A Volumetric Fusion Technique for Surface Reconstruction from Silhouettes and Optical Triangulation

by

Can James Wetherilt

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
September 2005

Koç University

Graduate School of Sciences and Engineering

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

Can James Wetherilt

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)

_____

Assist. Prof. Çağatay Başdoğan

_____

Prof. Lale Akarun

Date: _____

ABSTRACT

Optical triangulation, an active reconstruction technique, is known to be an accurate method but has several shortcomings due to occlusion and laser reflectance properties of the object surface that often lead to holes and inaccuracies on the recovered surface. Shape from silhouette, on the other hand, is a passive reconstruction technique that yields robust, hole-free reconstruction of the visual hull of the object. In this thesis, a hybrid surface reconstruction method that fuses geometrical information obtained from silhouette images and optical triangulation is proposed. Our motivation is to recover the geometry from silhouettes on those parts of the surface which the range data fail to capture. Silhouettes and laser range images of the object are acquired with a calibrated camera and re-projected onto a fixed 3D world coordinate system where the fusion process takes place. A volumetric octree representation is first obtained from the silhouette images and then carved by range points to amend the missing cavity information inherent in silhouette-based techniques. An average isolevel value on each corner of each surface cube in the carved octree structure is accumulated using partial surface triangulations obtained separately from range data and silhouettes. The marching cubes algorithm is applied for triangulation of the resulting isolevel surface and the final shape is constructed by fairing the 3D model. The performance of the proposed technique is demonstrated on several real objects.

Keywords: 3D reconstruction, shape from silhouette, shape from structured light, octree, isosurface extraction, volume carving, fusion

ACKNOWLEDGEMENTS

I would like to thank my supervisor Assist. Prof. Yücel Yemez for his guidance, help and encouragement without which this work would not have been completed.

It is a pleasure to thank Assist. Prof. Çağatay Başdoğan and Prof. Lale Akarun for taking part in my thesis committee and for their valuable comments and suggestions.

My sincere appreciation goes to friends whom I have been lucky enough to become acquainted with at Koç University and who have put up with me throughout the better part of the three-year odyssey. Special thanks to Ali Selim Aytuna, Umut Küçükkabak, Ferit Ozan Akgül, and Işıl Yıldırım, and all the other countless friends who have been there for me at good and bad times.

I am indepted to my family, my parents and *twinsis* Lale for enduring me all my life and for their infinite trust in me and encouragment to further my education.

I feel the person closest to me, Özge Artunç, deserves the most credit for her generous understanding, patience, and support.

I also would like to acknowledge all other friends who cared to keep asking me during all this time: "Haven't you finished your thesis yet?". It is a joy to be able to finally give a different answer.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF CODE LISTINGS

NOMENCLATURE

x_c	x-component of a point in camera coordinates
y_c	y-component of a point in camera coordinates
z_c	z-component of a point in camera coordinates
x_s	x-component of a screen coordinate
y_s	y-component of a screen coordinate
x_F	x-component of a point in the frame coordinate F
y_F	y-component of a point in the frame coordinate F
z_F	z-component of a point in the frame coordinate F
f	focal length of the camera
$\boldsymbol{R}$	Rotation Matrix used in the transformation from world to camera coordinates
$\breve{\boldsymbol{R}}$	The inverse of the rotation matrix $\boldsymbol{R}$
$\boldsymbol{t}$	Translation vector used in the transformation from world to camera coordinates
$f(.)$	Isolevel function (implicitly global)
R	Highest resolution level in the octree
G	Bilinearly interpolated value of a sub-pixelic point on the discrete image raster
H_n	n'th strip of range points gathered from a single optical triangulation
$\Re_i$	Range point of the sequence i
$S_{\Re_i}$	Octree node containing the range point $\Re_i$
O_i	Optical centre of the camera at the instance $\Re_i$ is scanned

L_i	Scan line associated with $\Re_i$
$\hat{f}(.)$	Local isolevel function for a given corner of a voxel
Δ_{ij}	The signed distance from the *i*'th corner to *j*'th triangle in a given voxel
C_i	*i*'th corner of a given cube
T_j	*j*'th triangle interpolated during shape from silhouette on a given cube
V_{jn}	n'th vertex on T_j
$\Re_{C_i}$	The closest range point to C_i
P_i	Plane perpendicular to the normal of and passing through the range point closest to C_i ($\overrightarrow{\Re_i}$)
l	Side length of a given voxel
$d(.,.)$	Signed distance function
$\boldsymbol{v}$	Position of a vertex in a triangular mesh
$\Delta\boldsymbol{v}$	Displacement applied to the vertex v during fairing
λ	Coefficient to Δv having a shrinking effect on the object surface
$\boldsymbol{v}'$	New position of v
μ	Coefficient to Δv having an unshrinking effect on the object surface

Chapter 1

INTRODUCTION

Today the art of inferring three-dimensional shapes of objects has become one of the major applications of Computer Vision. There are many areas concerned with 3D (three-dimensional) reconstruction among which VR (Virtual Reality) applications, digital demonstration of museum artifacts, machine vision, medical imaging are the most common. Techniques employed for 3D reconstruction of real objects are numerous and related literary work is profuse. In general, recognized methods can be collected under two groups: active and passive. 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, extract surface information by the use of images of the scene. Among the most common that fall into this category are the techniques known as Shape from Silhouette, Shape from Stereo, and Shape from Shading. Many results are available concerning reliable reconstruction of objects using these methods. However, there is still a growing need for improved reconstructions.

1.1 State-of-the-Art

Shape from Silhouette in the context of this work, basically draws shape information by back-projecting multiple silhouettes into world space as conical volumes and intersects these cones to generate a volumetric visual hull. Early examples of this technique are

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 robust output of this method constitutes a solid initial foundation for further volume carving or mesh deformations, depending on the representation used. The downside of this method is that it fails to capture hidden concavities. This deficiency has led to attempts to combine Shape from Silhouette with several other techniques, which alone do not produce complete or fully reliable model reconstructions. The common notion for integrating Shape from 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 the silhouettes and gradually recover the cavities with a cavity-sensitive method.

Methods of Shape from Stereo seek to find correlations on separate images using texture or colour information. This feature makes such techniques very sensitive to lighting conditions and renders them less effective as stand-alone methods. 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.

Shape from Shading methods are based on the diffusing properties of Lambertian surfaces. 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. Savarese et al. [26] integrate Shape from Shadow and Shape from Silhouette techniques in a volumetric space carving fashion, but the results show that the obtained reconstructions cannot compete with the accuracy of active methods.

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 still 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. They propose shaping such difficult parts of the object by carving via extraction of the background, which suggests a simple variation of fusing Shape from Structured Light and Shape from Silhouette.

Tosovic et al. [5] have attempted to combine Shape from Silhouette and Shape from Structured Light fully favouring an octree based combination that builds a single model without the need to convert the data structure of one method to the other. While promising, it lacks many specifications on crucial points such as how the volume triangulation is conducted and fails to produce clear results.

1.2 Contribution

In this work we explore an alternative to the known fusion techniques implemented and examined so far in the literature. A novel, hybrid 3D reconstruction scheme aimed at eliminating the shortcomings of the combined methods (see Figure 1.1) and enabling the generation of high quality, robust, and complete 3D models is explained. Without any prior assumptions about the object shape or orientation, data acquired from the two techniques Shape from Silhouette and Shape from Structured Light, are fused volumetrically and triangulated into a final mesh while preserving object topology. The major inspiration has been the work of Yemez and Schmitt [2] for building robust silhouette models and Levoy and Curless [10] for their important contributions to the engineering in the Shape from Structured Light related methods.

We are not aware of any published methods for combining Shape from Silhouette with Shape from Structured Light that both underpin its feasibility as a successful method and yield satisfactory results. This work's main contribution is to (i) show that it is possible to fuse silhouette-based and structured light based methods, (ii) provide examples of fusions conducted in this manner, and (iii) demonstrate the improvements of the fused results over the results obtained from the individual methods.

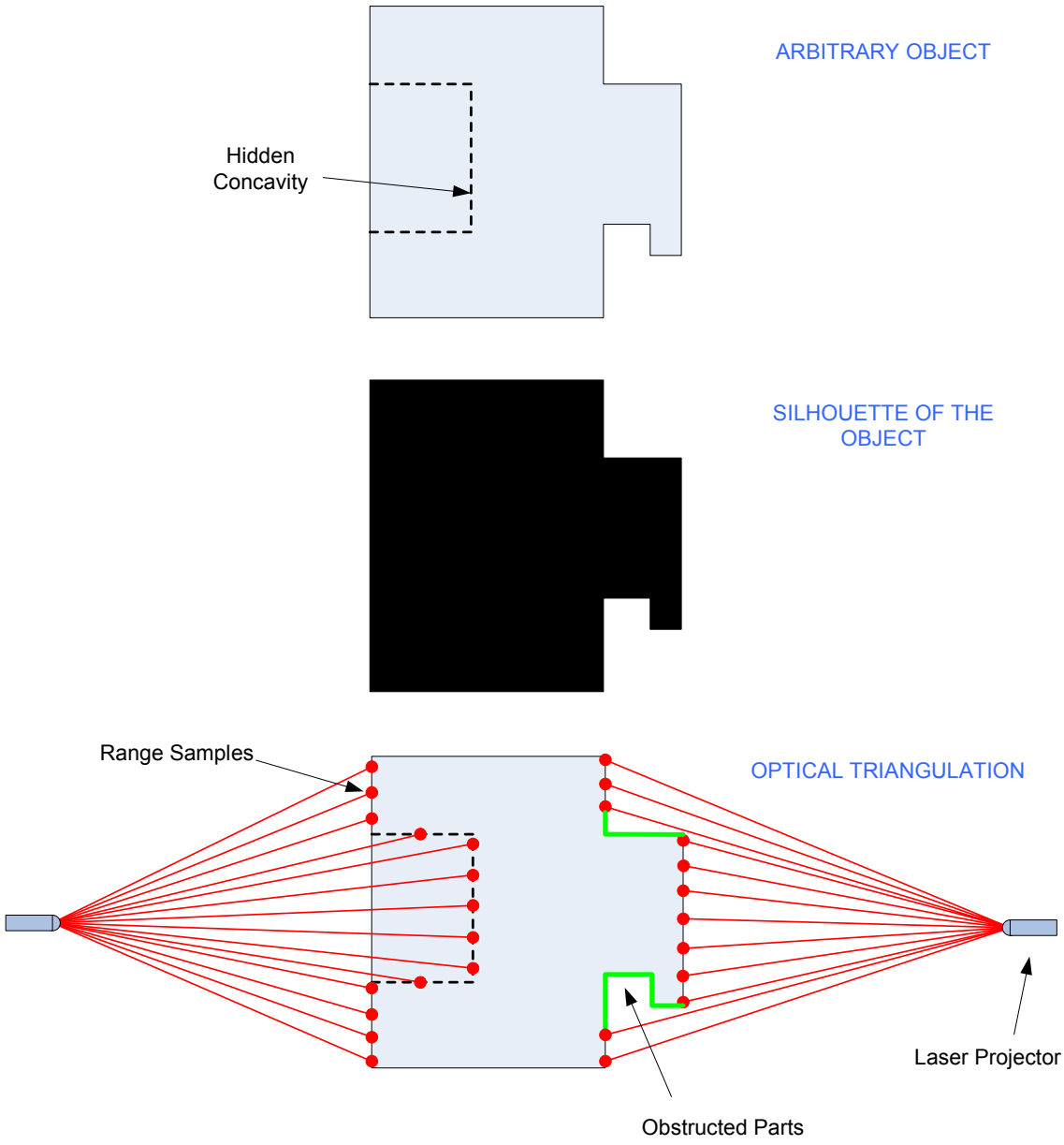

Figure 1.1: Motivation for merging the two methods, Shape from Silhouette and Shape from Structured Light. (Top) An arbitrary object with a hidden cavity is to be reconstructed. (Middle) The hidden cavity is not detected on the silhouette of the object. (Bottom) The range scanner produces range samples on the object surface that is sensitive to cavities. However, there remain occluded parts of the object that the range scanner cannot observe. The aim is to combine the data deducted from the two methods to obtain a more complete and robust reconstruction of the object than can be attained by using either one alone.

1.3 Overview and Organization

The reconstruction system is composed of a computer, a camera, single-stripe laser projectors, and a turntable. 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 rotation of the turntable and the image acquisition are controlled by the computer.

The proposed technique involves several important stages (see Figure 1.2). First, 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 reconstructed object is placed on the turntable and image acquisition is carried out for the silhouettes and the optical triangulation of each laser projector. The Shape from Silhouette and the Shape from Structured Light methods are implemented separately and later the obtained results are fused.

The Shape from Silhouette method is initiated with the extraction of the silhouettes from the obtained input images. Then, using the camera calibration parameters and the silhouettes a volumetric definition of the 3D visual hull of the object is obtained. Last, the volumetric data is triangulated into a mesh to be used in the fusion.

For the Shape from Structured Light method, first a sequence of 2D points is obtained from the input laser images and using the calibration parameters their 3D coordinates in the common reference frame is computed. Then, the cloud of 3D range points is triangulated into a mesh. Finally, this mesh is subdivided and faired to enrich the quality and quantity of the obtained results.

The fusion is carried out volumetrically by carving the superfluous parts of the visual hull (from the silhouettes) with the range points (from the optical triangulation). The

resultant object volume is triangulated and the final object shape is obtained after the mesh is decimated and faired.

In Section 2 the design of the acquisition system is described and calibration issues are reviewed. Shape from Silhouette and Shape from Structured Light related work are explained separately under Sections 3 and 4. In Section 5 the possible fusion techniques are investigated and the actual implementation is explained. In Section 6 the obtained results from Shape from Silhouette, Shape from Structured Light and the fused reconstructions are presented and compared. Finally in section 7 we provide concluding remarks and discuss possible future work.

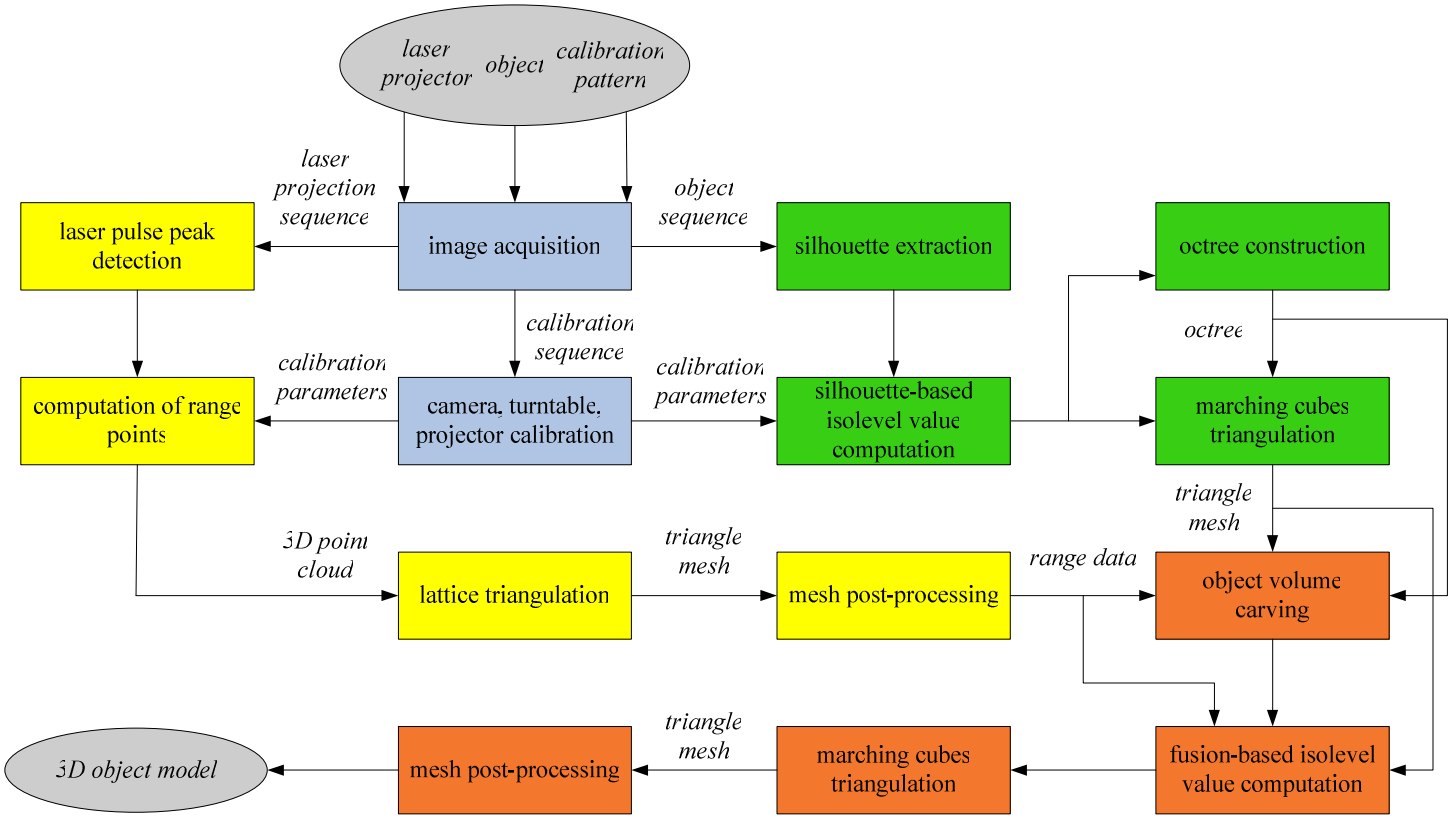

Figure 1.2: Block diagram of the object reconstruction scheme. Preliminary steps, Shape from Silhouette, Shape from Structured Light, and the final fusion related tasks are highlighted in blue, green, yellow, and orange, respectively.

Chapter 2

PRELIMINARY STEPS

Before 3D modeling can be carried out, some important issues have to be addressed namely: the design of the acquisition system, calibration, and silhouette extraction. Firstly, the 3D models of the object of interest are reconstructed through the use of images acquired by the acquisition system. Silhouette and laser images are required to reconstruct the individual models of the object which will be fused later. The quality of these images and the techniques that can be applied are affected by the structure of the acquisition system, which is explained in Section 2.1. Secondly, in order to be able to interpret and relate the data that can be drawn from the acquired images, the relative orientations of the camera, the laser projectors, and the turntable need to be known. These calibration issues are explained in Section 2.2. Finally, silhouette images need to be filtered in order to be able to extract the silhouettes of the object from the background. The success of the silhouette extraction, which is explained in Section 2.3, directly influences the quality of the Shape from Silhouette results.

2.1 Acquisition System

The acquisition setup consists of a CCD camera, a turntable, two laser projectors, a backdrop, and a computer the details of which are given in Table 2.1. The computer synchronizes the camera with the motion of the turntable and records the acquired data.

The position of the turntable, which supports only rotational movement around a vertical axis, can be set to an angular precision of one degree. The object to be reconstructed is positioned on the turntable in between the static camera and the backdrop and the laser projectors are pointed towards the object (see Figure 2.1). A single laser projector yields a planar laser beam which appears as a stripe on the object surface. The laser projectors are positioned so that the projected beams are as vertical to the rotational axis of the turntable as possible. Two laser projectors are used to improve the coverage of the object surface. Also, the same laser projector may be used several times in different orientations during the same acquisition to obtain several optical triangulations. The different laser projectors are not used simultaneously because of the difficulties involved in distinguishing the beams in discontinuous or occluded areas of the object. The laser projected images are acquired in the dark to discern the laser beams more easily. For the extraction of the model object from the background, the backdrop is used to create a contrast between the object and the rest of the scene. During the acquisition process, the laser projectors and the camera are maintained immobile and stable. The only actuated component in the system is the turntable on which the object is placed.

