary, is assigned to the triangle. Searching carvers within a local neighborhood is very fast, thus this two-step carver assignment procedure not only gives robustness but also significantly improves the overall computational efficiency.

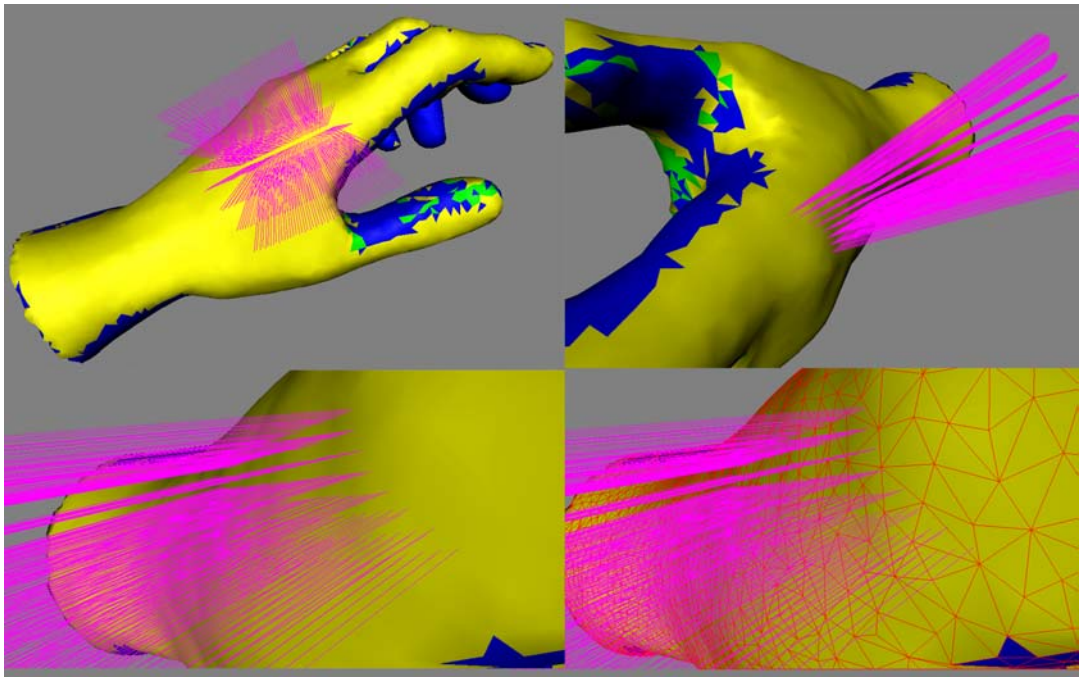


Figure 4.4. Scan line neighborhood of a triangle, used for update of carvers.

4.2.2 Computation of External Force

The external force $\mathbf{F}_{ext}(P,B)$ on each vertex P, that deforms the silhouette model towards the boundary B, is computed based on the carvers assigned to the triangles that share the vertex,

$$\mathbf{F}_{\text{ext}}(P,B)=\left(\frac{\sum_{i=1}^{M}\mathbf{F}_{\text{ext}}(C_i,B)}{M}\cdot\mathbf{N}(P)\right)\cdot\mathbf{N}(P) \tag{4.1}$$

where $\{C_i\}$, $i = 1,\ldots, M$, are the carvers of the triangles $\{T_i\}$ that are common to the vertex P, and $\mathbf{N}(P)$ denotes the surface normal vector. Each carver C_i exerts an external force $\mathbf{F}_{\text{ext}}(C_i,B)$ on vertex P. The average of all such external forces exerted by the neighboring carvers, when projected onto the surface normal, gives the overall external force at P. The direction of the external force is set to be perpendicular to the deformable surface. If a triangle has no carver, its contribution to the external forces of its vertices is assumed to be zero.

Each external force component $\mathbf{F}_{\text{ext}}(C_i,B)$ is determined based on positioning of the vertex P with respect to the boundary and its magnitude should be bounded above by $\varepsilon_{\min}/2$. We have considered three alternative ways of computing $\mathbf{F}_{\text{ext}}(C_i,B)$ as illustrated in Fig.4.5. The first one is based on the Euclidean signed distance of the plane S_{T_i} of the triangle T_i to the range point R_i that the carver C_i targets,

$$\mathbf{F}_{\text{ext}}(C_i,B)=\begin{cases}\text{Dist}(S_{T_i},R_i)\mathbf{N}(T_i) & \text{if } \left|\text{Dist}(S_{T_i},R_i)\right|<\dfrac{\varepsilon_{\min}}{2}\\ \dfrac{\varepsilon_{\min}}{2}\mathbf{N}(T_i) & \text{if } \text{Dist}(S_{T_i},R_i)\geq\dfrac{\varepsilon_{\min}}{2}\\ -\dfrac{\varepsilon_{\min}}{2}\mathbf{N}(T_i) & \text{if } \text{Dist}(S_{T_i},R_i)\leq-\dfrac{\varepsilon_{\min}}{2}\end{cases} \tag{4.2}$$

where $\mathbf{N}(T_i)$ denotes the unit vector perpendicular to triangle T_i. Note that the surface normals of the range points, calculated over the optical triangulation models, are not taken account in this formulation. The second alternative for computing $\mathbf{F}_{\text{ext}}(C_i,B)$ makes use of this additional information about the target boundary. It is based on the Euclidean signed

distance of the vertex P to the tangent plane S_{R_i} at the range point R_i that the carver C_i targets,

$$\mathbf{F}_{\text{ext}}(C_i, B) = \begin{cases} \text{Dist}(S_{R_i}, P)\mathbf{N}(R_i) & \text{if } \left|\text{Dist}(S_{R_i}, P)\right| < \dfrac{\varepsilon_{\min}}{2} \\ \dfrac{\varepsilon_{\min}}{2}\mathbf{N}(R_i) & \text{if } \text{Dist}(S_{R_i}, P) \geq \dfrac{\varepsilon_{\min}}{2} \\ -\dfrac{\varepsilon_{\min}}{2}\mathbf{N}(R_i) & \text{if } \text{Dist}(S_{R_i}, P) \leq -\dfrac{\varepsilon_{\min}}{2} \end{cases} \tag{4.3}$$

where $\mathbf{N}(R_i)$ denotes the normal vector at the range point R_i. In both cases, the strength of the external force is defaulted to $\varepsilon_{\min}/2$ for secondary carvers.

Both alternatives involve computing an estimate of the distance of the deformable mesh to the object boundary (see Fig. 4.5). The second alternative relies on the distance between a vertex of the deformable mesh and the range surface. Hence it locally approximates the range surface with the tangent plane of the range point. However, this is not always a reliable approximation and may yield over-carving, especially if the carver range point is distant to the vertex and the vertex does not evolve by deformation towards that range point, as it usually happens with oblique carvers.