Table 2.1: Specifications for the different parts of the acquisition system.

System Component	Make	Model	Specification
Camera	Nikon	D1H	5 Megapixels of Resolution
Laser 1	Stocker & Yale [35]	Lasiris SNF Series	10 mW of diode power
Laser 2	Stocker & Yale [35]	Lasiris SNF Series	5 mW of diode power
Turntable	KAIDAN [30]	Magellan M-2500	Supports objects up to 1 meter and 90 kg.

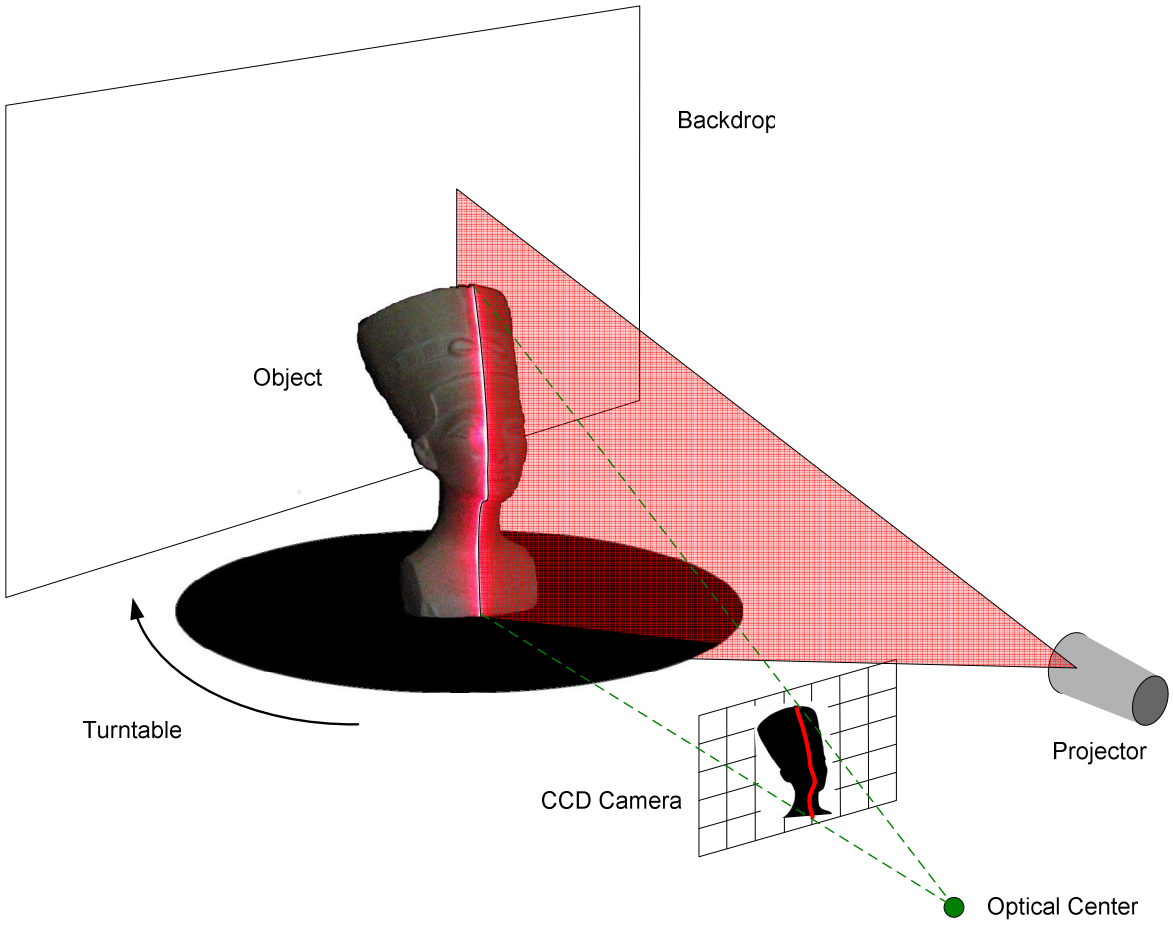

Figure 2.1: Layout of the acquisition system using single-striped laser projectors.

2.2 Calibration

The reconstruction process requires the calibration of the turntable and the laser projectors with respect to the camera. In order to correlate geometrical information from different sources, definitions of the geometries of the camera, the laser projector(s), and the turntable must be known under a common coordinate frame. The transformation from the

camera coordinate frame to the world coordinate frame is explained in the *camera calibration* step (Section 2.2.1). The orientation of the laser planes in camera coordinates is computed in the *laser-camera calibration* step (Section 2.2.2). The problem of estimating camera coordinates with respect to a fixed object space is addressed in the *turntable calibration* step (Section 2.2.3).

2.2.1 Camera Calibration

The objective is to determine a set of camera parameters that describe a mapping between the 3D reference coordinates and 2D image coordinates. These parameters can be split into two categories. The intrinsic parameters, made up of the focal length, principal point, and distortion coefficients, determine the internal geometrical and optical characteristics. The values these parameters take on are independent of the position and orientation of the camera. The extrinsic parameters, which are composed of a translation vector and a rotation matrix for each view, determine the 3D position and orientation of the camera with respect to a reference world coordinate system. In the experiments conducted the distortion parameters are neglected because of the high quality of the camera.

For the calibration process a toolbox for Matlab, by Jean-Yves Bougeut from Caltech University [29], has been used. The calibration is performed by using a special calibration object, which is basically a smooth checker board surface. Utilizing a variation of the Tsai Technique [28], this toolbox handles the calibration in two steps. The initialization step computes a closed-form solution for the calibration parameters excluding any lens distortion parameters. The non-linear optimization step minimizes the total reprojection error (in the least squares sense) over all the calibration parameters (both intrinsic and extrinsic) via an iterative differential optimization technique.

In practice, calibration results allow for a means of converting a 3D point in world space to a 3D point in camera space using the extrinsic parameters and projecting this into screen coordinates with the help of the intrinsic parameters. Conversely, a screen coordinate can also be back projected as a ray into 3D camera space with the application of intrinsic parameters.

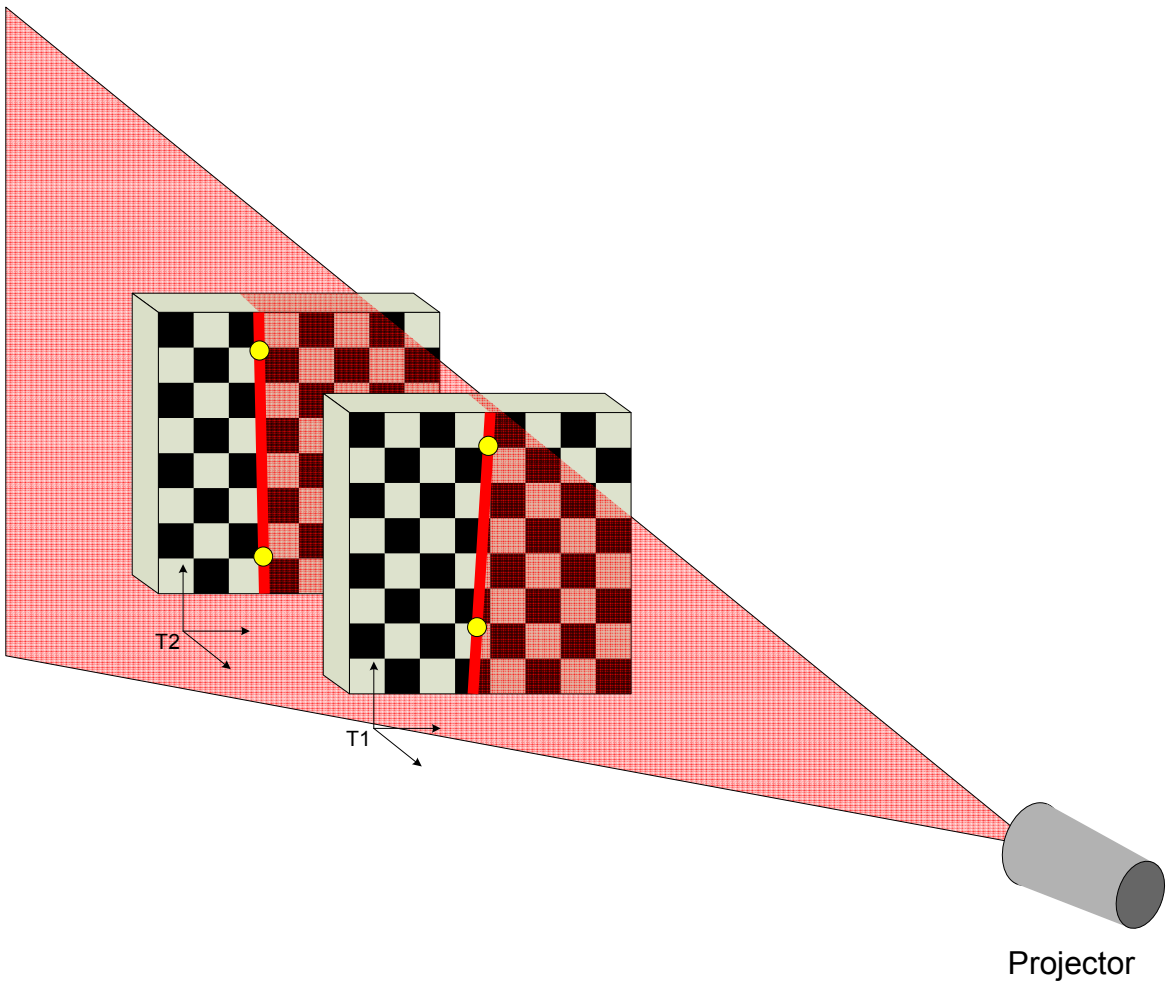

Figure 2.2: Computation of the geometry of a laser plane. Two sample points picked from each laser image are converted to 3D camera coordinates using the transformation matrices made available by the calibration pattern. At least 3 such points are a requisite to reconstruct the laser plane.

2.2.2 Laser-Camera Calibration

The 3D camera coordinates of a range point is identified by computing the intersection of the laser plane with the line that is back-projected from the coordinates of the range points on the 2D camera screens. Therefore, calibration of the laser plane in the camera frame is an essential part of the optical triangulation.

The computation of the laser plane with respect to the camera frame is conducted similarly to the work of Pless and Zhang [40] such that the system observes the laser projected checkerboard calibration pattern in several poses and makes use of the constraints imposed by the relation between the illumination of the laser stripe on the calibration pattern and the plane of the calibration pattern. At first, for the computation of the laser plane with respect to the camera frame, N images of the laser projected calibration pattern are acquired. Each image I_i, $i = 1, 2, ..., N$, contains the calibration pattern placed at a certain different position and orientation with respect to the camera as depicted in Figure 2.2. The calibration pattern is used to compute the extrinsic parameters of a coordinate frame F_i whose z-component is known to be orthogonal to the board. The illumination of the laser beam on the board hence, also lies on the z = 0 plane in the F_i frame. In light of this knowledge, conversion from a screen coordinate to a camera coordinate on each frame is achieved using the following formulae:

$$x_c = \frac{x_s z_c}{f} \tag{2.1}$$

$$y_c = \frac{y_s z_c}{f} \tag{2.2}$$

$$z_c = \frac{\breve{\boldsymbol{R}}_{31}\boldsymbol{t}_x + \breve{\boldsymbol{R}}_{32}\boldsymbol{t}_y + \breve{\boldsymbol{R}}_{33}\boldsymbol{t}_z}{\frac{\breve{\boldsymbol{R}}_{31}x_s}{f} + \frac{\breve{\boldsymbol{R}}_{32}y_s}{f} + \breve{\boldsymbol{R}}_{33}} \tag{2.3}$$

where x_c, y_c, and z_c are the camera coordinates of the screen points x_s, and y_s; f is the focal length, $\breve{\boldsymbol{R}}$ is the inverse of the rotation matrix $\boldsymbol{R}$, $\boldsymbol{t}$ is the translation vector for the particular frame. Equations (2.1) and (2.2) are a direct consequence of the equations deduced from similar triangles of the pinhole model, whereas in Equation (2.3) z_c is simply derived through plugging (2.1) and (2.2) into the coordinate transformation equation:

$$\begin{pmatrix} x_F \\ y_F \\ 0 \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \boldsymbol{R} + \boldsymbol{t} \tag{2.4}$$

where x_F, y_F, and $z_F = 0$ are the coordinates in the particular frame of the calibration pattern.

In order to predict the orientation of a laser source with respect to the 3D camera frame, at least three non-collinear points must be sampled from the planar laser beam. A larger number of samples help refine the prediction and eliminate erroneous samples incurred by noise. In practice, approximately 10 images are used to predict the geometry of a single laser plane and from each image two points are extracted. The geometry of the laser plane is computed by applying a least-squares plane fitting algorithm to the 2N points extracted from N images. This plane delivers the description of the laser plane with respect to the camera coordinate system.

Initially, for the extraction of the laser stripes, input images are converted to HSV (hue, saturation, value) format to distinguish better the red component. The images are then filtered to leave only the illuminated parts of the board visible.

The laser images along with their extrinsic parameters are fed into a program that undergoes the steps listed in Listing 2.1.

For each laser image

- Extract the red component from the image,
- Gather 2D sample points from the laser stripe at regular intervals by computing the sub-pixelic centre of the pulse on the visible stripe,
- Interpolate a 2D line through the sample points collected,
- Choose two separate points from the fitted line and compute their coordinates in 3D camera frame,

Fit a least-squares plane on to the set of points defined in the camera frame.

Listing 2.1: Computation of the laser plane geometry.

2.2.3 Turntable Calibration

Although the camera is fixed while the turntable with the object is rotating, it can equally be assumed that the camera is rotating around the fixed object. Knowing each of the camera positions is necessary if the data acquired from different angles are to be associated to assemble a full model. To achieve this, the turntable's rotation axis is computed on a common coordinate frame, allowing for regular sampling of the position of the camera around the object.

The calibration pattern is placed on the turntable facing the camera and a succession of images of the calibration pattern is acquired while the turntable is rotated approximately 10 degrees for each new image. The camera settings are kept the same to maintain the matching camera optical centre. This means the intrinsic parameters are unchanged. The extrinsic parameters are computed providing a rotation matrix and a translation vector for each view of the calibration pattern. These transformations map to the same point with respect to the rotation axis. The realization that points on the rotation axis remain stationary when the turntable is rotated is the key to solving the geometry of the rotation axis. For precise details refer to [1].

The estimation of the rotation axis produces two parameters: the rotation axis direction $\boldsymbol{u}$ and the position of a point on the rotation axis $\boldsymbol{T'}$. The object frame in which the model is reconstructed is aligned with $\boldsymbol{u}$ and centered at $\boldsymbol{T'}$. These variables provided, the rigid transformation from the object frame to the camera frame at any desired angle can be easily computed, since the exact degree of rotation around the turntable is known for each image [2].

2.3. Silhouette Extraction

The Shape from Silhouette method estimates the 3D model of the object from the calibrated images of the silhouettes. Accurate extraction of the silhouettes is crucial for the quality of the reconstructed object. There exist two major techniques for extracting objects from images. In the first scheme the background, which needs to be uniform in colour, is differentiated from the object by picking the pixels with a certain colour. The second scheme requires an additional image of the scene without the object in it. Each acquired image of the object taken from different viewpoints is subtracted from the image of the scene without the object. A threshold is applied to the resultant differences to obtain the

object silhouettes. Depending on the uniformity of lighting, object's shape, material and texture, and quality of the camera used these two methods may sometimes confuse the background with the object. The first problem is radiosity related where the edges of the object may tint to the colour of the background. Second, the object texture may happen to be seen the same colour as the background due to lighting conditions or colour similarity.

Our object extraction method, which is adopted from the work of Yemez and Schmitt [2], depends on the use of a sharp contrast that must be maintained between the background and the object for precise results. In this scheme the backstage is saturated with light while the object is left in the dark creating a natural silhouette of the object. This scheme works very well for objects with low reflectance properties. However, in certain circumstances, depending on the reflectance of the reconstructed object, the background and the object may become indistinguishable (see Figure 2.3). To avoid these problems and 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 shiny surfaces as explained, in general produces very clean and accurate results and the radiosity and colour confusion problems are circumvented (see Figure 2.4).

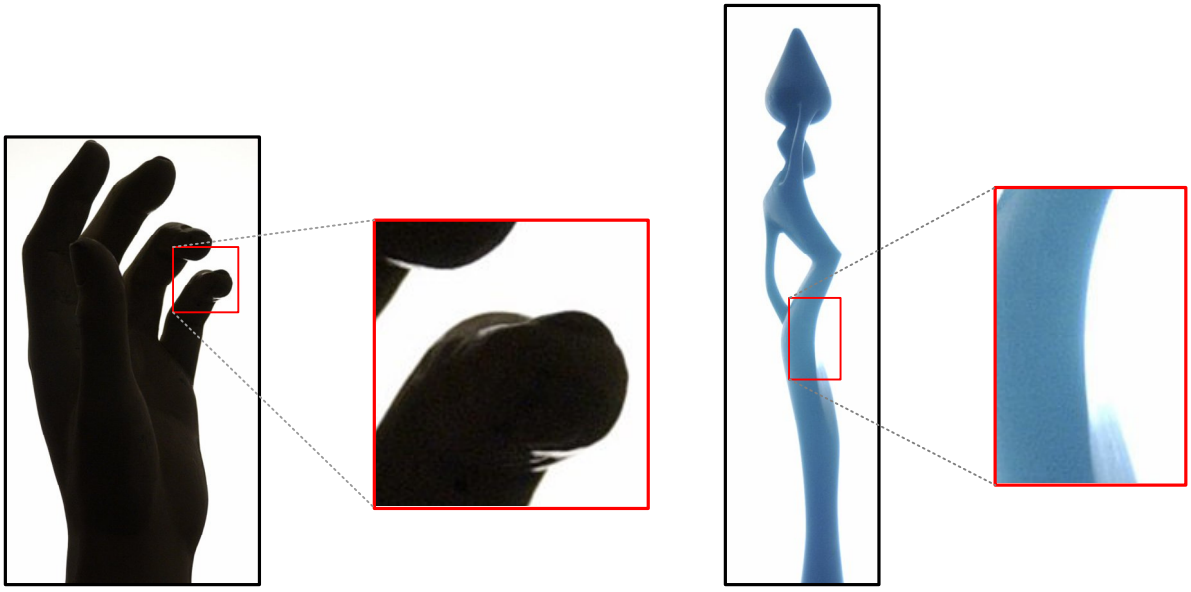

Figure 2.3: Images of two objects with saturated backgrounds. The zoomed sections reveal the difficulty in discerning the object from the background.

The silhouettes are extracted with the use of some filters that will distinguish the object from the background. The applied filters utilize thresholding on the red, green, and blue components of the image. The extraction is not automated in the sense that the filters used in the extraction process are reconfigured for each new acquisition. In order to save up in space the extracted silhouettes are stored in binary form to be used in the Shape from Silhouette method.