The first alternative on the other hand relies on the distance between a range point and the deformable surface. This is not as accurate as the second one since it does not make use of the available range surface orientation. However, it can be used to constrain the strength of the external force in order to avoid over-carving in cases where the second alternative fails. Thus we end up with the third alternative which is actually a combination of the first two. We compute the external force as defined in Eq. 4.1 in two alternative ways using Equations 4.2 and 4.3 and use the one that gives a force with smaller magnitude. This formulation is the one that we have used in our experiments since it yields more reliable

reconstructions as compared to the others. In the experimental results section, we will also provide comparative results for these three alternatives.

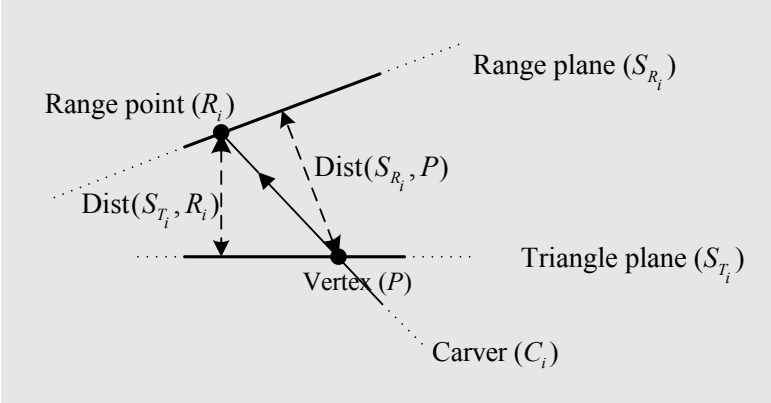


Figure 4.5. Illustration of the two alternatives to compute an estimate of the distance of the deformable mesh to the object boundary.

4.2.3 Update of Carvers

As the deformable mesh evolves, its geometry and structure change. Thus the carver information has to be updated at each iteration. If a primary carver no longer intersects its triangle, a new carver is sought for that triangle. The search for a new carver is fast and affordable, since it suffices to search only in a small scan line neighborhood of the invalid carver (see Fig. 4.4). If a primary carver can not be found, the triangle keeps the old one. The update procedure for secondary carvers is basically the same, but in case a new carver can not be found to replace an invalid secondary carver, the associated triangle is left without carver for the rest of the iterations, since secondary carvers do not provide reliable targets for the deformable mesh to rely on. As the deformable mesh evolves, due to possible split operations, new triangles appear on the restructured mesh. The carver of a

new triangle is also sought within the scan line neighborhood of the carver of its parent. If a carver can not be found, the new triangles inherit the carvers of their parents.

4.2.4 Algorithm

The input to the surface deformation algorithm is the initial silhouette model associated with carvers along with one or more triangulated range surfaces. The initial model is iteratively deformed under the guidance of the external and internal forces, each time restructuring the deformable mesh via local mesh transform operations. The vertices with initially no neighboring carver and those that no longer move during surface evolution are inactivated. The deformation process continues until all vertices become so. The overall algorithm can be expressed as follows:

Iterate

- Move each active vertex with the external force computed by Eq. 4.1
- Regularize the active vertices of the mesh using Eq. 2.5,
- Collapse active edges with length smaller than ε_{min},
- Split active edges with length exceeding ε_{max},
- Flip active edges where necessary, favoring the vertices with valence close to 6,
- Update carvers
- Inactivate vertices that no longer move

Until all vertices are inactivated

Once the algorithm converges, the obtained mesh is regularized by the Laplacian operator on parts of the surface, that are recovered using secondary carvers. This post-

processing smoothes out most of the redundant irregularities which may appear on the reconstructed mesh due to unreliable surface information provided by secondary carvers. Finally the whole mesh is faired by using Taubin's technique that regularizes the vertex distribution of the mesh while preserving its details [18].

4.3. Experimental Results

We have tested our fusion technique on three real objects. The original images of these objects, which are Elephant (made of wood), Greek (made of stone) and Hand (made of plastic) are displayed in Fig. 4.6. The resolution of images acquired for both silhouette and range data is 2000 by 1310 pixels. Table 4.1 lists the number of images acquired for each object.

Figure 4.6: Original images of Elephant, Greek, and Hand objects.

Table 4.1: Number of images taken for each object.

Model	Silhouette	Laser 1	Laser 2	Laser 3
Greek	72	360	-	-
Elephant	72	360	360	-
Hand	72	180	180	180

Figures 4.7-4.10 display the steps/results of fusion for the Elephant object. In Figure 4.7, we observe the initial silhouette model from different views with assigned carvers. Fig. 4.8 displays the triangulated range surfaces obtained from two separate scans of optical triangulation. The silhouette models and those obtained by fusion are compared in Figure 4.9. More views from the fused reconstruction is provided in Fig. 4.10. The obstructed inner faces of the legs and the trunk and though not clearly visible at the depicted angle, the top sections of the model, are missing from the optical triangulation and are complemented with the silhouettes in the final fused reconstruction

.

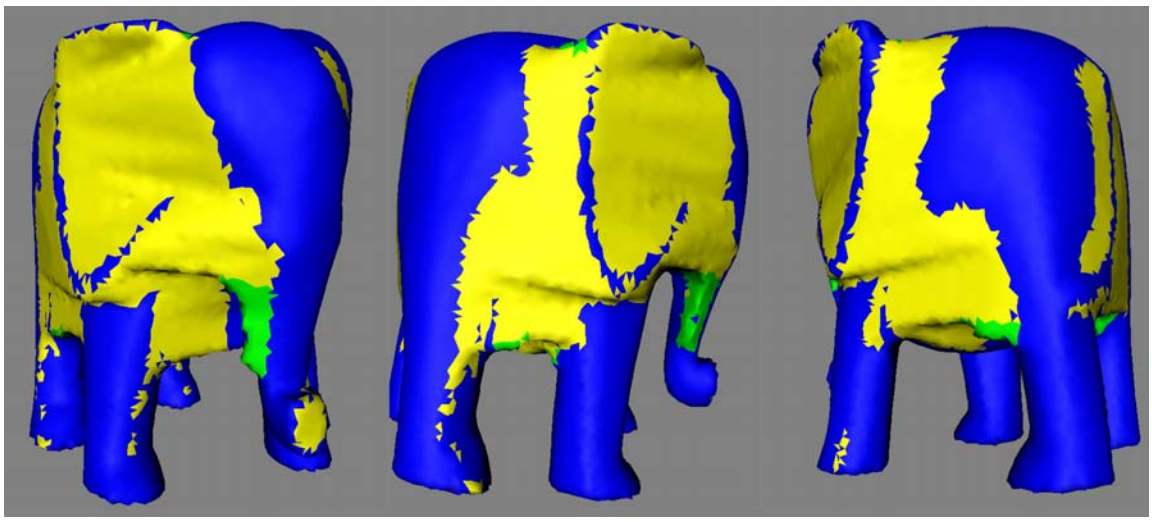


Figure 4.7: Initial carver assignment on silhouette models of the Elephant. Surface parts in yellow and green represent triangles with primary and secondary carvers, respectively, whereas the blue parts represent the triangles with no carver.