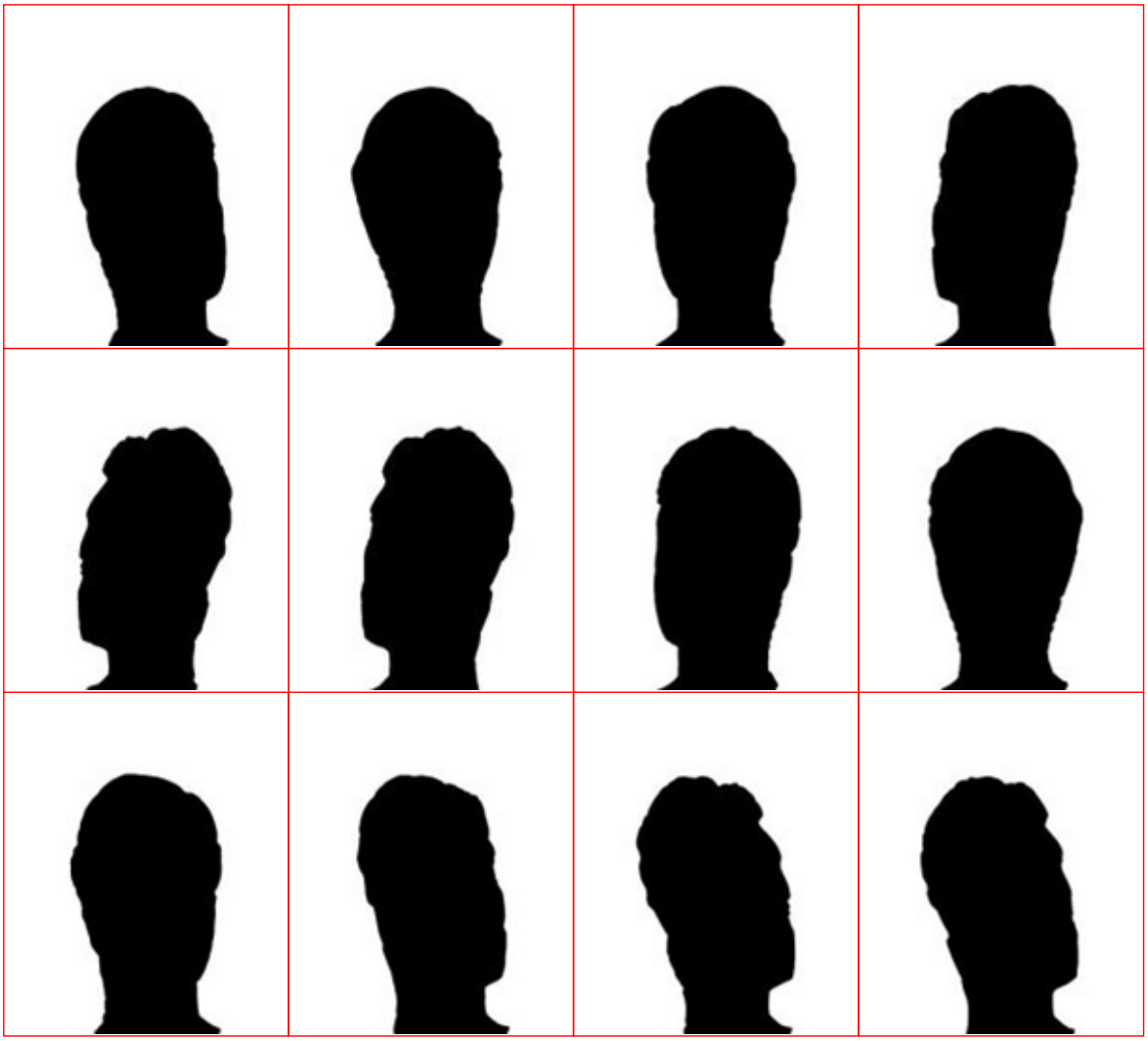

Figure 2.4: A sequence of extracted silhouettes.

Chapter 3

SHAPE FROM SILHOUETTE

3.1. Introduction

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 parameter for an image plane, it is possible to back-project the 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 finite set of image planes from different views provides an approximation to the 3D *visual hull* [37], a superset of the object shape. In this work, the computation of the convex hull is achieved by volume carving, which entails the need to address how the volume is to be represented efficiently. A typical voxel raster contains large homogenously empty regions, and by avoiding allocating space for large chunks of these regions, it is possible to reduce memory requirements substantially. The octree does just that and makes it feasible to work in very high resolution representations, making it the ideal candidate for this task.

The octree is a well-known structure used in a wide spectrum of graphics applications. [6], [7], [36]. Although the basics are similar, every application implements the structure in a specialized manner suited to its needs. Thus, a brief explanation to what the structure of the octree is composed of and how it is constructed and manipulated is provided. Following this, volume triangulation, which is necessary to obtain a final surface model, is described.

3.2. Octree Representation

An octree is a hierarchical tree structure that can be used to represent volumetric data in terms of cubes of varying sizes. These cubes may be in one of three states: either totally inside the object (IN); totally outside the object (OUT); or intersecting the surface of the object (ON). The surface of the object is assumed to be represented by an implicit function $f(x, y, z) = 0$ which determines whether a given cube is IN, ON, or OUT.

A single node in the octree represents a cube with a specific size and location in the world space (nodes and cubes are often used analogously in this report). The structure is constructed starting with a root node (representing the bounding box) at level 0. Subdividing a node, results in 8 child nodes that are equal in size. The children are typically one level higher than their immediate parent. The octree space can be represented as a 3D grid of 2^R x 2^R x 2^R unit cubes, where R denotes the highest resolution level of the octree. The centre of the octree is chosen to coincide with the centre of the world coordinate system and the edges of cubes are aligned with the axes. Each cube in the structure is represented by its level and its indices on the X, Y, and Z axes. Only the root node has the indices (0, 0, 0). Figure 3.1 depicts the simple indexing scheme in 2D.

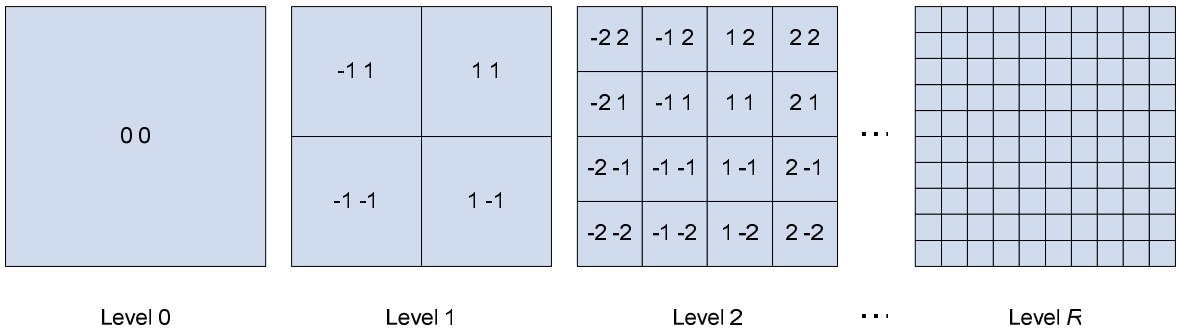

Figure 3.1: Indexing scheme for the octree nodes in 2D. The bounding box of the world space is at level 0. A node is identified by the level and the indices it is assigned.

The octree representation is obtained by recursively subdividing each parent cube, starting from the root node. IN or OUT cubes need not be further subdivided since they do not comprise any surface information. The recursive subdivision continues only for ON cubes, until the unit cubes corresponding to leaf nodes of the highest octree level are reached.

3.3. Octree Construction

When the camera geometry is known with respect to the object coordinate frame, using the pinhole model of the camera a bounding silhouette cone can be constructed by back-projecting the silhouettes from every viewpoint (Figure 3.2). The intersection of these silhouette cones provides a close approximation to the object shape. As more images contribute to the overall intersection, the resultant shape approaches the visual hull. The visual hull can be described as the largest geometrical entity that produces the same silhouettes as the object itself observed from any possible viewpoints outside the convex hull [38]. Similar to a convex hull, the visual hull is always larger than the original object, and no part of the object is outside of the visual hull. What is lacking in the visual hull, is the representation of hidden concavities self obstructed by the object.

A given cube represented by a node in the octree can be either inside, outside, or on the boundary of the visual hull. When constructing the octree the state of the traversed nodes must be determined and for this purpose it is appropriate to first explain how a single point's position relative to the visual hull is computed. In order to find out whether an arbitrary 3D point in the object space is inside or outside the visual hull, using the calibration parameters the point is projected to each of the available 2D images. If any one of the projections lies outside of the corresponding silhouette then this point is declared to be outside the visual hull (point is said to be OUT). Otherwise the visual hull is considered

to cover the point within its boundaries (point is said to be IN). On a 2D image, the estimation of the location of a subpixelic point with respect to the object silhouette is provided by the isolevel function, which is defined in Section 3.4.2.

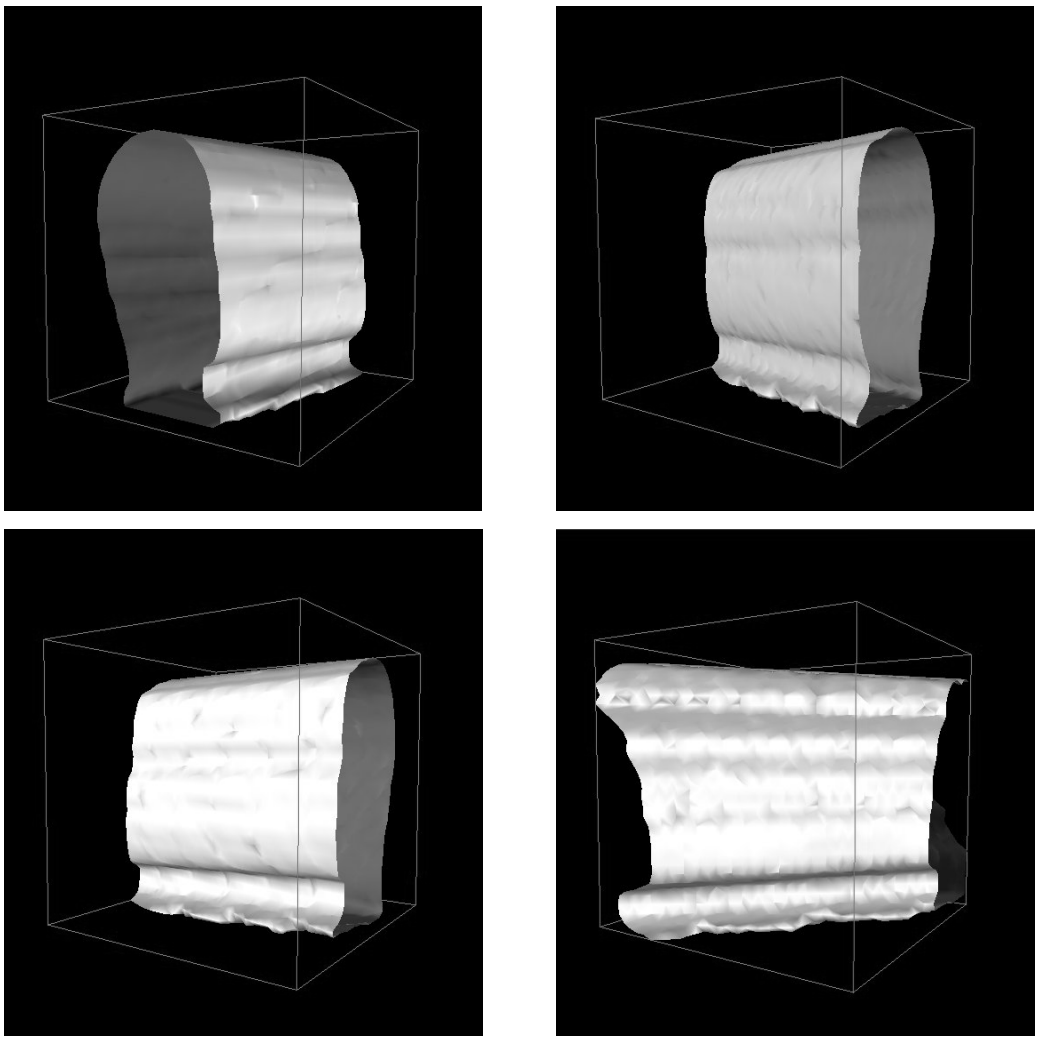

Figure 3.2: Bounding silhouette cones from four separate viewpoints displayed within the limits of the bounding box of the object. The intersection of all silhouette cones defines an approximation to the visual hull.

The initial prediction of the state of an octree node is made on the basis of the states of the 8 corners of the represented cube. If all the corners are found to be inside the visual hull then the node is temporarily marked as IN. Similarly, if all the corners fall outside of the visual hull, then the node is temporarily marked as OUT. Cubes with some corners inside and some corners outside the visual hull, can safely be set to ON and freed for further subdivision (unless it is a leaf node). If the node is marked as ON initially or the node of interest is a leaf node, then no further inspections need to be made to determine the state of the node. The current state of the node will be final.

If the examined node is not a leaf node and its state is initially marked as IN or OUT, then more points need to be inspected to establish the true state of the node, since the corners alone do not ascertain the assumed state of the node as illustrated in Figure 3.3. The visual hull may pass through some part of the cube without covering any of the corners or conversely the visual hull may cover all the corners but not the whole cube. To detect these cases, the states of the points sampled along each face of an IN or OUT cube must be crosschecked with the current state of the cube. If a point which is outside the visual hull is encountered along the faces of an IN cube, then the state of that cube is modified to ON and thus freed for further subdivision. The same procedure is carried out for OUT cubes with points on the faces of the cube that are actually inside the visual hull.

In order to preserve the object detail at the highest level R, the discrete surface points along the faces of a cube must be sampled at level R. That is, on each face of a cube of level *r*, a sum of $\left(2^{R-r}+2\right)^2$ points must be tested (including the corners). This adds up to $\left(2^{R-r}+2\right)^2\times 6-\left(2^{R-r}+2\right)\times 12$ points on the whole surface of the cube. This test can be cut short when a point that falsifies the current state of the node is encountered. That is, if the first guess of the state of a given node is OUT and a point on the surface of that node is found to be inside the object hull then the node is marked as ON and freed for further

subdivision and vice versa. This way, the exhaustive procedure is substantially shortened. If no contradicting points are found then the state of the node is left unchanged and no further inspection is required for this node and its children since the object surface is known not to pass through it.

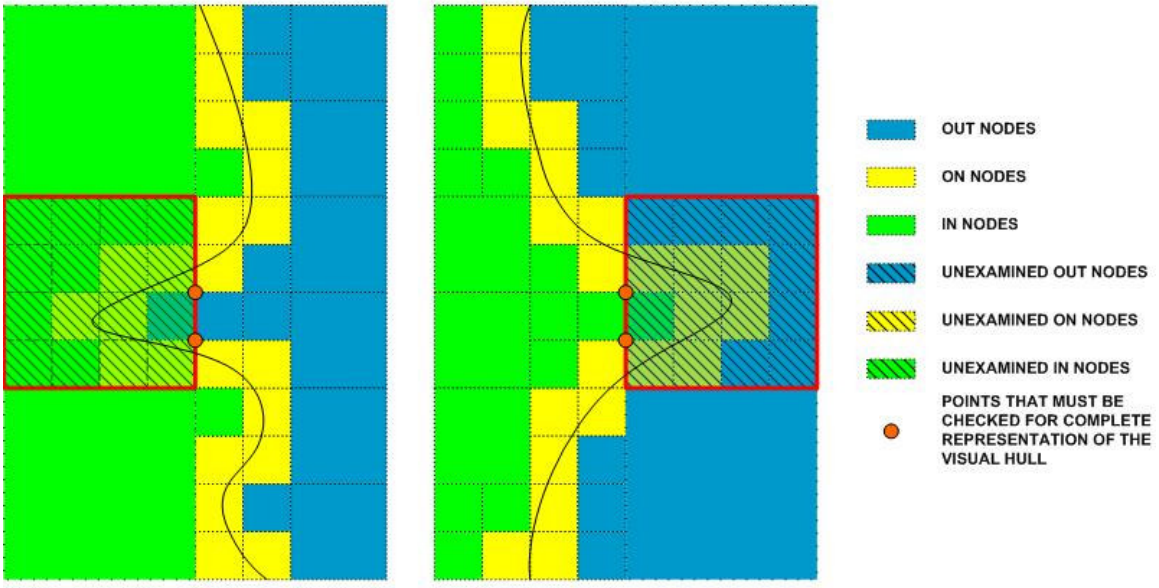

Figure 3.3: Sketch of two possible configurations of the object surface. The red highlighted squares outline the nodes whose corners are either completely IN or completely OUT. The sides of these nodes need further investigation to reveal their true states. Those points that hint at their true nature are indicated with circles. Without testing these points, the nodes highlighted in red on the left and on the right depictions would erroneously remain IN and OUT respectively and these sections of the object would appear as holes in the final reconstruction.

3.4. Surface Triangulation

3.4.1. Surface Representation

The surface of a 3D model may be represented in several ways such as voxels, particles, polygons, and more complicated primitives like splines. A voxel is the smallest distinguishable spatial element of a 3D image arranged in a regular grid and a particle is a primitive defined by its orientation and position arranged in an irregular manner. Voxels and particles present problems when zooming in on the object (exhibit cubes and holes) and perform slowly when rendering. Triangles or higher degree parametric surfaces, on the other hand, are true surface representations but require the extra work of extraction from volumetric data. In this work, the silhouette 3D models are represented with the triangle mesh, which basically is a set of connected triangles.

To extract a triangle mesh from the volumetric model, the octree representation is processed using the marching cubes algorithm [25]. The algorithm proceeds through voxels (cubes) in the object space, and determines the polygons (triangles in this case) that must be created to represent the part of the isosurface that passes through each voxel. This is accomplished by creating an index to a precalculated array of 256 (2^8) possible polygon configurations within the cube, by taking into account the IN/OUT states of the 8 corners. The precalculated polygon configurations are recorded in a look-up table of 256 entries and this table is used to find efficiently the correct orientation of the isosurface passing through each cube. The precalculated array of 256 cube configurations can be obtained from 15 unique cases, by reflections and symmetrical rotations.

One problem with the marching cubes is that some cube configurations may be interpreted and triangulated in more than one way. The look-up table employed in the

experiments deals with such ambiguous cases and preserves the correct topology of the object shape [24].

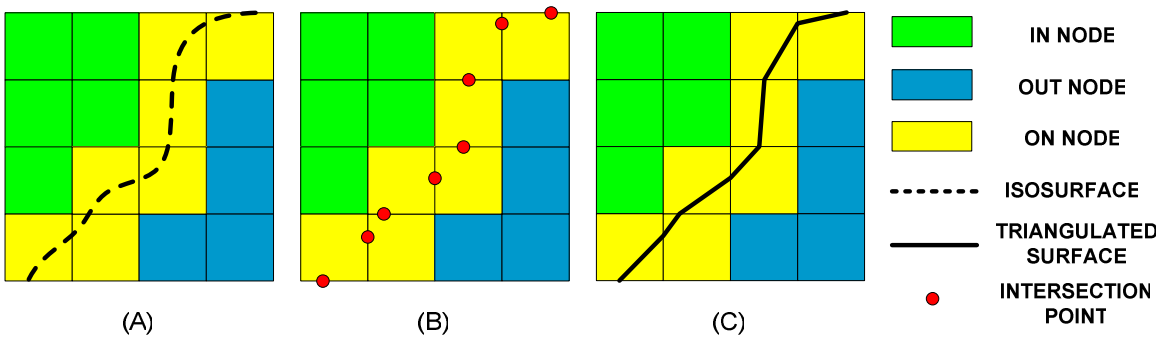

Figure 3.4: A sample grid of object space depicting the triangulation procedure. (A) The isosurface passing through the voxel grid. (B) Interpolation of the vertices at the intersection of the isosurface with the edges of the ON cubes. (C) The triangles are obtained by connecting the vertices.

Each vertex of the generated polygons is placed on the appropriate position along the cube's edge where the isosurface passes (Figure 3.4). The computation of where the isosurface cuts the cube edges is explained in Section 3.4.2. Once the triangulation is carried out for each voxel, the patches from all cubes that are on the isosurface are connected to produce a final closed surface model of the object (see Figure 3.5).

.

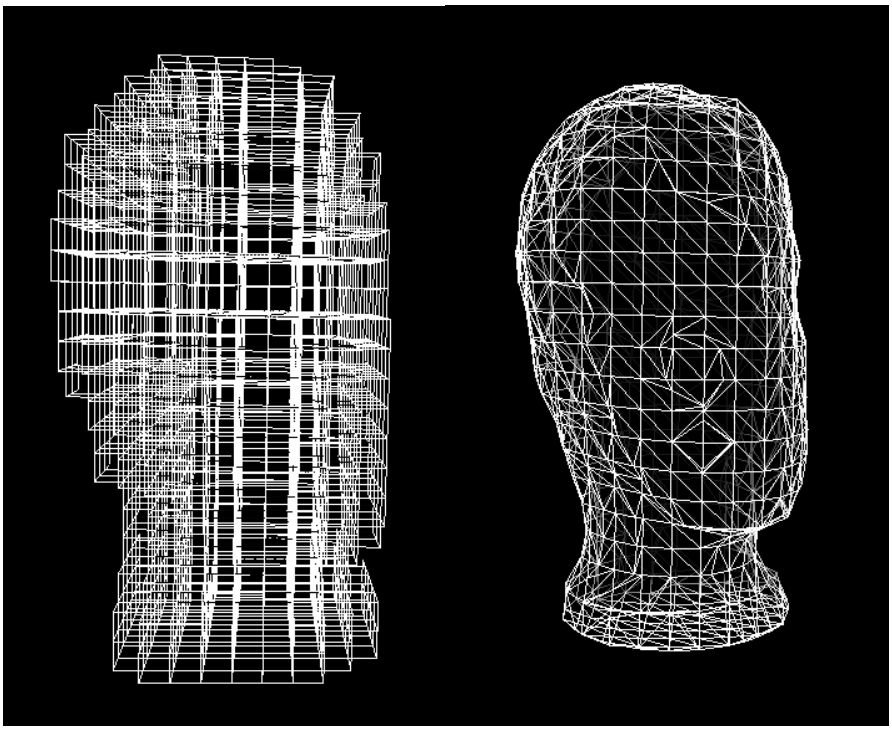

Figure 3.5: (Left) ON cubes of a sample octree. (Right) Triangulated display of the same octree.

3.4.2. Definition of the Isolevel Function

The object surface is assumed to be represented by an implicit surface $f(x, y, z) = 0$. The isolevel function $f(x, y, z)$, which determines the measure at which a given point is inside or outside the object, is computed using the sequence of binary silhouettes. A point in the object space can be projected to the image planes using camera calibration parameters. The sub-pixelic projection (u, v) on the discreet raster is bilinearly interpolated to a value G between 0 and 1:

$$G(u,v) = (1-\alpha)((1-\beta)I(\lfloor u \rfloor,\lfloor v \rfloor) + \beta I(\lfloor u \rfloor,\lfloor v \rfloor+1)) \\ +\alpha((1-\beta)I(\lfloor u \rfloor+1,\lfloor v \rfloor) + \beta I(\lfloor u \rfloor+1,\lfloor v \rfloor+1)) \quad (1)$$

where $(\lfloor u \rfloor,\lfloor v \rfloor)$ denotes the integer part and (α,β) is the fractional part of the coordinate (u, v) in the binary discreet image $I(m, n)$. To produce a smoother transition across the object boundary on a single binary image, the interpolation is thresholded with a value $\varepsilon \in (0,1)$. Using all the available binary silhouette images, the isovalue at a given point (x, y, z) in the object space is computed as follows:

$$f(x, y, z) = \min_n \left\{ G\left[\text{Proj}_{I_n}(x, y, z) \right] - \varepsilon \right\} \quad (2)$$

where Proj_{I_n} is the projection of the point (x, y, z) to I_n, the n'th binary image in the sequence. The function $f(x, y, z)$ takes on values between $-\varepsilon$ and $1-\varepsilon$, and the zero crossing of this function reveals the isosurface. As defined in (2), the isovalue of a 3D point (x, y, z) is provided by the image of the silhouette that is farthest away from the point, or in other words, where the 2D isovalue $G-\varepsilon$ assumes its minimum value. This scheme is consistent with the rule that a point is marked OUT whenever a single projection that is outside the corresponding silhouette is found.

The ε value gives control over how much the isosurface contracts or expands from the actual shape of the object [2]. In practice ε is kept at 0.5 to preserve the object's proper shape.

During the triangulation of the object model, the marching cubes needs to determine where the isosurface cuts the cube edges to discern where to place the vertices of the extracted polygons. The isosurface only cuts those edges that connect one IN and one OUT corners. Along such an edge that intersects with the isosurface a dichotomic subdivision is

carried out to search for the point where the isosurface passes and the isolevel function f is close to 0. A sufficiently small threshold value T, where $-T < f(x, y, z) < T$, is used to determine how close the isolevel of a given point must be to pass it as being on the isosurface (see Figure 3.5).

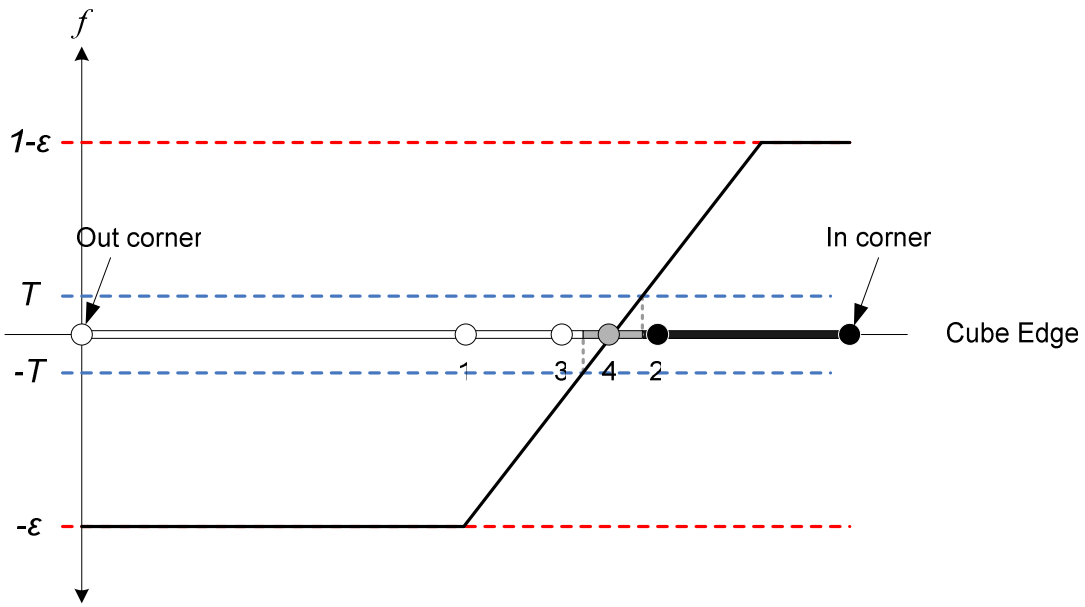

Figure 3.6. Locating the isosurface along the edge of a cube. One corner of the edge is OUT (assigned a negative isovalue) and the other corner is IN (assigned a positive isovalue). The isosurface is assumed to pass where the isolevel function f takes on a value bounded by T and $-T$. For this example, the binary search takes 4 probes to home in on the target region of the cube edge.

3.5. Summary

In this chapter, a scheme for creating 3D models from silhouettes was explained. First, the object is constructed volumetrically using the octree structure and later surface

information is extracted from the volume in the form of a triangle mesh. The obtained results can be delivered in a raw polygonized format or the triangulation can be embedded into the octree structure for further volume carving. The choice depends on the nature of the proposed fusion technique explained in a subsequent stage.

Chapter 4

SHAPE FROM STRUCTURED LIGHT (OPTICAL TRIANGULATION)

Shape from Structured Light 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. Some systems rotate the object to obtain a cylindrical scan, and others may sweep the object across the field of view. A depth profile is computed from the sample points gathered from the illuminated parts of the object, creating a cloud of range points. By connecting the nearest neighbouring range points with triangles, a range surface of the scanned object is generated.

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 [10]. This main flaw in range scanning systems is the motivation behind the attempts to fuse optical triangulation with other techniques.

4.1. Considerations for the System Setup

Structured light range scanners come in many shapes and sizes. In the context of this work, only scanners built for small scale acquisitions, which permit the scanning of objects under 1 meter, are of interest, as opposed to larger scale scanners that are used in recovering the shape of whole rooms, buildings or large museum artifacts as illustrated in the Digital Michelangelo Project [12].

The structure of the scanner setup used, directly affects the acquisition process and ultimately the quality of the reconstruction. Therefore, it is important to distinguish the different types of available structured light range scanners. The simplest and most common of all the structured light systems is the single-camera, single-striped 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. Additionally, some systems incorporate translational movement either to the scanner or the object, as elaborated in [10]. A single sweep produces parallel stripes of sampled range points from which a patch of the object surface is triangulated. However, this system introduces the added computation of translational calibration and the problem of aligning overlapping parts of

the multiple patches obtained from different views. Figure 4.1 illustrates range samples obtained from rotational and translational sweeps. Full range images produce separate patches of the object. The alignment of multiple patches that are scanned separately is another important issue that must be addressed with these systems.

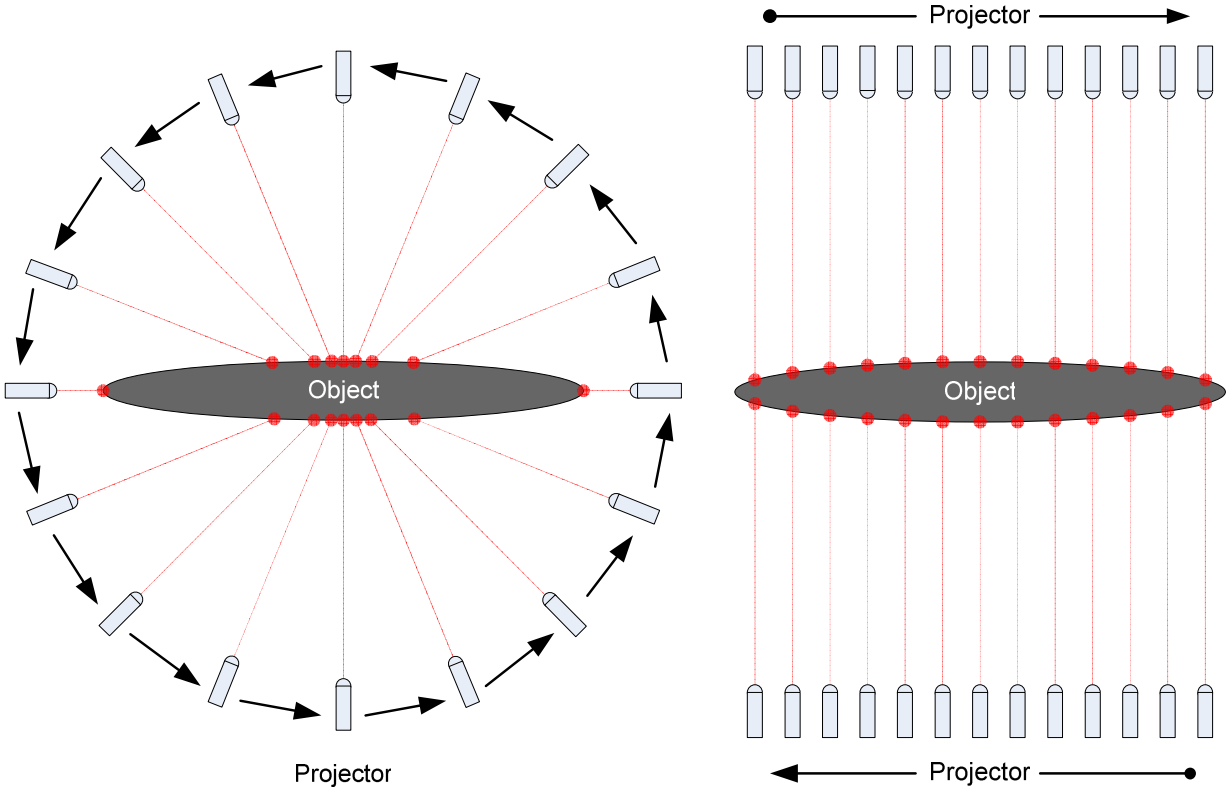

Figure 4.1: Top view of a slice from an ellipsoidal object with two separate scanner configurations. (Left) A rotational single-stripe projector takes samples from the object surface. The samples happen to be denser towards the centre of the object. This irregularity in the distribution causes some parts of the object to be under-sampled for the extraction of proper surface information. Even if the object surface is extracted and the obtained samples are enriched by subdivision, it is easy to miss voxels that should be carved in volume carving applications. (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.

Systems using multiple-stripe projectors bear the requirement of distinguishing the stripes from one another. They can be categorized into three classes in the way the stripes are recognized. The first class, assumes surface continuity allowing adjacent stripes to be viewed in the same adjacent order, but at the same time restricting the object to a certain shape [16]. The second class differentiates stripes based on colour but works only for objects without texture [17]. In the third class, stripes can be coded by varying their illumination over time, which may require several frames to compute depth and introduces the problem of finding boundaries between stripes of the same illumination [11]. Depending on the system structure, these methods may also produce multiple patches of the object surface which require aligning in the object space.

Using multiple cameras, some systems allow manual sweeping of a single projector over the object by finding corresponding points on the different views of the stripe using epipolar constraints [15]. This scheme greatly reduces the calibration complexity while increasing the chance to focus on those parts of the object that are suspected to produce holes.

The current system setup fits into the single-camera, single-striped scanner category, supporting only a rotational movement of the object. Two laser projectors are used in the setup, which are run separately in order to avoid confusion between their projected beams. It is possible to make multiple scans with the same projector positioned differently, to cover even more of the object surface. This setup is simple and commercially available, and alignment of patches is not required. However, as the obtained results, shown in Chapter 6, indicate the deficiency in the amount of range data needed for a successful fused reconstruction, other methods delivering full range images may become more favourable, despite the fact that they are more sophisticated and entail more effort.

4.2. Overview

The input to this process is a series of laser images of the object on a full rotation around the turntable. The camera's intrinsic parameters, extrinsic parameters for each image, and the description of the laser plane with respect to the camera must be known.

The routine starts with processing the laser images to draw 2D sample points from the laser strips. The position of these points in the object frame is then computed and triangulation is carried out on the obtained point cloud to produce normal values for range points. Finally, subdivision and smoothing is applied to the generated mesh to enhance the quality of the results.

4.3. Processing the Laser Images

Laser images are analyzed in HSV format to extract the illuminated pixels. Pixels not exceeding a certain intensity threshold are eliminated. The right choice for the applied threshold is crucial for eliminating the noise inherent to range scanner systems. Low values chosen for the threshold cause proliferation of noisy data and deformations on the final shape of the reconstruction. Too high values for the threshold, on the other hand, reduce the number of useful and valuable range points that can be extracted. The applied filters may become ineffective in extracting the stripe's peak correctly. Figure 4.2 illustrates some ambiguous cases that are difficult to settle.

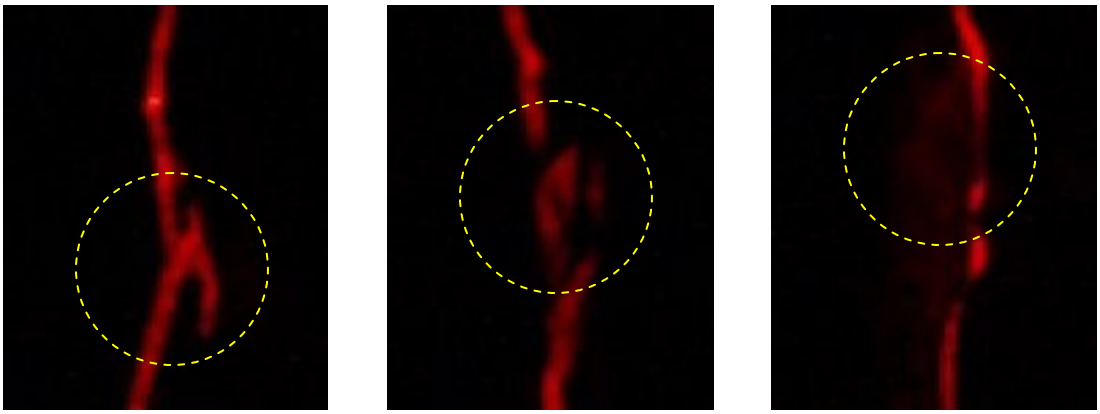

Figure 4.2: Samples of laser stripes that are troublesome in discerning where the peak of the pulse lies. Possible causes are (left) splitting of the laser stripe, (middle) scattering, and (right) reflectance off the object surface.

In high resolution images the width of the laser stripe covers several pixels. The laser plane is assumed to pass through the peak of the depicted laser stripe. Among the possible ways for locating the pulse is determining the mean, median, weighted average of, or fitting a quadratic parabola to the consecutive illuminated pixels across the width of the stripe. Computing the weighted average and fitting parabolas are the most common and accurate methods and generate similar results as witnessed in the relevant literature. Currently, both methods are used in the implementation.

In order further to refine the extraction, no pulse is sought on streams of consecutive pixels that amount to less than a certain threshold. This way, isolated pixels resulting from reflections, scattering or other phenomena are neglected. When a valid stream of pixels is spotted, the sub-pixelic peak point where the laser pulse culminates is computed and recorded. This is performed at regular intervals along the laser stripe for each image.

4.4. Derivation of Range Points

The 3D positions of the scanned points on the object are determined from the intersection between the rays protruding from the camera and the laser plane. The pseudo-code in Listing 4.1 summarizes how the 2D sub-pixelic points from N images are converted to world coordinates.

For each image I_n of the sequence, $n = 1, 2, ..., N$

  For each 2D point P extracted from I_n

    Back-project P to camera coordinate frame, using the intrinsic parameters of the camera; this will define a ray.

    Find the intersection of the laser plane and the back-projected line to find the position of the range point in camera coordinates.

    Retrieve the rotation and translation variables associated with image I_n and apply these extrinsic parameters to find the position of the range point in world coordinates.

    Collect the range point in the strip $\boldsymbol{H}_n$.

Listing 4.1: Conversion from screen coordinates to world coordinates

This simple procedure yields a set of strips $\boldsymbol{H}_n$, where $n = 1, 2, ..., N$. Each strip contains the set of range points collected from its corresponding image. Range points are sorted into strips to support a simple triangulation method.

4.5. Triangulation

The fusion technique that will be presented in Chapter 5 does not require the connectivity information of a polygon mesh generated from the range points. Triangulation of range points serves only as a rough means to estimate a surface normal for each range point. Therefore, an elaborate triangulation in order to obtain good visualization for the model is avoided in this stage.

Connecting nearest neighbours with triangles is a common strategy for systems that produce lattices of regular samples, such as [10], [11]. The triangulation that we have adopted undertakes a similar path by weaving a web of faces across range points of adjacent strips. This method, though simple, incurs some problems for the current acquisition system that produces intersecting laser planes. Our scanner setup causes irregular sampling and convergence towards the intersection of the planes. Depending on the object shape, this can have detrimental effects on the outcome such that range points that are located too closely produce very jagged surfaces and sparsely distributed points cause inevitable gaps.