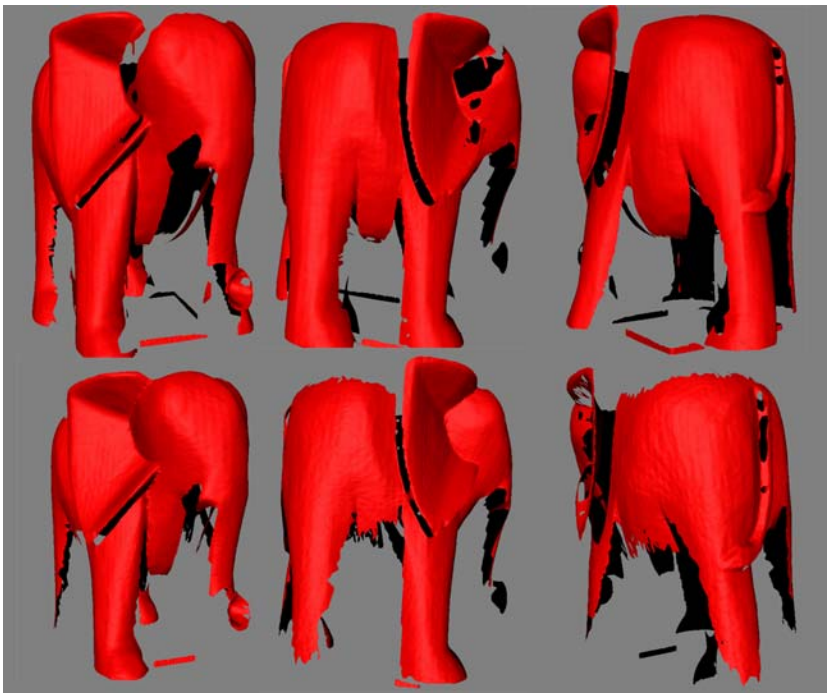


Figure 4.8: Triangulated range surfaces obtained from two separate scans of optical triangulation.

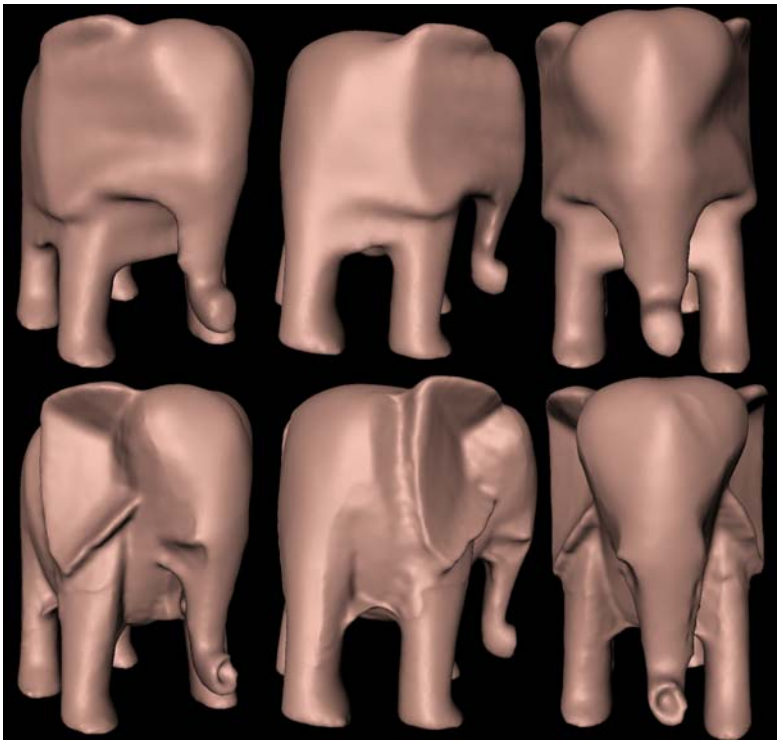


Figure 4.9: Silhouette models (top) vs. fused reconstructions (bottom) of the Elephant.

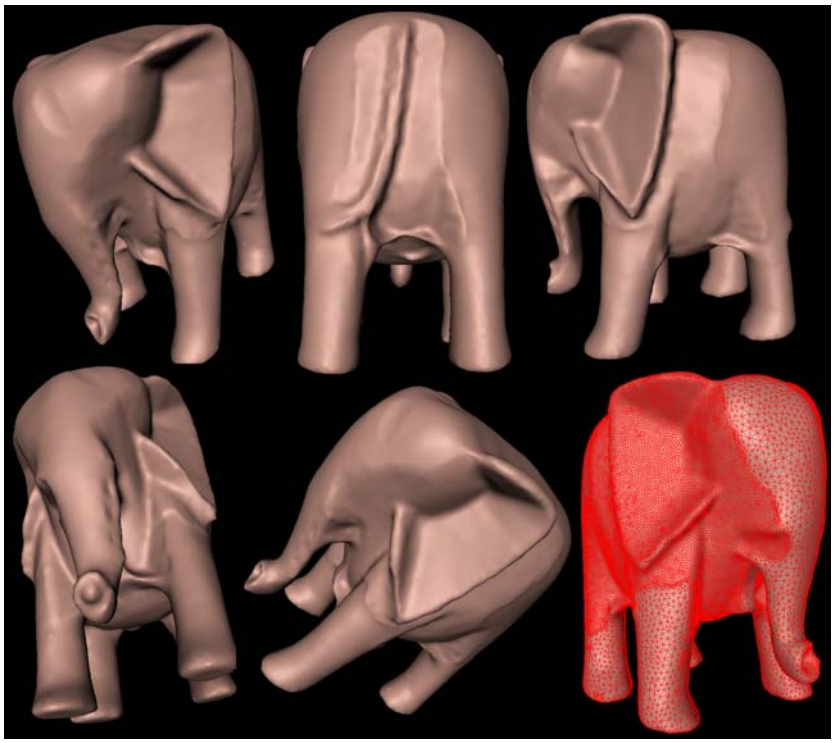


Figure 4.10: Fusion results of the Elephant from various views.