When the projector is rotated with respect to the object, the projected laser planes intersect in space. Range points located before and after these intersections must be triangulated inversely to preserve coherent shading and surface orientation (see Figure 4.3). Again, other triangulation systems discussed previously, perform better in this respect since they emit parallel stripes or stripes that intersect at the projector.

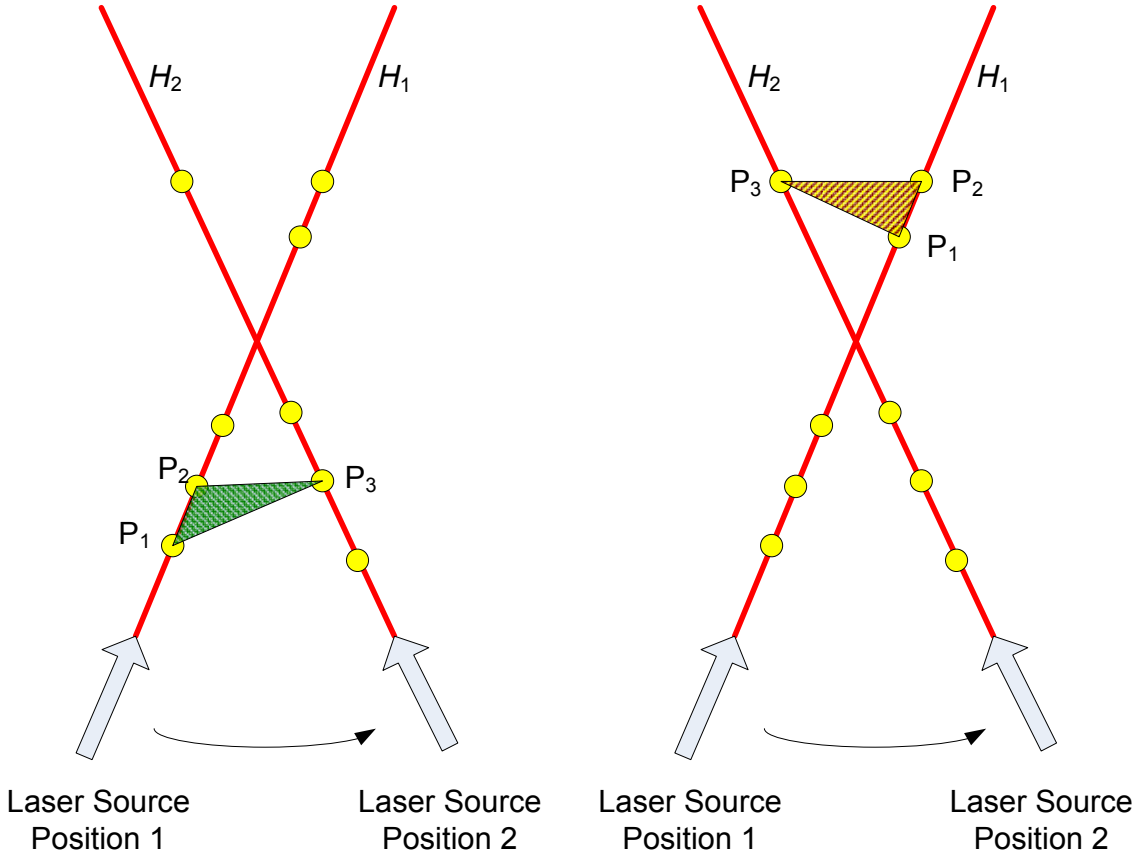

Figure 4.3: Range points are sampled from two consecutive laser stripes $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$. Samples, depicted in circles, are used to polygonize the object surface. The vertices of the depicted triangles on the left and right must be sequenced in opposite order so that their normal vectors are computed correctly and consistent shading is maintained. If the triangle on the left is triangulated in the order P_1, P_2, P_3 then the one on the right must be ordered as P_1, P_3, P_2. Substituting P_1 and P_2 in the equation of $\boldsymbol{H}_2$ and P_3 in the equation of $\boldsymbol{H}_1$ reveals which side these points lie with respect to the intersection.

In order to produce more complete models of the object, multiple scans may be conducted, which also causes most visible parts of the object to be covered multiple times by different scannings. With the intention of reducing redundant data originating from

multiple definitions of the same parts of the object from separate scans, the obtained range points can be volumetrically sorted into grid cells and then merged. However, our aim is not to obtain a single surface definition from the optical triangulation, but to provide range points for volume carving in the fusion phase. As observed in the obtained results presented in Chapter 6, supporting a denser set of range points allows volume carving to be performed at higher resolutions, which produces results in greater detail. Therefore, surface definitions collected from different scans are not merged into a single surface definition. Data drawn from the separate optical scannings are to be incorporated separately to the final model.

4.6. Subdivision

The faces produced in the triangulation of the range points are subdivided to produce new points that can be used in the fusion. This enrichment in range data is carried out to support the high levels of octree structures created in the shape from silhouette method. In practice, since the number of faces grows very fast with subdivision, more than one round of subdivision produces redundant and burdensome results.

The methodology adopted for this application produces one new vertex and three new faces per old face. The new vertex is positioned at the centroid of the old face. The camera position is computed by averaging the camera positions associated with the old vertices. More elaborate methods may also be implemented such as Loop [19] or Butterfly [31], which produce four new faces for each old one. The current scheme is simpler to implement and serves its purpose adequately, which, as stated before, is to achieve a certain density of range points when carving the octree space. However, in the future, Loop's refining process may be employed together with the smoothing process as explained in his work.

4.7. Smoothing

The occurrence of noise incurred errors in range scanner systems is a prominent problem. Smoothing the resultant triangular mesh aims to alleviate this shortcoming to a degree. The fairing algorithm of Taubin [18], which is applied to the optical triangulation results in this work, provides a comprehensive solution to smoothing large polyhedral surfaces of arbitrary topology. Taubin, basically, computes the new position of each vertex as a weighted average over its neighbourhood and applies a low-pass filter to prevent shrinking. The weights are chosen as the length of the edge between the vertex being repositioned and its individual neighbour. One of the extensions of Taubin to the work conducted by Loop was that different masks were applied to boundary and non-boundary vertices. The first level vertex neighbourhood of a boundary vertex comprises only boundary vertices whereas the neighbourhood of a non-boundary vertex carries all the neighbouring vertices. This prevents the prevalent holes in the range triangulation from expanding, as smoothing is carried out.

4.8. Summary

This chapter covered the issues related to a simple scheme for generating optical triangulations. Reading the laser images, creating sets of range points, extracting surface information, and finally, application of subdivision and smoothing algorithms were explained. For each conducted scan, the output of the optical triangulation is composed of a set of smoothed faces, which is eventually discarded, and a set of vertices with associated normals and camera positions. The normals are needed in the final triangulation of the fused volumetric reconstruction while the camera positions are used in volume carving.

Chapter 5

FUSION

5.1 Overview

Shape from Silhouette method defines a volumetric model from which a surface representation can be derived. Shape from Structured Light method produces point clouds that can be triangulated to obtain a surface model. As stated before the silhouette model lacks the concavity information but is robust. The range data are accurate but incomplete mostly due to camera and light occlusions and partly due to the nature of the scanner used.

The problem of fusing Shape from Silhouette and Shape from Structured Light information can be approached in several ways. The models for each method can be constructed separately and the results can be merged after converting one model representation to the other. This can be achieved by a process of refining one model with what is inferred from the other. The silhouette model that assumes a solid and sound framework can be carved away volumetrically at the hidden hollow sections by the more precise cloud of range points to be followed by a triangulation process. Implementations for volumetric stereo and silhouette fusion schemes have previously been examined in [20]. Volumetric model reconstruction using range information is investigated also in [10].

Employment of model deformation frameworks is common for systems containing different kinds of information about the surface of the reconstructed object [39]. With the use of triangulated surface models derived from silhouettes, a deformation scheme can be adopted, where the range points reshape the object surface defined by the silhouettes. Two

different techniques can be used under this scheme: a classical snake approach [22] or a level-set approach [23]. As has been pointed out in [21], using a classical snake approach constrains the initial topology of the object to be constant, which implies that the topology has to be completely recovered (by the silhouettes) before snake evolution can be carried out. Level-set methods [23] have the ability to overcome this problem while constraints applied to the smoothness and the topology of the final surface become difficult to control.

Alternatively, a common 3D representation can be used for both methods to construct a single model from the ground up as in Tosovic et al. [5], in which a volumetric reconstruction is performed by fusing silhouette and range data. Tosovic cross-checks the state of each IN and ON voxel derived from the silhouettes by projecting the coordinates of the voxel to the laser image of the nearest laser plane. The work undertaken is not very clear in how the binarization of the laser images is carried out, which is crucial in understanding how the laser image and the projected coordinates are intersected. Because of the main purpose of the work, which is to compute the volume of archeological vessels, triangulation of the object volume is not carried out. Moreover, the final results are not presented well. Therefore, a proper assessment of this method is difficult to make for now.

In our work, a volumetric fusion method, illustrated in Figure 5.1, is chosen mainly to avoid topological problems that are likely to arise should a surface based fusion be applied. The silhouette model, which ideally encompasses the structured light model, can be carved where the hidden concavities lie and the missing range data can be filled in by the silhouette data to attain a complete and accurate model. Initially, triangulation information for ON nodes will be available. However, as the carving progresses many new voxels that do not contain surface information will emerge. The foremost concern is to categorize these new voxels and to find ways of drawing surface information from them, to achieve a final complete mesh.

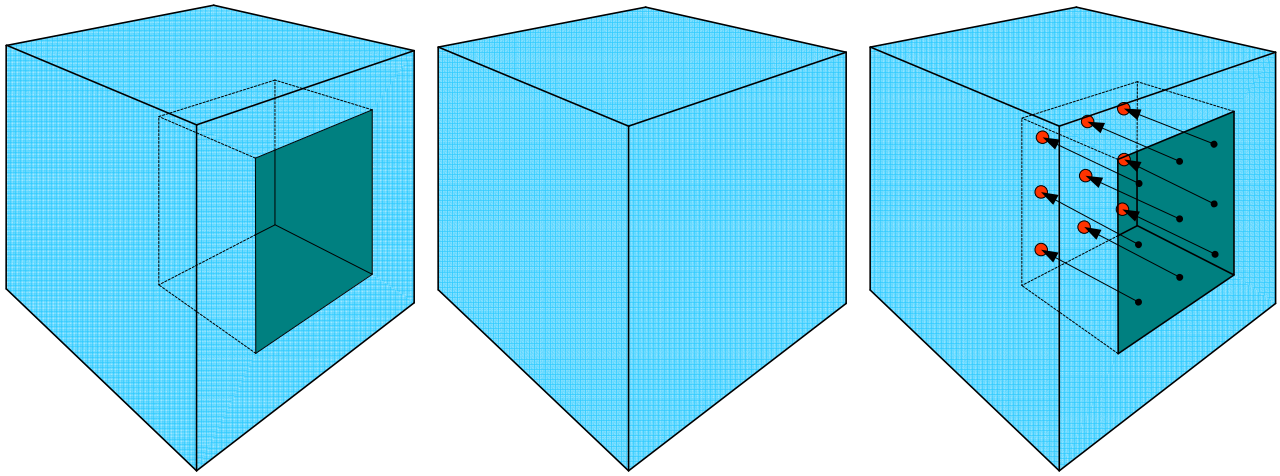

Figure 5.1: Volume carving. (Left) A rectangular box shaped object with a cubic hole on one face. (Middle) Visual hull of the object on the left, which can be obtained from the object's silhouettes. (Right) Range points sampled from the interior surface of the hole, are used to carve away the extraneous volume.

5.2 The Volumetric Fusion Process

Initially, the object space constructed by the intersection of the silhouette cones is composed of voxels that are marked as either inside the object (IN), outside the object (OUT), or on the surface of the object (ON). Since hidden holes may elude silhouettes, this volumetric structure, which is an approximation to the visual hull, contains voxels marked as ON or IN, that are in fact outside of the actual object boundaries. For effective carving, it is essential that these voxels be detected and marked accordingly. Using the cavity sensitive range data, voxels located in between the surface of the visual hull and the surface of the actual object are eliminated (converted to OUT), and those voxels located inside the visual hull that contain range data are changed (converted to ON) so that they represent the new surface of the object. Carving the object space produces holes on the edges of the hollowed parts (IN and OUT voxels sharing surfaces), which must be patched up in order

to obtain a complete model. After the holes are removed, the object volume is triangulated to generate the surface of the object, for which, only ON voxels (either preexisting from the visual hull or newly created during carving) are needed. Finally, the resultant mesh is decimated and faired to obtain more visually pleasing results.

The voxels making up the object space are represented in octree form rather than a run-length grid to conserve memory. At the beginning of the fusion process the available data are:

- An octree representing the convex hull. The state of each node in the octree indicates if the voxel is IN, ON, or OUT. Each ON leaf node contains triangulation information from the visual hull, which constitutes a patch of the surface enclosed by the dimensions of the cube. From each leaf node, representing a voxel in the structure, the states of the corners, the level, the indices, and the centre coordinates of the voxel can be inferred.
- A list of range points. Each range point is linked with its position in world space, a position of the camera's optical centre used to discern the cubes to be carved, and a normal used during triangulation.

5.2.1 Carving the Object Space

5.2.1.1 Extension to the Node States

The carving process changes the description of the octree and introduces new types of ON nodes that require different means of extracting surface information. This differentiation is categorized under 4 types:

- TYPE 1 – Untouched ON nodes. Contain only silhouette data
- TYPE 2 – Nodes that comprise both silhouette and range data
- TYPE 3 – Nodes that comprise only range data

- TYPE 4 – Nodes that have no associated silhouette or range data

5.2.1.2 Carving

Observing the fact that no part of the object volume intervenes the line between a visible range point and its projection on the camera screen is key to carving volume. A range point, $\Re$, and the position of the optical centre of the camera, *O,* at the time the range point is acquired are tied to produce a line segment, *L*, or a *scan line* as it will be referred to from here on. This line segment does not intersect with the correct definition of the object surface. Therefore, removing those voxels from the initial volume that actually do intersect with scan lines will yield the volumetric definition that is sought. The pseudo-code in Listing 5.1 summarizes the routine for carving the octree space using a set of *X* range points.

Initially, each range point is associated with the maximum level octree node (or the voxel) that it resides in the object space. Ideally, an accurate acquisition would produce all the scanned range points to be inside the boundaries of the silhouette based visual hull, since the silhouette model lacks only hidden holes and should cover the whole range data model. In practice, however, due to noise from input images and erroneous calibration parameters, some range points may happen to fall outside the visual hull, i.e. they may lie inside OUT nodes. In such cases the association with the particular range point may or may not be carried out. When the range data fits closely with the visual hull, the outlier range points can be simply overlooked and made obsolete. However, in situations where the noise in the system is too high and the majority of the range points are situated outside the visual hull, the outsiders can be used to compensate for the heavy loss of range data. In such cases it follows that when an OUT node with a range point is encountered it is converted to ON TYPE 3, if that OUT node is adjacent to an original ON TYPE 1 node from the initial visual hull.

Furthermore, if the outlier range points are not taken into account, merging highly uncorrelated data from the two models may create jagged surfaces as depicted in Figure 5.2. The pits on the surface of the fused model appear as a result of the discontinuous sampling of the range data. The conducted tests reveal that a high percentage of the acquisitions result in uncorrelated outputs from Shape from Structured Light and Shape from Silhouettes, and thus it is favourable to let the range points lying outside of the visual hull contribute to the fusion.

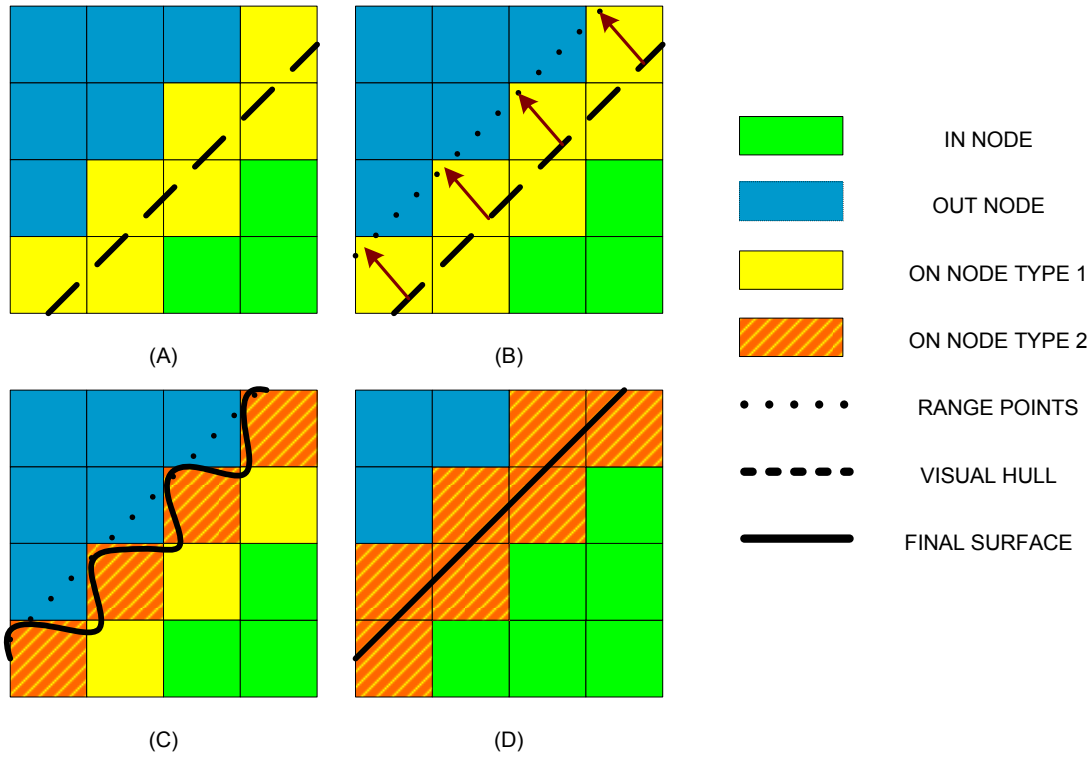

Figure 5.2: Incorporation of outlier range data. (A) Visual hull of a part of a sample object surface. (B) Range points are placed on their respective positions in the grid. (C) After the data from two sources are fused, triangulated patterns of dents begin to appear where the range data are missed out. (D) When those OUT nodes with range points are converted to ON, the curving patterns disappear on the final triangulation. Those ON nodes which no longer share a corner to an OUT node are converted to IN to prevent a multi-layering on the triangulation.

For each range point $\Re_i$, $i = 1,2,\ldots,X$

Locate the leaf node $S_{\Re_i}$ representing the voxel that $\Re_i$ resides in

If $S_{\Re_i}$ is not OUT

Associate $S_{\Re_i}$ with $\Re_i$

Else

If $S_{\Re_i}$ is adjacent to an ON node

Convert $S_{\Re_i}$ to ON TYPE 3

Associate $S_{\Re_i}$ with $\Re_i$

For each maximum level ON node S_j

If S_j is no more adjacent to any OUT node

Convert S_j to IN

For each range point $\Re_i$ at coordinates P_i, camera's optical centre O_i, and the line segment joining P_i and O_i, denoted as L_i, $i = 1,2,\ldots,X$

From among the ON nodes that intersect with L_i, find the closest to O_i and mark this node as S_{L_i}, the *intersection node*

Locate the leaf node $S_{\Re_i}$ representing the voxel that $\Re_i$ resides in, or the *range node*

For each leaf node S_j between S_{L_i} and $S_{\Re_i}$ that intersects L_i

If S_j is associated with any range point

If the state of S_j is ON TYPE 1 or TYPE 2

Extract the triangulation of S_j using silhouettes

If L_i intersects the patch of triangles derived from S_j

Change the state of S_j to ON TYPE 2

Break out of carving with the current range point

Else (the state is IN or ON TYPE 3)

Change the state of S_j to ON TYPE 3

Break out of carving with the current range point

Else (the node is not associated with any range points)

If the state of S_j is ON TYPE 1 or TYPE 2

Extract the triangulation of S_j using silhouettes

If L_i intersects the patch of triangles derived from S_j

Change the state of S_j to OUT

Else (the state is IN)

Change the state of S_j to OUT

Listing 5.1: Basic routine for carving the octree structure with a set of *X* range points.