In Figure 4.11 we display the optical triangulation surface (in white color) as overlaid by the silhouette model (in blue color) of the Greek object. Ideally all range points are supposed to be sampled inside the silhouette model which bounds the true object surface. However as observed some range points (white) may fall outside the visual hull due to calibration errors or acquisition noise.

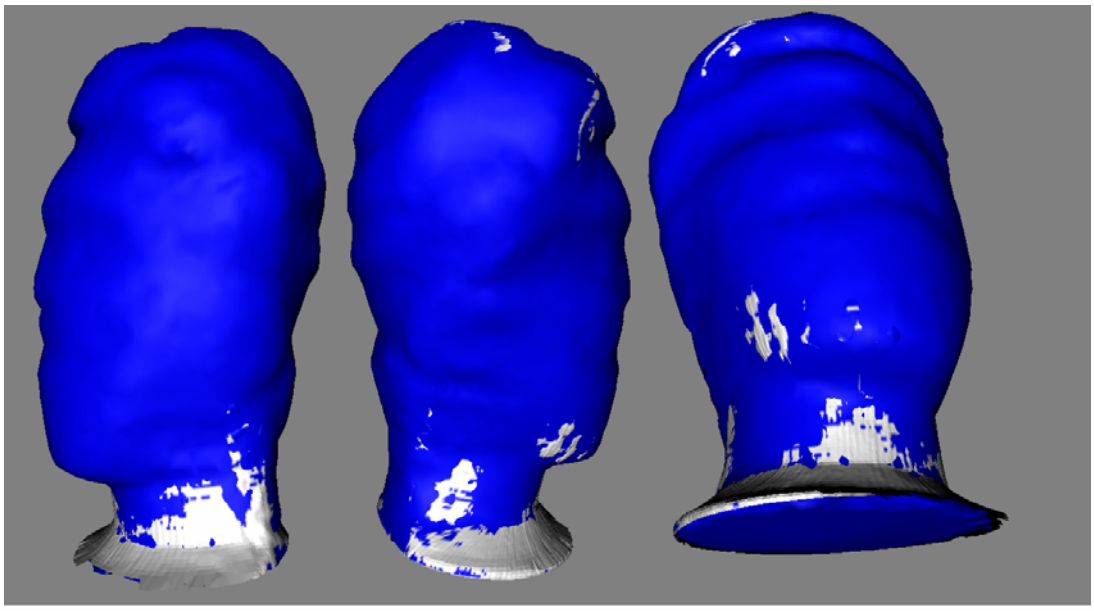


Figure 4.11: The optical triangulation surface overlaid by the silhouette model of the Greek object. The blue color represents visual hull whereas white colors are for OUT range points with respect to the silhouette model.

Figures 4.12-4.17 display the steps/results of fusion for the Greek object. In Figure 4.12, we observe the initial silhouette model from different views with assigned carvers. Fig. 4.13 displays the range surfaces obtained by optical triangulation. The silhouette models and those obtained by fusion are compared in Figure 4.14. More views from the fused reconstruction is provided in Fig. 4.15. Finally, Figures 4.16 and 4.17 compare the final models reconstructed at two different resolutions. The silhouette reconstruction of the Greek object typically lacks detailed features of the face while the optical triangulation exhibits many holes. The fusion reveals a watertight reconstruction that includes accurate cavity shape deduced from the optical triangulation. The final model constructed at high resolution achieves the desired level of detail.

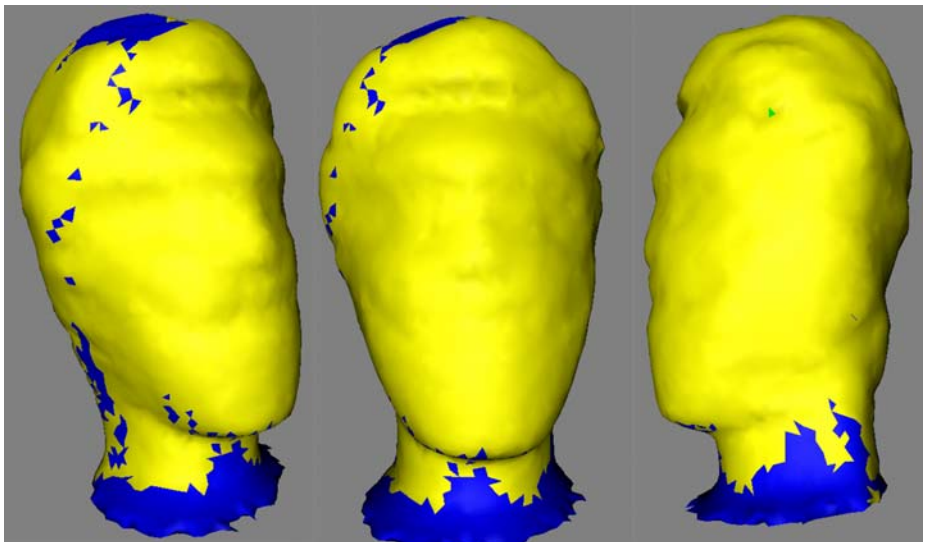


Figure 4.12: Initial carver assignment on silhouette models of the Greek. Surface parts in yellow and green represent triangles with primary and secondary carvers, respectively, whereas the blue parts represent the triangles with no carver. Note that secondary carvers are very few for the Greek object.

Figure 4.13: Range surface of the Greek object, obtained by optical triangulation.

.

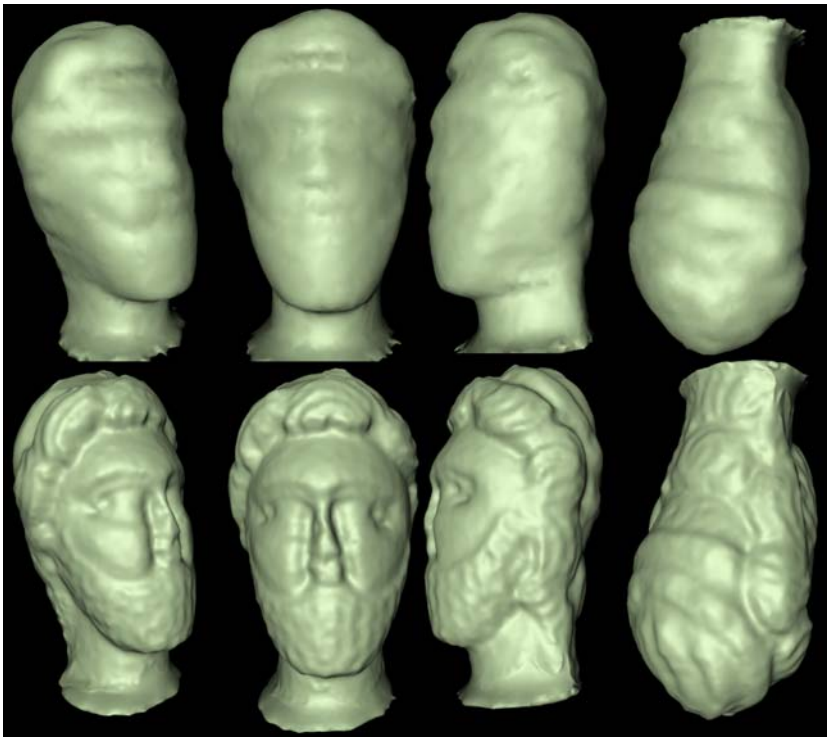


Figure 4.14: Silhouette models (top) vs. fused reconstructions (bottom) of the Greek object.

Figure 4.15: Fusion results of the Greek object from various views.

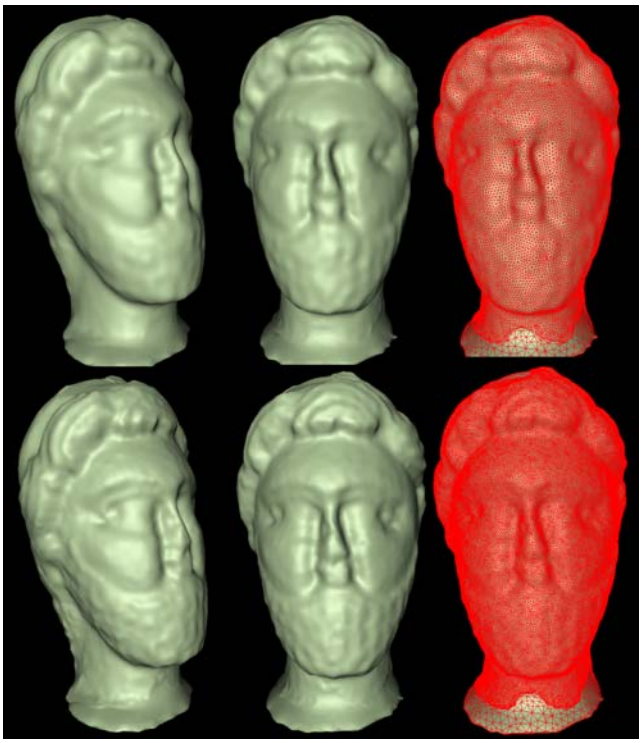


Figure 4.16: (Top vs. bottom) Low vs. high resolution reconstructions.

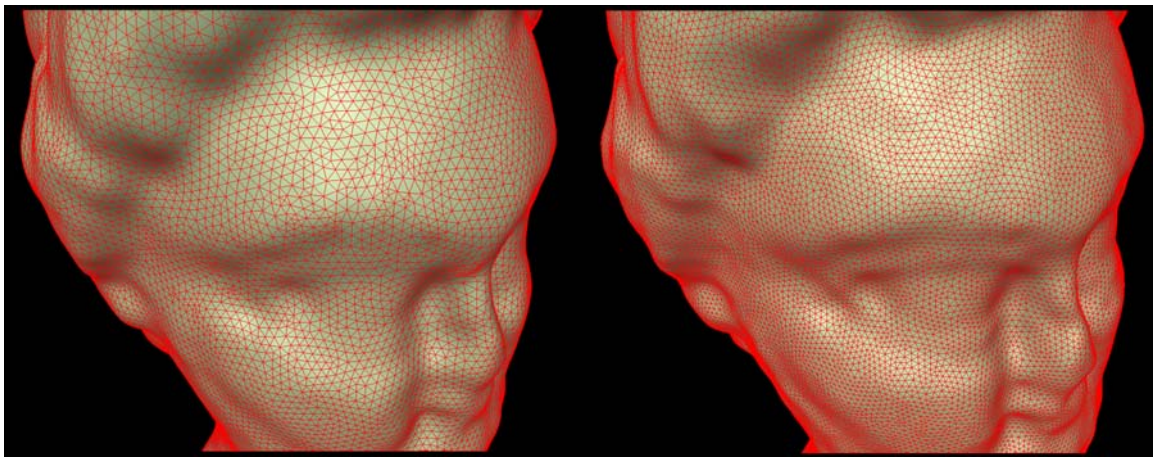


Figure 4.17 (Left vs. right) Low vs. high resolution reconstructions in wireframe.

Figures 4.18-4.22 display the steps/results of fusion for the Hand object. In Figure 4.18, we observe the initial silhouette model from different views with assigned carvers. Fig. 4.19 displays the triangulated range surfaces obtained from three separate scans of optical triangulation. The silhouette models and those obtained by fusion are compared in Figure 4.20. More views from the fused reconstruction is provided in Fig. 4.21. As we observe from figures, the fingers of the Hand are mostly recovered by silhouettes whereas the cavity information of the palm mainly comes from the range data in the fused reconstructions. Finally, in Fig. 4.22, we provide fusion results by using the two alternative ways of computing the external force as given by Equations 4.2 and 4.3 of Chapter 4. As observed the first alternative yields problems especially in reconstruction of the fingers whereas the second overcarves the palm of the Hand.

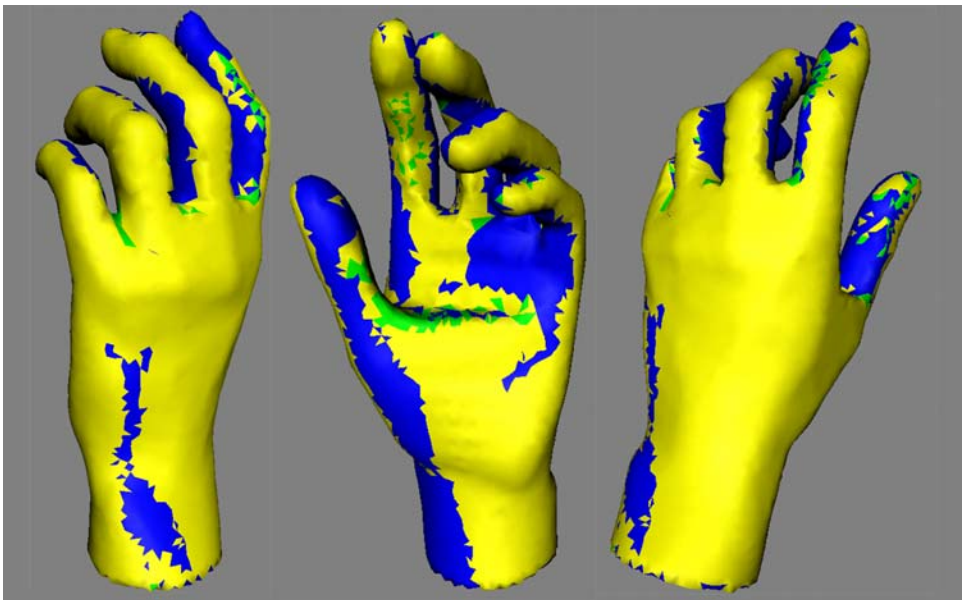


Figure 4.18: Initial carver assignment on silhouette models of the Hand. Surface parts in yellow and green represent triangles with primary and secondary carvers, respectively, whereas the blue parts represent the triangles with no carver.