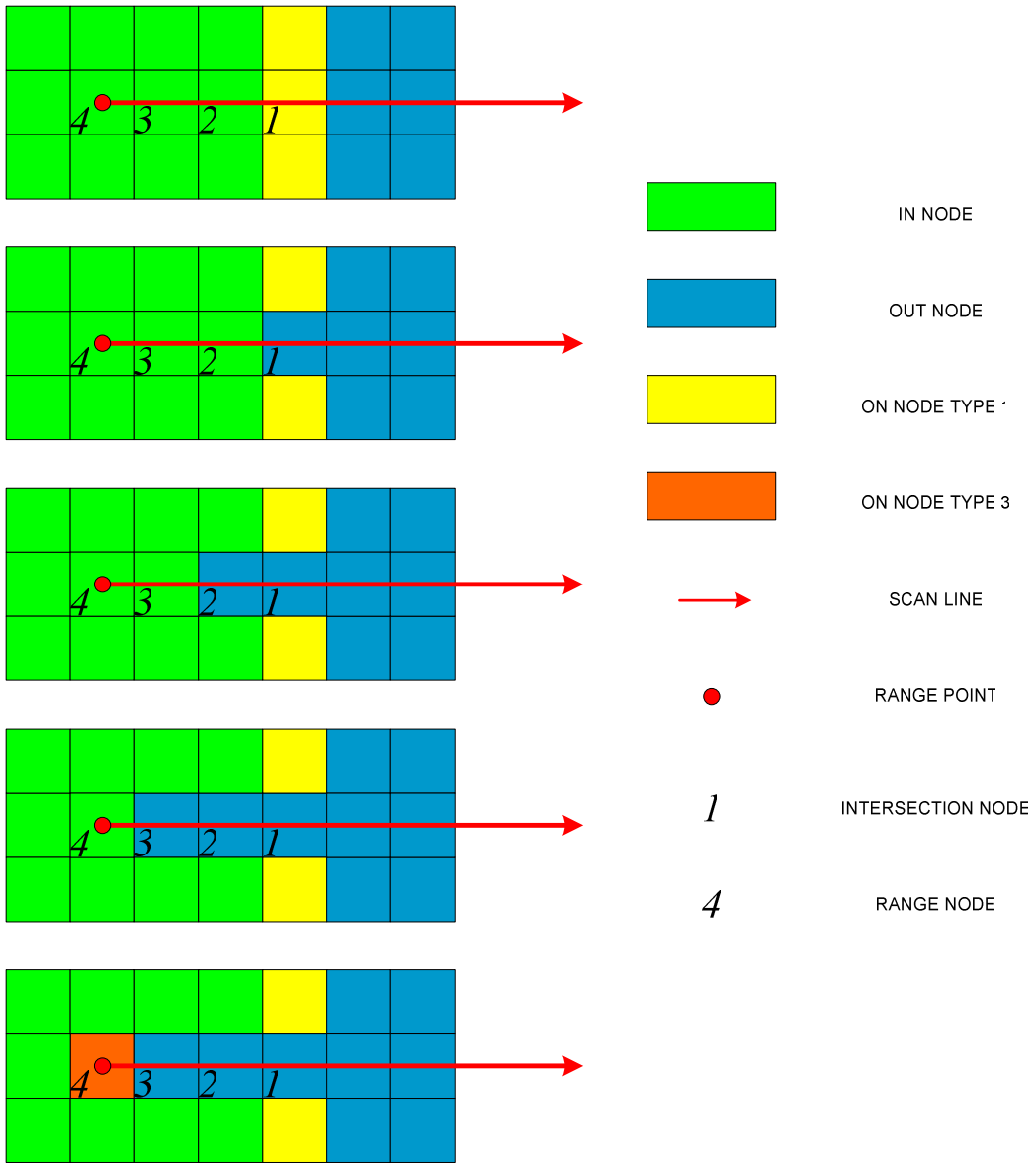

Figure 5.3: Carving sequence**.** (Top) The scan line successively carves the object grid starting from the intersection node, until the range node is encountered (Bottom). The intersection and range nodes are specific to this single range point and take on different values for each new range point.

The depicted routine continues for each range point, $\Re$, by carving the object space starting from the *intersection node*, which is the outermost node intersecting the scan line (see Figure 5.3). The intersection node can simply be found by comparing the distance from intersection points of the relevant nodes to the camera position. Along the scan line, each IN or ON node is turned to OUT until essentially the node that contains $\Re$, the *range node*, is reached. There can be made exceptions to this indiscriminate carving. A question is raised as to what should be done when a node associated with another range point is encountered along the scan line of $\Re$. It can be opted to halt the carving for this particular range point $\Re$ or continue with the carving. The latter, which indicates converting every node in between the intersection node and the range node to OUT, regardless of whether any intermediate node contains any range points or not, gives way to possible degradation in the estimation of the object shape since some ON nodes that are known to be on the object hull are forced to be carved out. It is possible that a whole layer of ON nodes with range points representing accurate shape get eliminated as depicted in Figure 5.4. Therefore, when advancing along the scan line if a node with a range point is encountered, it is favoured to stop the carving and resume with the next range point.

Furthermore, other complications can arise for ON nodes that are originally of TYPE 1 state. The scan line running through the relevant ON nodes do not necessarily cut the surface as depicted in Figure 5.5. As indicated in the routine, in order to determine if the scan line indeed intersects the surface of the object defined by the silhouettes, these ON nodes are triangulated and each generated triangle inside the node must be tested for intersection with the scan line. If no intersection is encountered, then the node remains uncarved. In reconstructions of higher resolutions the loss of a voxel is of less significance, since the associated error is reduced with the decrease in the voxel size. The test for surface intersection becomes redundant and presents a tradeoff between the accuracy of the results and computation time.

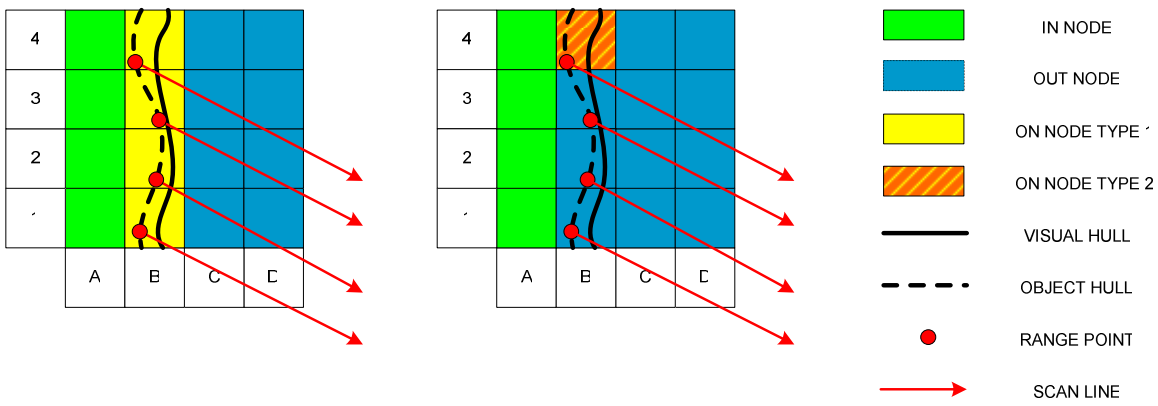

Figure 5.4: Breaking out of carving. (Left) A section of the world grid before carving is initiated. The range points, sensitive to indentation on the object surface, are located on the object hull. (Right) The same world grid after the range points are used to carve away the voxels that intersect with their scan lines. The scan line of the range point located in the voxel B2 runs through B1, causing B1 to convert to OUT and get eliminated. Similarly, B3 scans through B2 and B4 scans through B3, ultimately, eliminating the whole layer of ON nodes. In practice, this is prevented by breaking out of the carving for a scan line when a node that comprises another range point is encountered.

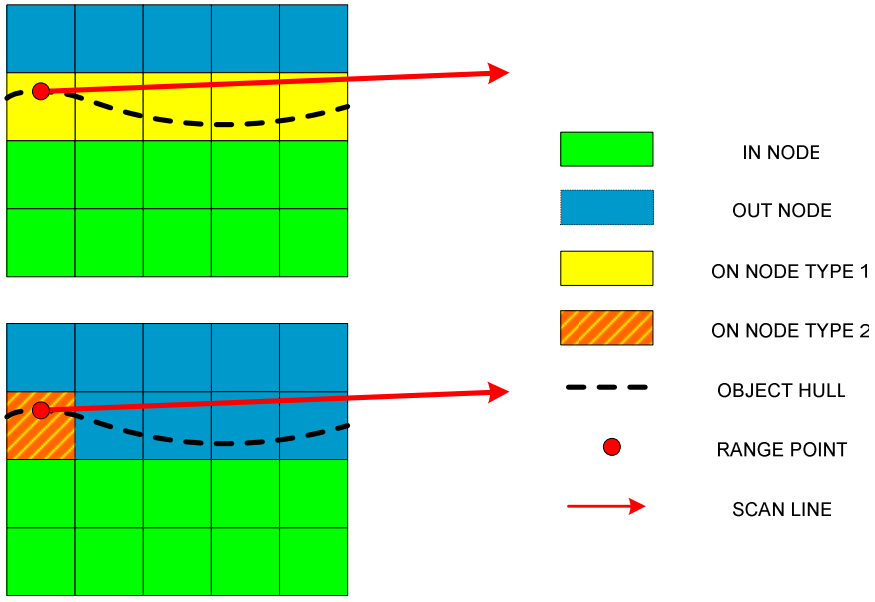

Figure 5.5: Further inspecting the scan lines. (Top) A single range point is positioned on the surface of the object with the scan line running through the rest of the depicted ON nodes. (Bottom) Carving an ON node without first checking if the scan line intersects with the object surface within its boundaries, may lead to its erroneous exclusion from the object volume. Thus, during carving, a scan line that does not intersect with the interpolated surface of an ON voxel does not carve that voxel.

At the end of this initial carving process, the state of the ON nodes in the octree will vary from TYPE 1 to TYPE 3. Figures 5.6 and 5.7 depict the transformation of a sample object space from the visual hull to the post-carved state. Those original ON voxels from the visual hull that are left uncarved remain as TYPE 1 if no range point lies within the voxel. Those original ON voxels from the visual hull that do comprise range data, however, are converted to TYPE 2, meaning they are associated with surface information from both the silhouettes and range data.

While some range points are located inside the ON voxels of the visual hull, others may be bounded by voxels that were originally completely inside the visual hull before volume carving is initiated. Since only ON nodes contain silhouette based surface information, IN voxels that are associated with range points after carving is initiated make use of the range data to extract surface information. Leaf nodes representing such IN voxels in the world grid are converted to ON TYPE 3, which classifies the node as being associated only with range data for the surface extraction.

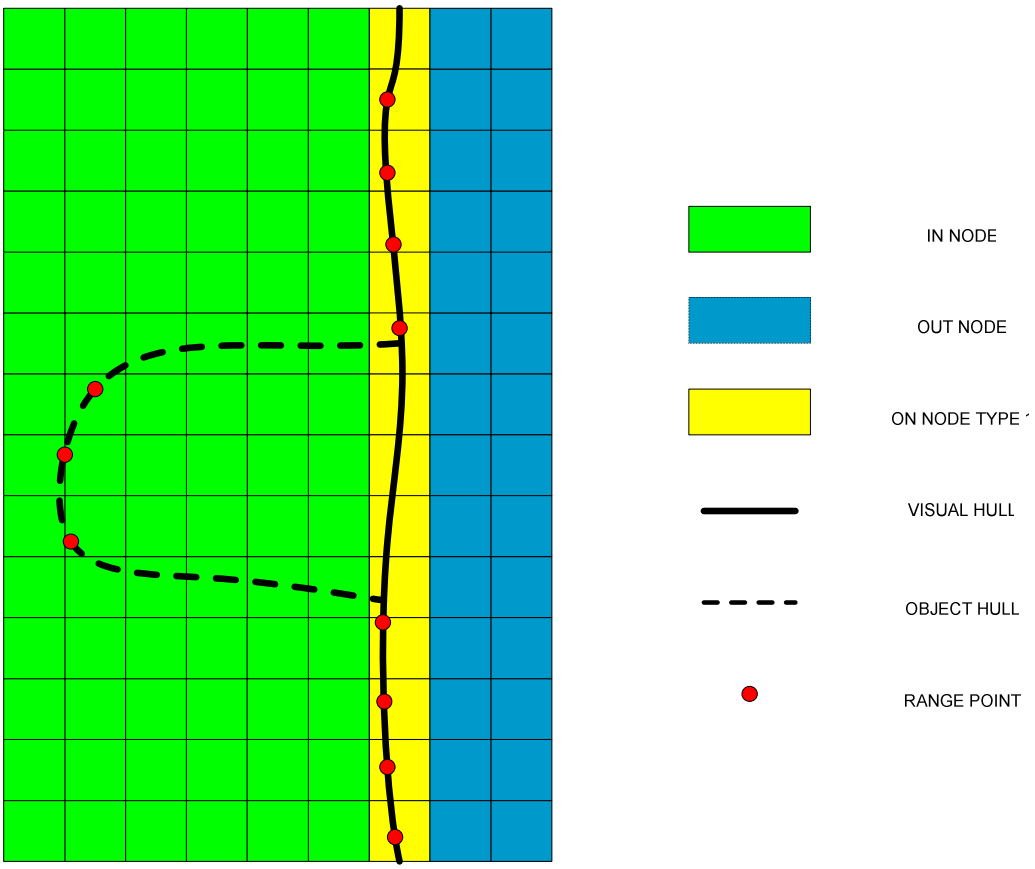

Figure 5.6: A partition of a sample world grid before carving is initiated.

The definition of the octree structure asserts that no IN and OUT nodes can share a common corner, edge, or face on the space grid. This requirement is imposed to prevent holes from appearing on the final triangulated surface. As observed in Figure 5.7 the altered object space is liable to violate this requirement, especially around the edges of deeply scooped out portions of the object. Thus, further modification to the octree based object space is needed to account for the appearance of new OUT nodes that happen to be contiguous to IN nodes.

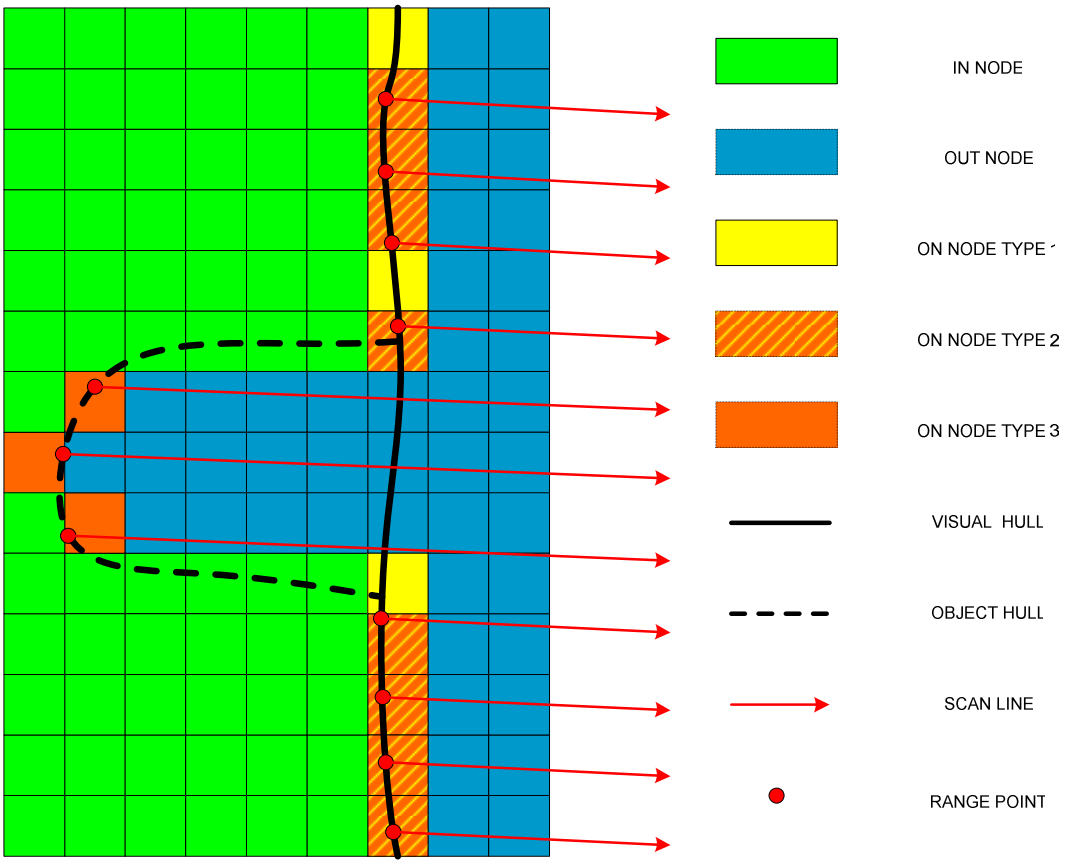

Figure 5.7: The same partition of the world grid of Figure 5.6 after the range point carving is complete. Note the exposing of IN nodes to OUT nodes along the edges of deep cuttings.

The hole filling algorithm is virtually adopted from the work of Curless and Levoy [2]. Curless and Levoy, basically, establish a wall on the transition between the carved out voxels and the unseen voxels in the grid. Similarly, those IN nodes sharing a common corner with any OUT nodes should be converted to ON, specifically of TYPE 4. Use of a new type of ON node is compelling since for these nodes no range or silhouette

information is available for the reconstruction of the object surface. The only way to estimate the object shape is by examining the setting of neighbouring nodes.

Once each potential ON TYPE 4 node is located and marked in the octree, each one is associated with a tuple (P, $\boldsymbol{n}$) to serve as a range point-surface normal pair during the triangulation. The positioning of each IN and OUT voxel in the neighbourhood (constituting 26 voxels) is used to interpolate the point and vector pair, P and $\boldsymbol{n}$. Neighbouring IN and OUT voxels simply act as opposing forces, as IN voxels are used to repel and OUT voxels are used to attract when determining the position of P and the direction of $\boldsymbol{n}$ inside the ON TYPE 4 cube.

Although the concept of octree construction dictates that no IN and OUT nodes can reside sharing a common corner (and implicitly an edge or a face too) on the grid, some exceptions are allowed in the implementation. In fact, when picking potential ON TYPE 4 nodes only those IN and OUT nodes that share a common face (rather than an edge or a corner) are taken into account. In other words, only 6 of the 26 surrounding nodes are examined for violations of the IN/OUT connectivity. The reason behind this less aggressive approach is to avoid the creation of a multilayered shape incurred during triangulation. The IN/OUT connectivity is resolved later in the triangulation process.

Figure 5.8 illustrates the same sample grid of Figure 5.7 after hole filling is carried out.