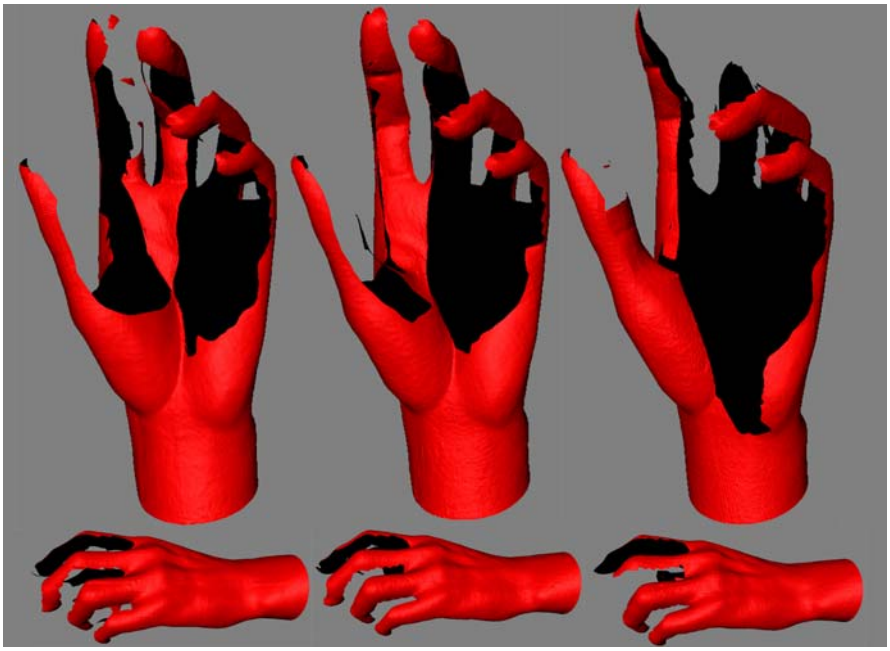


Figure 4.19: Triangulated range surfaces obtained from three separate scans of optical triangulation.

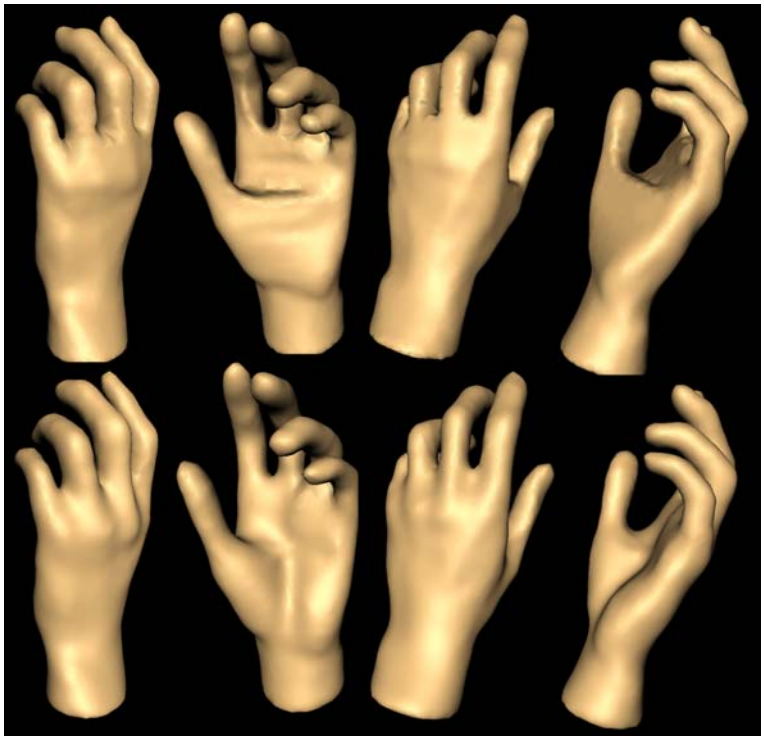


Figure 4.20: Silhouette models (top) vs. fused reconstructions (bottom) of the Hand object.

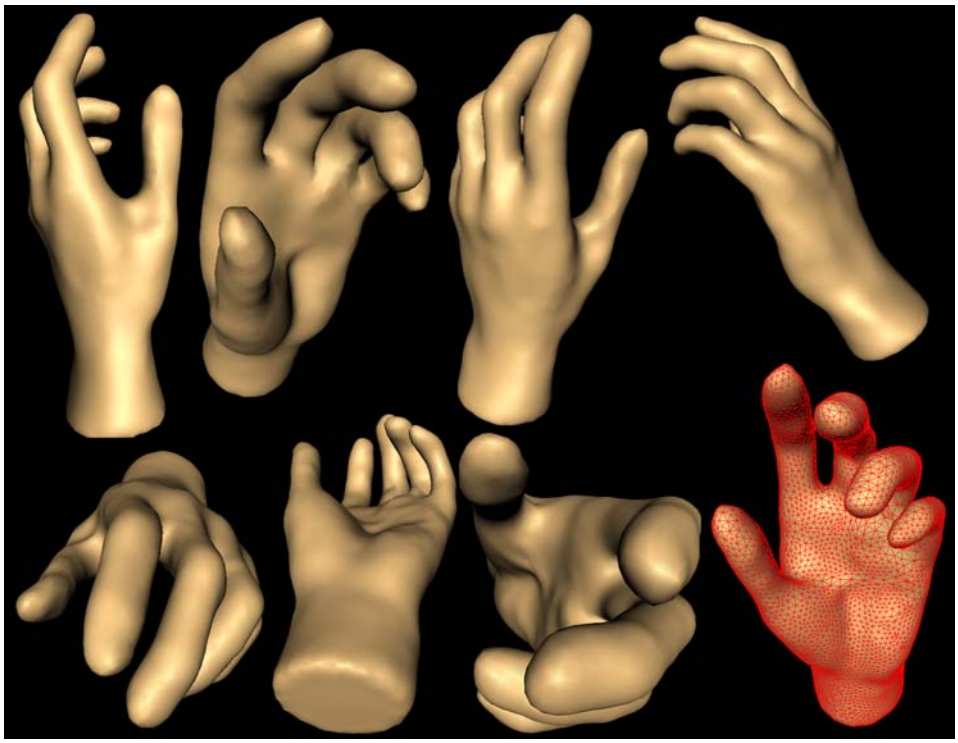


Figure 4.21: Fusion results of the Hand from various views.

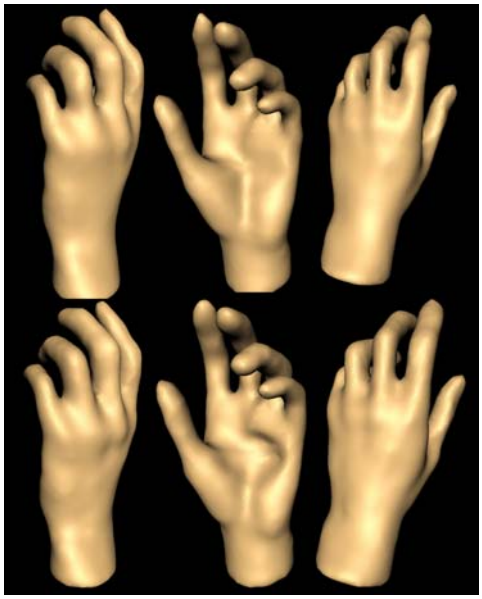


Figure 4.22: Fusion results by using the two alternative ways of computing the external force as given by Equations 4.2 and 4.3 of Chapter 4. As observed the first alternative yields problems especially in reconstruction of the fingers whereas the second overcarves the palm of the Hand.

Finally, Table 4.2 displays the structural information for the final mesh reconstructions and the compact execution times of the fusion algorithm measured on an Intel Pentium-4M 3GHz PC with 512MB of RAM.

Table 4.2: Structural mesh information and execution times for fusion.

Object	**Vertices (#)**	**Faces (#)**	**Resolution**	$\varepsilon_{\min}$	**Time (secs)**
Elephant	30992	61980	High	1.3/204 = .006	485
Greek	25832	51660	Low	.75/94 = .008	407
Greek	56496	112988	High	.5/94 = .005	840
Hand	6393	12782	High	2/176 = .011	66

Chapter 5

CONCLUSIONS AND FUTURE WORK

This work has described a generic surface deformation framework for the 3D shape recovery problem. An initial surface model, represented as a triangle mesh, is iteratively deformed towards the target boundary in a smooth manner under the guidance of external and internal forces, by restructuring the deformable mesh at each iteration using local mesh transform operations. By appropriately defining the external forces, the deformation framework can be applied to any type of data that can be used to infer information about 3D geometry. One limitation of the presented framework is that the initial topology of the deformable mesh can not be altered, however this limitation can be overcome by employing special procedures to detect possible splitting and merging as proposed in [31].

We have first applied the proposed deformation framework to the shape from silhouette problem for 3D reconstruction of static objects. Since our deformation technique always produces topologically correct mesh representations, the obtained models are eligible to further deformation. The fusion technique makes use of this initial robust silhouette model, that describes the object visual hull, and then deforms it further to amend missing concavities. The aim of fusion is actually to compensate for the problems associated with each method by the benefits of the other. The experiments show that it is possible to produce robust and accurate reconstructions. The most prominent property of the presented fusion method is the ability to construct cavity-sensitive and hole-free models.

One restraining factor in the overall system performance regarding fusion was found to be hardware related. The fixed single-striped range scanner used in our system can perform

inadequately in delivering well distributed range samples for some objects. Although our deformation-based fusion method can more successfully handle deficiencies of the acquired range data when compared to the other fusion techniques proposed in the literature, the quality of the reconstructions can even further be improved by employing more sophisticated range scanners such as the ones that produce full range images as discussed in Chapter 4.1. The employment of such scanner setups can greatly enhance the distribution of the range data and permit fusion at higher resolutions, which would produce even sharper results.

A new and emerging problem in the field of 3D shape recovery is time-varying scene modeling. The primary challenge here concerns the size of the resulting representations. A time-varying scene sampled at a standard rate of 30 frames per second would yield enormous 3D model data if no particular care is shown to exploit redundancies between consecutive time frames. The current solution to these problems is the use of object-specific models and to animate the dynamic scene through animation parameters. However this approach is not applicable to general dynamic scenes. The real challenge here is to generate once an initial model for the object under consideration with arbitrary geometry and then to track its motion (or deformation) through time. In this respect, time-varying mesh representations with a connectivity as fixed as possible, but with changing vertex positions, would certainly provide enormous efficiency both for storage, processing and visualization. There have been very few attempts to achieve such time-consistent representations such as in [24], but these works are quite premature and can obtain time-consistent meshes only for very short time intervals. Since our deformation framework is based on Lagrangian approach, the connectivity information is not lost through iterations and thus the presented framework can also be employed for building efficient time varying surface representations, that we plan to address as future work.

BIBLIOGRAPHY

[1] J. Montagnat, H. Delingette, N. Ayache N., A review of deformable surfaces: topology, geometry and deformation, Image Vision and Computing, Vol. 19, (2001), 1023-1040.

[2] Y. Yemez, and F. Schmitt, 3D Reconstruction of Real Objects with High Resolution Shape and Texture, Image and Vision Computing, Vol. 22, (2004), 1137-1153.

[3] Y. Yemez, and F. Schmitt, Progressive 3D Meshes from Octree Particles, Proceedings of Int. Conf. on 3D Digital Imaging and Modeling 3DIM'99, Canada, Ottowa, (1999), 290-299.

[4] Y. Yemez and F. Schmitt, Multilevel Representation and Transmission of Real Objects with Progressive Octree Particles, IEEE Trans. on Visualization and Computer Graphics, Vol. 9., No. 4, (2003), 551-569.

[5] S. Tosovic, R. Sablatnig, and M. Kampel, On Combining Shape from Silhouette and Shape from Structured Light, 1st International Symposium on 3D Data Processing Visualization and Transmission, (2002), 108-118.

[6] C. H. Chien and J. K. Aggarwal, Volume/Surface Octrees for the Representation of Three-Dimensional Objects, Computer Vision Graphics and Image Process., Vol. 36, No. 1, (1986), 100-113.

[7] C.H. Chien and J. K. Aggarwal, Identification of 3D Objects from Multiple Silhouettes Using Quadtrees/Octrees, Comp. Vision Graphics Image Process., Vol. 36, No. 2/3, (1986), 256-273.