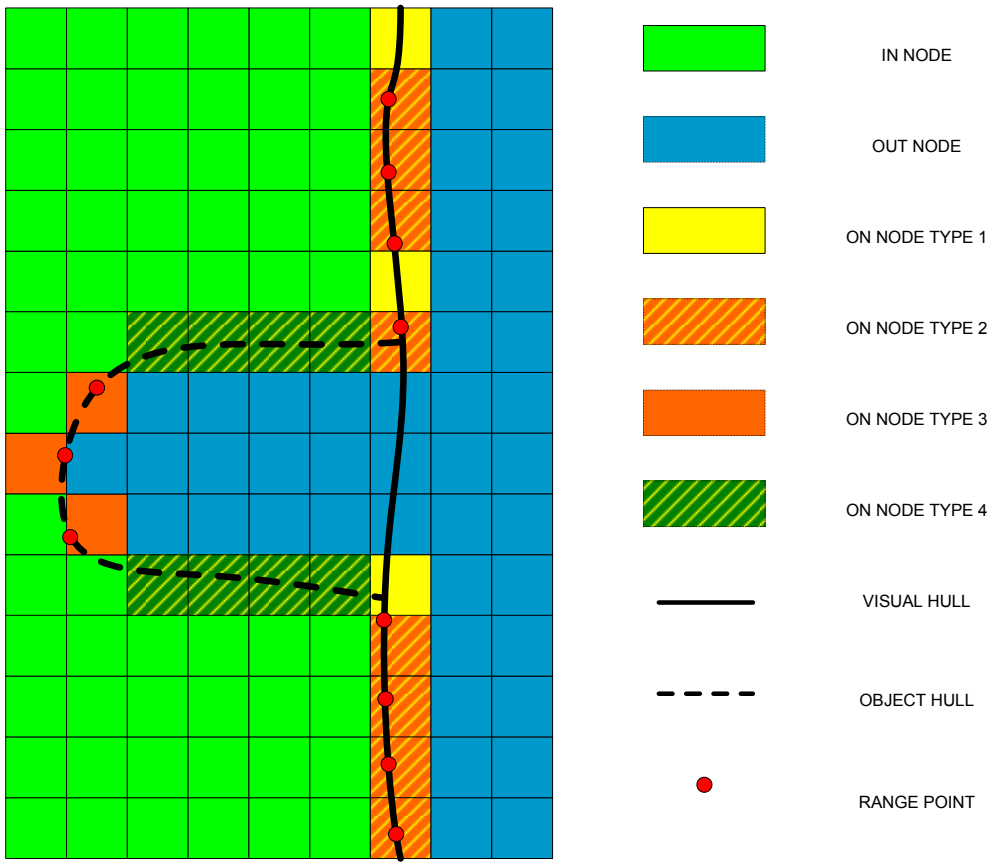

Figure 5.8: The same section of the world grid as Figures 5.6 and 5.7, after IN nodes sharing common sides with OUT nodes are converted to ON TYPE 4.

5.2.2 Triangulation

For the triangulation of the object volume, the marching cubes algorithm is employed with a lookup table that is designed to resolve ambiguous cases. It is simple, fast, and robust. As explained in Chapter 3, marching cubes technique triangulates one cube at a

time and depending on the octree structure usually produces complete and coherent triangulation.

Each ON voxel contains a small piece of the object surface. The partial definition of the isosurface in each voxel is extracted separately and then merged and pasted to form the whole object. To produce hole-free and complete reconstructions, marching cubes relies on the fact that common edges on neighbouring voxels must hold coherent isosurface information. Contrary to the Shape from Silhouette method (Chapter 3), with the different kinds of voxels that are created during the volume carving in this scheme, isosurface information from adjoining voxels will not be consistent. In order to solve this problem, voxel based isosurfaces must be represented in such a way that conflicting information on adjacent voxels is merged soundly and simply. If for each ON node a data structure that gives the exact location where the isosurface cuts the edge is kept, it will be more difficult to combine contradicting information across neighbouring voxels. Figure 5.9 illustrates this by pointing to a probable setting where one voxel decrees that the isosurface passes through an edge while a neighbouring voxel indicates the opposite.

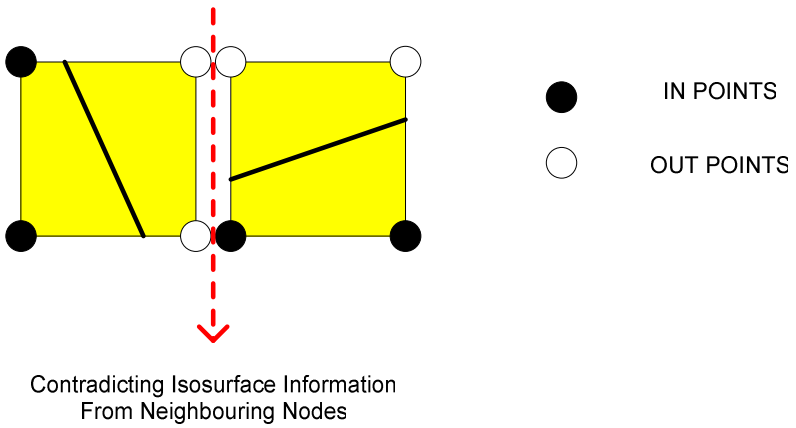

Figure 5.9: Adjacent nodes with incoherent isosurfaces. Resolving such problems is much easier when utilizing isolevel values assigned to the corners of the cubes.

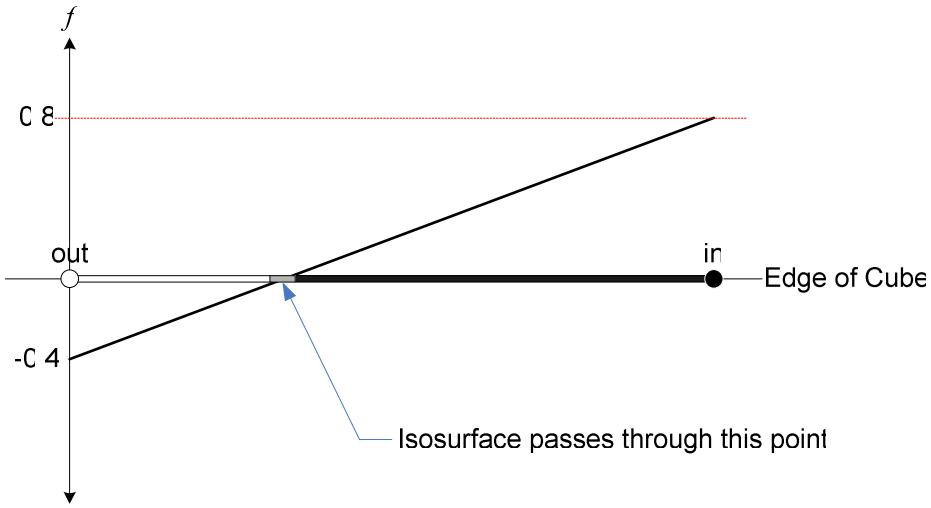

Figure 5. 10: Interpolation of the isosurface across an edge. The intersection takes place closer to the corner whose isolevel value is closer to 0.

This problem can be settled by representing the isosurface in a voxel in the form of eight local isovalues, each assigned to one corner of the cube. An isovalue for a point on the grid determines the extent to which that point is away from the surface of the object hull. The closer to the object hull the closer the value approaches 0. IN and OUT points naturally have opposite signs. Using this scheme, the intersection of the isosurface with the cube edges can be approximated by interpolating between corners of opposite signs (see Figure 5.10). Those edges whose corners have the same isovalue sign do not intersect with the isosurface. In this way, rather than having to resolve isosurface differences on an edge to edge basis, cube corners are used to postulate the final shape of the object.

Once each of the isolevel values for each shared point on neighbouring nodes (referred to as the local isovalues) is computed, a single (global) isovalue must be interpolated for each point, to perform marching cubes. The blending of the local isolevel values from neighbouring nodes to obtain a global and final isolevel is straightforward as explained in section 5.2.2.2. The algorithm relies on the simple ability to accumulate the values from separate nodes for the same point. The main challenge lies in choosing ideal ways for computing local isolevel values for the corners of different types of ON nodes.

5.2.2.1 Computing The Isovalues

TYPE 1: Triangulation information for ON TYPE 1 nodes is known in advance. The algorithm for assigning isovalues to the corners of ON TYPE 1 nodes basically assumes that the local isovalue for each corner is determined by the distance to the closest triangle inside that cube. The distance of a corner to its projection on the plane of the closest triangle generally gives a sound approximation to how far away from the isosurface that point lies. Some poor calculations arise when, although the triangle itself is distant to a particular corner; the plane of the triangle passes relatively closely to the corner. Here, it is important to note that the vertices of the known triangles are on the edges of the cubes. If the distance from a corner to a vertex of the closest triangle is more than l, the side length of the maximum level cube, then it is safe to assume that that corner is not on any edge that contains a vertex of a triangle. In such cases the distance to the closest triangle is computed as the minimum of the distances from the corner to the three vertices of the associated triangle. The local isovalue $\widehat{f}$ of each corner C_i, (of the sequence $i = 1,\ldots,8$) of an ON TYPE 1 cube of side length l, is computed as:

$$\hat{f}(C_i) = \frac{\min_j \Delta_{ij}}{l} \tag{5.1}$$

where Δ_{ij} (the interpretation of the signed distance from the corner C_i to the j'th triangle T_j, inside that cube, connecting the vertices V_{jn}, n = 1,2,3) is defined as:

$$\Delta_{ij} = \begin{cases} d(C_i, T_j) & \text{if } \min_n d(C_i, V_{jn}) < l \\ \min_n d(C_i, V_{jn}) & \text{else} \end{cases} \tag{5.2}$$

where d denotes the signed distance function from a corner of the cube, C_i, to either a triangle, T_j, defined inside the cube or to a vertex, V_{jn}, constituting the triangle T_j. The distance to a triangle T is interpreted as the distance to the projection on the plane of T. The sign of the distance function indicates whether the corner of interest is IN or OUT.

TYPE 3: An ON TYPE 3 node may contain several range points. A single range point carries a normal vector from which a plane can be interpolated that is perpendicular to the normal (see Figure 5.11). If there is a single range point in a TYPE 3 node, then the local isolevel values of the corners can simply be set proportional to the distance from the individual corners to the plane passing through the range point and orthogonal to the range point's normal. When more than one range point is involved, the isolevel values of each corner are computed based on the distance to the plane obtained from the closest range point. The local isovalue of each corner C_i, (of the sequence i = 1,...,8) of an ON TYPE 3 cube of side length l, is computed as:

$$\hat{f}(C_i) = \frac{d(C_i, P_i)}{l} \tag{5.3}$$

where d is the signed distance and P_i is the plane perpendicular to the normal of and passing through $\Re_{C_i}$, which is the closest range point to the corner C_i. By checking which side of the plane P_i, C_i is situated on, the sign for the distance function is determined, which resolves whether C_i is IN or OUT.

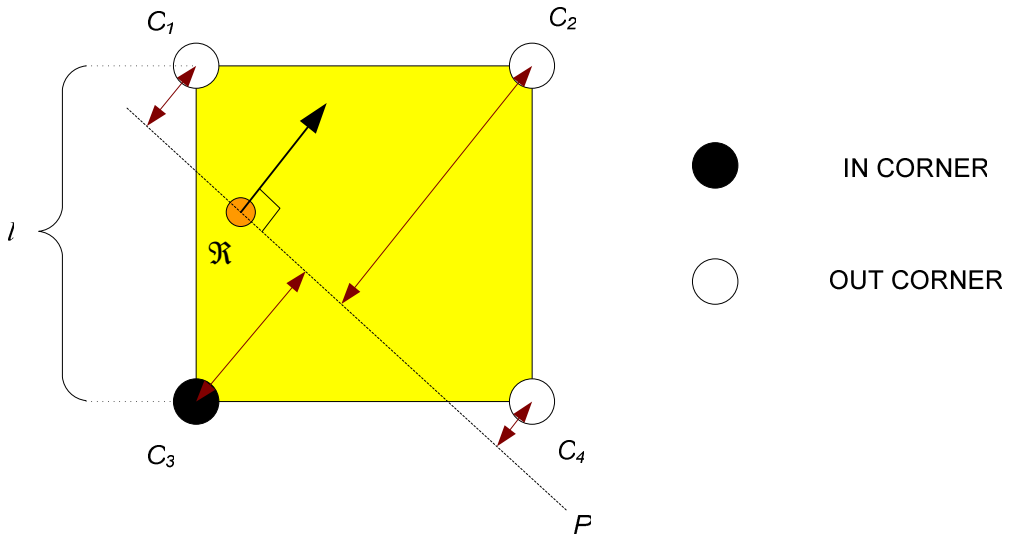

Figure 5.11: Assignment of the local isovalues on the corners of an ON TYPE 3 voxel of sidelength l. A single range point $\Re$, defining a perpendicular plane P, is located in the voxel. The isovalue of a corner is computed as the ratio of the distance to P and l.

TYPE 2: A TYPE 2 ON node, which comprises both silhouette and range data, can be triangulated in three ways. The triangulation methods explained, previously, for TYPE 1 and TYPE 3 nodes are both applicable for TYPE 2 nodes. An alternative is to use a hybrid method that interpolates an isosurface between the triangulation results acquired from both the silhouettes and range data. In practice, since the range data are assumed to be more definitive and precise than the silhouettes, the triangulation method for TYPE3 nodes is adopted.

TYPE 4: Each TYPE 4 ON node is associated with a point and a vector which enables its local isovalues to be computed in the same way as a TYPE3 node that comprises a single range point.

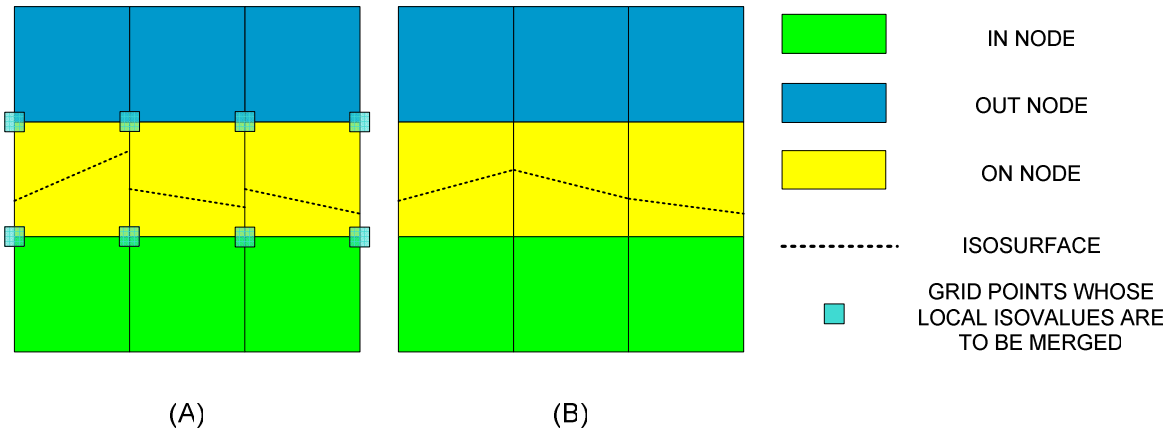

Figure 5.12: A sample grid of object space is given to depict the merging of local isovalues. (A) The marked grid points are on the corners of ON nodes and thus subject to computation of isovalues. Also, triangulation results of the local isovalues obtained from individual nodes are given. (B) After the local isovalues from adjacent nodes are merged, a coherent set of surface patches are generated.

5.2.2.2 Merging The Isovalues

Considering the triangulation technique employed, in order to attain a continuous and coherent surface, the isovalues for the same grid points on adjacent nodes must be consistent. As Figure 5.12 shows, adjacent nodes in the grid may and most probably do, give contradictary values for the isovalues that they indicate on shared corners. If these values are not somehow equated, then marching cubes will produce unconnected patches of surfaces which will generate a final triangular mesh filled with holes.

Each grid point which is a corner to at least one ON leaf node is assigned a local isovalue by each of the ON leaf nodes the grid point is a corner to. The global isovalue *f* for

a point P on the grid is computed by simply averaging over those local isovalues $\hat{f}_i$, that are extracted from N number of ON nodes, where $i = 1,2,...,N$:

$$f(P) = \frac{\sum_{i=1}^{N} \hat{f}_i(P)}{N} \tag{5.4}$$

Local isolevel values for the same grid point gathered from adjacent ON leaf nodes may vary greatly such that some of the ON nodes may indicate the shared point to be IN (with a positive value) while others may indicate that it is OUT (with a negative value). By averaging the negative and positive isovalues for a grid point, the estimated distance of the isosurface to the grid point is also averaged.

Occasionally, a single range point close to one corner of a node leads the isolevel values of the far corners to be misinterpreted as depicted in Figure 5.13-A,B. These mistakes mostly result in protuberances on the object surface that look unnatural. The fusion scheme of averaging (5.4) is modified in order to overcome this effect: for each point on the grid that is assigned local isovalues by ON nodes, a count of the number of OUT nodes that share that grid point is kept also. Note that this number cannot be more than 7 since out of the 8 nodes that share a grid point at least one of them must be ON in order to qualify for an isolevel assignment. If the OUT node count exceeds 6 for a point P on the grid (leaving at most 2 ON nodes sharing P), the corresponding isovalues of the OUT nodes that contain P (which are defaulted to a fixed negative value) contribute to the average that determines the global isovalue f of P, such that the global isovalue f is computed as:

$$f(P) = \frac{\sum_{i=1}^{M+N} \hat{f}_i(P)}{M+N} \tag{5.5}$$

where M denotes the number of OUT leaf nodes and *N* is the number of ON leaf nodes that *P* is a corner to. In effect, this causes *P* to convert to OUT because of the majority of the OUT nodes sharing *P* (see Figure 5.13-D). In the case where the count of OUT nodes sharing *P* does not exceed 6, equation of (5.4) is used.