[8] R. Szeliski, Rapid Octree Construction from Range Sequences, Computer Vision, Graphics and Image Processing, Vol. 58, No. 1, (1993), 23-32.

[9] B. Curless, Overview of Active Vision Techniques, Proc. SIGGRAPH 99 Course on 3D Photography, (1999).

[10] B. Curless and M. Levoy, A Volumetric Method for Building Complex Models from Range Images, SIGGRAPH ’96, Vol. 30, (1996), 303-312.

[11] S. Rusinkiewicz, O. Hall-Holt, M. Levoy, Real-Time 3D Model Acquisition, SIGGRAPH ‘02, Vol. 21, No. 3, (2002), 438-446.

[12] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira , M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, D. Fulk, The Digital Michelangelo Project: 3D Scanning of Large Statues, SIGGRAPH’00, (2000), 131-144.

[13] J. Park, G. N. DeSouza, and A. C. Kak, Dual-Beam Structured-Light Scanning for 3-D Object Modeling, 3rd International Conference on 3D Digital Imaging and Modeling (3DIM’01), Canada, Quebec City, (2001).

[14] C. Liska, R. Sablatnig, Adaptive 3D Acquisition Using Laser Light, Proc. of Czech Pattern Recognition Workshop”, (2000), 111-116.

[15] J. Davis and X. Chen, A Laser Range Scanner Designed For Minimum Calibration Complexity, Proc. of 3rd Intl. Conf. on 3-D Digital Imaging and Modeling, (2001), 91-98.

[16] U. Dhond and J. Aggarwal, Structure from Stereo—A Review, IEEE Trans. Systems, Man, and Cybernetics, Vol. 19, No. 6, (1989).

[17] R. Zhang, P. Tsai, J.E. Cryer, M. Shah, Shape from Shading: A Survey, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 21, No. 8, (1999) 690–706.

[18] G. Taubin, A Signal Processing Approach To Fair Surface Design, Computer Graphics Proceedings, (1995), 351-358.

[19] M. A. Magnor, B. Goldlücke, Spacetime-coherent Geometry Reconstruction from Multiple Video Streams, Int. Symp. 3DPVT, (2004), 365-372.

[20] Y. Matsumoto, K. Fujimura, T. Kitamura, Shape-from-Silhouette/Stereo and Its Application to 3-D Digitizer, Proceedings of Discreet Geometry for Computing Imagery, (1999), 177-190.

[21] C. H. Estaban and F. Schmitt, Silhouette and Stereo Fusion for 3D Object Modeling, Computer Vision and Image Understanding, Vol. 96, No. 3, (2003), 367-392.

[22] M. Kaas, A. Witkin, D. Terzopoulos, Snakes: Active Contour Models, International Journal of Computer Vision, Vol. 1, No. 4, (1988), 321-332.

[23] R. Malladi, J. A. Sethian, B. C. Vemuri, Shape Modeling with Front Propagation: A Level Set Approach, IEEE Trans. Pattern Analysis and Mach. Intelligence, Vol.17, (1995), 158-175.

[24] K. Mueller, A. Smolic, P. Merkle, M. Kautzner, and T. Wiegand, Coding of 3D Meshes and Video Textures for 3D Video Objects, Proc. PCS 2004, Picture Coding Symposium, (2004).

[25] W.E. Lorensen, H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," Computer Graphics, Vol. 21, No. 4, (1987), 163-169.

[26] S. Savarese, H. Rushmeier, F. Bernardini, P. Perona, Implementation of a Shadow Carving for Shape Capture, 1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02), Italy, Padova, (2002).

[27] U. Yılmaz, O. Özün, B. Otlu, A. Mülayim, V. Atalay, Inexpensive and Robust 3D Model Acquisition System for Three-Dimensional Modeling of Small Artifacts, ISPRS CIPA 2003 New Perspectives To Save Cultural Heritage, Antalya, Turkey, (2003).

[28] J. Carranza, C. Theobalt, M. A. Magnor, H. Seidel, Free-viewpoint Video of Human Actors, ACM Trans. Graph. Vol. 22, No. 3, (2003), 569-577.

[29] D. Terzopoulos, A. Witkin, M. Kass, Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motions, Artificial Intelligence, Vol. 36, No. 1, (1988), 91-123.

[30] L. D. Cohen, On Active Contour Models and Balloons, CVGIP: Image Understanding, Vol. 53, (1991), 211-218.

[31] Y. Duan, L.Yang, H. Qin, D. Samaras, Shape Reconstruction from 3D and 2D Data Using PDE-based Deformable Surfaces, Proc. ECCV, (2004), 238-251.

[32] Y. Yemez and C. J. Wetherilt, A Volumetric Fusion Technique for Surface Reconstruction from Silhouettes and Range Data, submitted to Computer Vision and Image Understanding, (2006).

[33] B. Curless, M. Levoy, Better Optical Triangulation through Spacetime Analysis, IEEE Int. Conference on Computer Vision, (1995), 987-994.

[34] R.Szeliski, Real-Time Octree Generation from Rotating Objects, DEC, Cambridge Research Lab., Technical Report Series, 1990.

[35] L. P. Kobbelt, T. Bareuther, H. Seidel, Multiresolution Shape Deformations for Meshes with Dynamic Vertex Connectivity, Proc. EUROGRAPHICS, Vol. 19, (2000).

[36] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Mesh Optimization, ACM SIGGRAPH, (1993), 19-26.

[37] A. Laurentini, The Visual Hull Concept for Silhouette Based Image Understanding, IEEE Trans. On PAMI, No. 16(2), (1994), 150-162.

[38] F. Bernardini and H. E. Rushmeier, 3D Model Acquisition, Eurographics State of the Art Reports Proceedings,(2000), 41-62.

[39] J. Foley, A. Van Dam, Fundamentals of Interactive Computer Graphics, Addison-Wesley, (1984).

[40] Y. Sahillioğlu, Y. Yemez and V. Skala, 3D Shape Recovery and Tracking from Multi-Camera Video Sequences via Surface Deformation, SIU'2006.

[41] Y. Sahillioğlu and Y. Yemez, 3D Shape Recovery Using Surface Deformation, Workshop on Multimedia Content Representation, Classification and Security, Istanbul, September, (2007).

VITA

Yusuf Sahillioğlu was born in beautiful Mersin, Turkey on March 25, 1982. He graduated from Toros College, Mersin in 2000. He received his B.S. degree in Computer Engineering from Bilkent University, Ankara in 2004. In September 2004, he joined the M.S. Program in Electrical and Computer Engineering at Koç University, as a research and teaching assistant. Having received the M.S. degree in June 2006, he is now preparing to join University of Florida/USA for a Ph.D. degree in Computer Engineering.