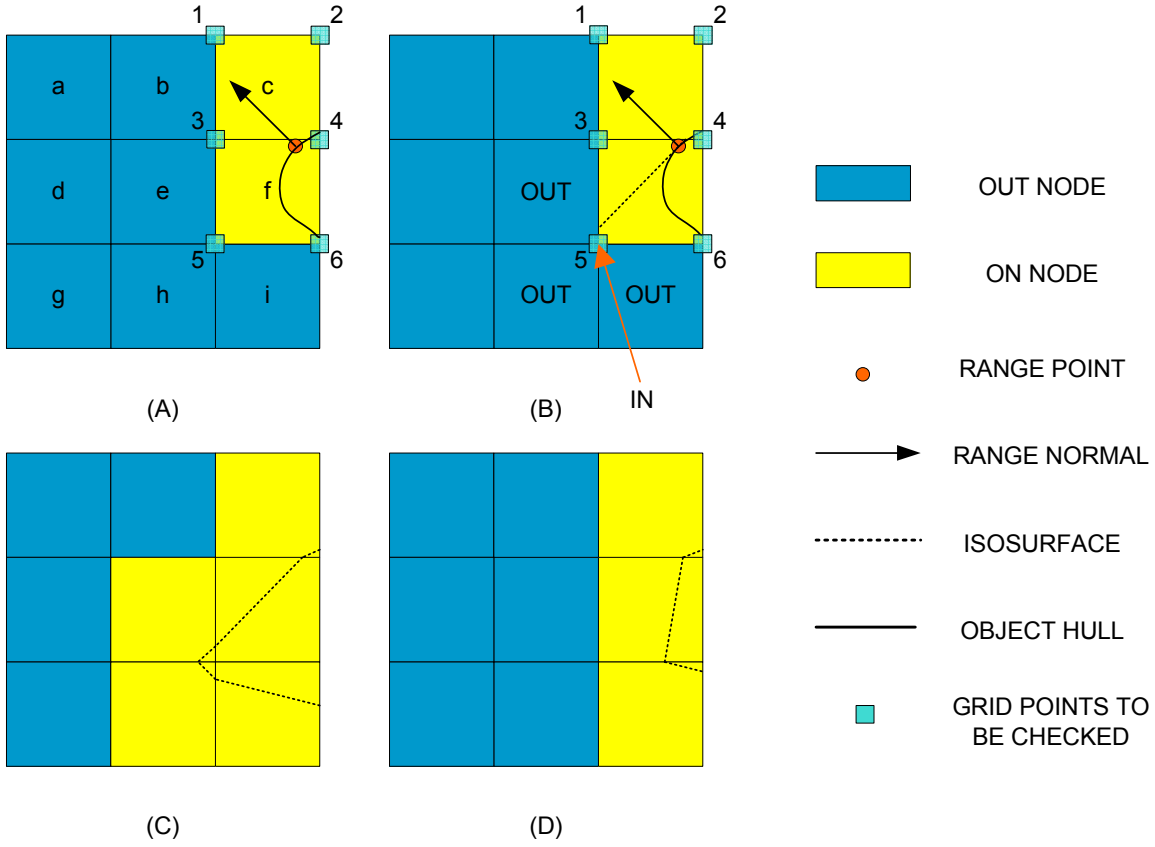

Figure 5.13: Attenuating protrusions caused by misinterpretation of range data. (A) A section of an object hull is given with the marked grid points whose isovalues are sought. The optical scan reveals a range point in node f with the given direction of its normal vector. (B) The interpolated isosurface in the individual ON nodes are computed. The isosurface of node f is the plane perpendicular to the normal of the range point, which makes the grid points 4, 5, and 6 be interpreted as IN and point 3 OUT of the hull. Note that point 5 is surrounded by OUT nodes. The OUT nodes that are found to have IN corners (e, h, i) must be converted to ON (section 5.2.2.3). This can be followed in two different scenarios as illustrated in (C) and (D). (C) If the OUT nodes e, h, and i do not contribute to the computation of the isolevel on point 5, point 5 will remain IN and nodes e, h, i will turn ON at a later stage to form a misshaped extrusion. (D) If the OUT nodes e, h, i do contribute to the computation of the isovalue of point 5, point 5 will convert to OUT and the incorrect extension will be attenuated to a degree.

5.2.2.3 Final Modification to the Octree

By definition of the octree, all the corners of OUT cubes must be OUT and likewise, all the corners of IN cubes must be IN. After merging the isovalues of the corresponding corners on adjoining nodes, the resultant states of the grid points may break this rule, i.e. the same point on the grid may be shared by both an OUT cube that defaults the shared point to be OUT and an ON cube whose final isovalue assigned to the shared point is positive (defining the point as IN). If these contradictions are not resolved, there may appear holes on the hull reconstruction after marching cubes is applied.

In practice, as depicted in Figure 5.14-A,B,C, OUT nodes with at least one corner declared as IN by an adjoining ON node are converted to ON. The same applies for IN nodes that have corners declared to be OUT by adjoining ON nodes, as depicted in Figure 5.14-D,E,F. Since IN and OUT cubes contain no isovalue information for their corners, the isovalues of the corners of the cubes that are converted must be copied from the corresponding corners on the adjoining ON nodes if available. In the case where no isovalue is available for a corner (when there is no adjoining ON cube) the isovalue defaults to -1 for OUT cubes and 1 for IN cubes. The default values are kept large to prevent over-peaking where the holes are sealed.

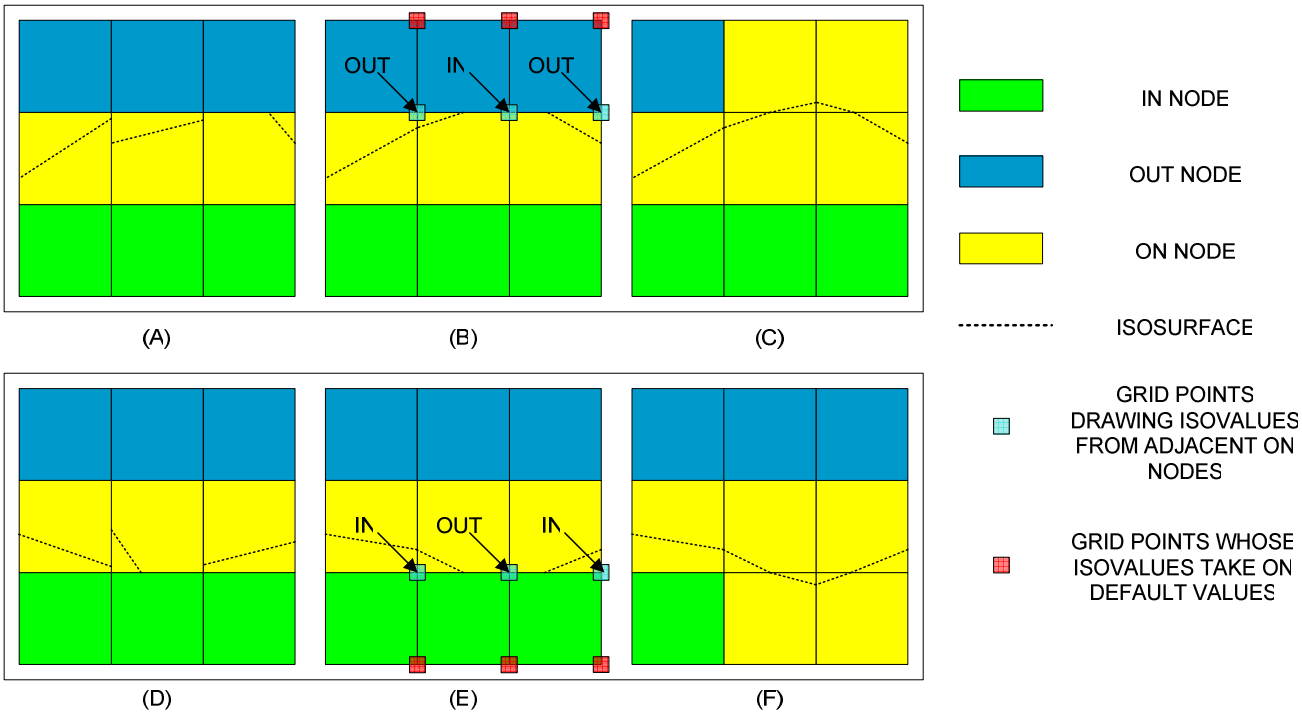

Figure 5.14: Demonstration of covering up holes. OUT and IN nodes, by definition, do not get triangulated. Some OUT and IN nodes end up sharing IN and OUT corners, respectively, with ON nodes and if not triangulated will produce holes in the reconstruction. Depictions on the top and bottom rows are such examples. (A)-(D) A section of a sample object space before the fusing of isovalues of shared grid points are carried out. (B) After fusion, the object space of (A) is triangulated such that a hole on the hull is exposed. There are two OUT nodes that share an IN corner with ON nodes. (D) Two IN nodes share an OUT corner with ON nodes. The OUT nodes in (B) and the IN nodes in (D) must be converted to ON to seal shut the visible holes. (C) and (F) reveal the final states of the nodes and triangulation results of (A) and (D), respectively.

5.2.3 Triangle Decimation

The final triangular mesh obtained from applying marching cubes algorithm to the octree contains large number of faces that are too small to make much difference to the overall shape. Abundance of these superfluous faces slows down the visualization, degrades the shading, allocates valuable resources, and performs poorly with texture mapping. The implemented decimation algorithm [9], [2] eliminates those faces with edges smaller than a threshold of half length of a cube. The face is either collapsed into a point (on the condition that two or three of the edges are below the threshold), or a segment (if only one edge of the face is below the threshold). The position of the merged vertices is established as the mean of the collapsed points. The implemented decimation process eliminates approximately 35 % of all the vertices and faces created by marching cubes (see Figure 5.15).

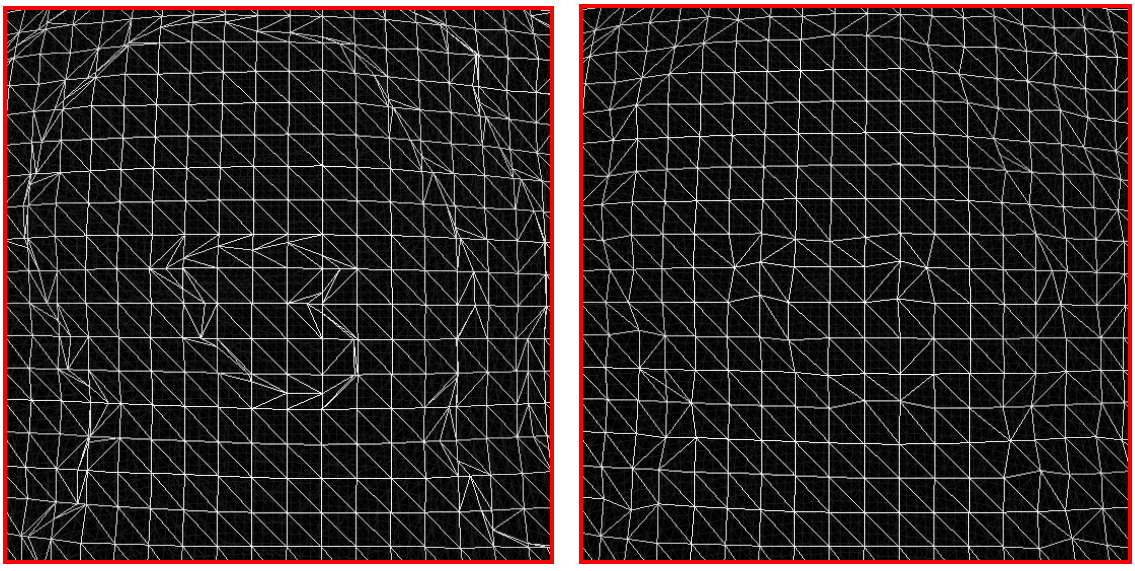

Figure 5.15: A sample triangular mesh before and after decimation is performed.

5.2.4. Surface Fairing

Surface fairing is conducted to further smooth the object shape at sharp and jagged protrusions that look unnatural. Fairing is sometimes exclusively applied to faces obtained from TYPE 4 nodes because these nodes produce especially rough surfaces on the transition from TYPE 4 nodes to other types of neighbouring nodes. The employed method [18] computes a displacement $\Delta \boldsymbol{v}$ for each vertex of the triangle mesh and later each vertex is shifted by a multiple λ, of its computed displacement:

$$\boldsymbol{v}' = \boldsymbol{v} + \lambda \Delta \boldsymbol{v} \tag{5.6}$$

where $\boldsymbol{v}'$ is the new position of the vertex, and $\boldsymbol{v}$ is the old position of the vertex. The displacement $\Delta \boldsymbol{v}$ of a vertex is the weighted mean over the individual displacements from the involved vertex to its first level neighbours. The weight can be chosen as the surface area of the two faces that share the edge, or some power α of the length of the edge $\|\boldsymbol{v} - \boldsymbol{v}_n\|^{\alpha}$, where $\boldsymbol{v}_n$ denotes the position of a neighbouring vertex. In practice the weight is chosen as the length of the edge to the power of α = -1.

The application of the computed displacements to vertices with the coefficient λ, where $0 < \lambda < 1$, serves as a Gaussian filter that attenuates high frequencies on the triangle mesh [18]. Since Gaussian filters are not low-pass filters, application of (5.6) alone produces shrinkage (see Figure 5.16-D,E,F). To counter the shrinking effects, two sets of displacement are carried out giving the effect similar to that of a low-pass filter. First the computed displacement is applied with the coefficient λ as in (5.6), where $0 < \lambda < 1$, and after the vertices are repositioned the displacement vectors are recomputed and applied this

time with an unshrinking coefficient μ as in (5.7), where $\mu < -\lambda$, which in effect prevents the shrinking of the object volume (see Figure 5.16-B,C).

$$\boldsymbol{v}' = \boldsymbol{v} + \mu\Delta\boldsymbol{v} \tag{5.7}$$

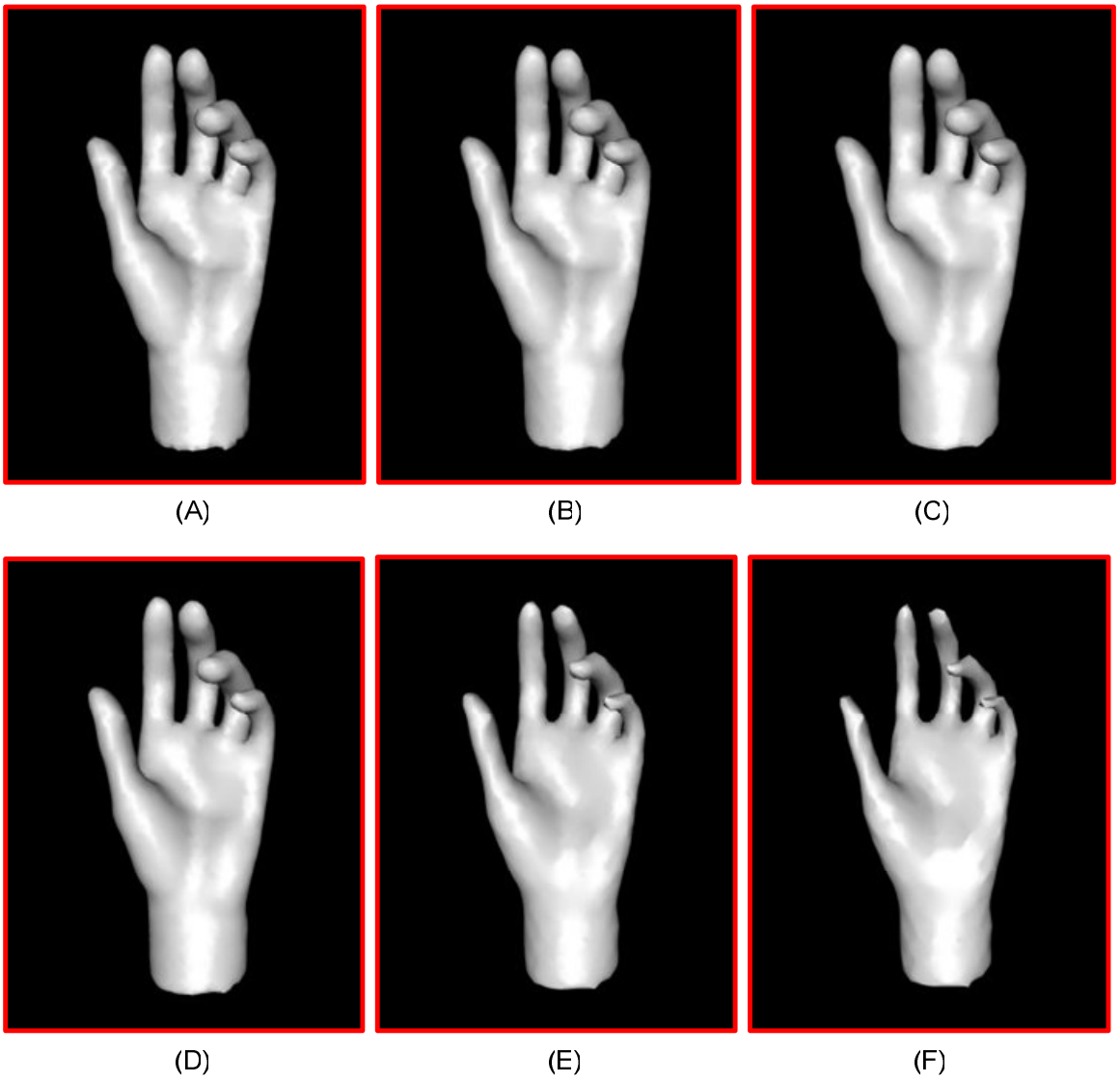

Figure 5.16: Reconstructions of a hand model having undergone different fairing steps. (A) Gouraud-shaded rendering without any fairing. (B) After application of 20 steps of non-shrinking fairing $(\lambda = 0.63, \mu = -0.67)$. (C) After application of 50 steps of non-shrinking fairing. (D) After application of 2 steps of shrinking fairing $(\lambda = 1.0, \mu = 0.0)$. (E) After application of 10 steps of shrinking. (F) After application of 20 steps of shrinking.

Chapter 6

RESULTS

Reconstructions of 4 models are presented to emphasize the robustness of the algorithm and indicate where the system succeeds and fails. As depicted in Figure 6.1 the models are:

- Greek1
- Greek2
- Elephant
- Hand

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 images acquired varies regarding the shape and size of the reconstructed object. Table 6.1 lists the number of images acquired for each model.

Table 6.1: Number of images taken for each model.

Model	Silhouette	Laser 1	Laser 2	Laser 3
Greek1	72	360	-	-
Greek2	72	360	-	-
Elephant	72	360	360	-
Hand	72	180	180	180

Figure 6.1: Original images from Greek1, Greek2, Elephant, and Hand models.

Figures 6.2 through 6.13 illustrate the results obtained. The Greek1 sequence in Figure 6.2 demonstrates the fusion process at resolution $R = 7$. The silhouette reconstruction typically lacks detailed features of the face while the optical triangulation exhibits many holes. The fusion at resolution $R = 7$, reveals a watertight reconstruction that includes accurate cavity shape deduced from the optical triangulation. However, the fused result is less sharp than the optical triangulation which is caused by the low resolution of the octree used. Figure 6.3 shows the result of fusing the same Greek1 model in resolution $R = 8$. Here, the observed fused model achieves the desired level of detail but due to the insufficient density of the range scan lines some voxels of the silhouette volume remain uncarved. Figure 6.4 demonstrates reconstructions of the Greek2 model. The fusion result for this model shows the need for further carving. It is important to note that the hollow parts on the handles of Greek2 model are obtained from the silhouettes. The range scanner of our acquisition system fails to detect these parts. However, even more sophisticated range scanners alone would not be able to reconstruct these parts of the object completely or reliably because of the high angle of incidence involved. The Elephant sequence in Figure 6.5 further shows that our fusion scheme can be used to generate very satisfactory results. The obstructed inner faces of the legs and the trunk and though not visible at the depicted angle, the top sections of the model, are missing from the optical triangulation and are complemented with the silhouettes. Figure 6.6 depicts a succession of optical triangulations of the Hand model. It is observed that with each scan the range scanner produces laser planes that converge towards certain locations on the object surface. As a result, the range scanner scans much less of the visible object surface that is actually not occluded. Even though three separate 360-degree scans were made for the Hand model the fusion results shown in Figure 6.7 manage a satisfactory carving at resolution $R = 7$ (with an oversized bounding box making the effective resolution $R \approx 6.5$). At resolution $R = 8$ the prominent defects become visible once more. Separate fusion results of the silhouette model at $R = 7$ (Figure 6.7, left) and the individual optical triangulation results (Figure 6.6) are depicted in Figure 6.8.

Statistics for the reconstructions are provided in Table 6.2. Figures 6.8 through 6.13 show other views of the reconstructions.

The results show that when performing the fusion process at high resolutions there begin to appear uncarved cubes that reside within the boundaries of the object reconstruction and are ought to be carved to attain the correct shape of the object. Even when working in low resolutions, as experienced in the hand model, there may be required several range scans, which are time consuming, before the fusion results are acceptable. This problem can be solved with the use of a different range scanner that produces more complete and well distributed scans. Discussed in Chapter 5, scanners that produce full range images promise great potential in achieving higher resolution and more precise results for the optical triangulation and the final fusion. Note that using range scanners that produce better models than the one currently being used does not invalidate this work. The occlusion problem, which is the main cause for the holes, will persist for such systems and more accurate means of reconstructing objects such as the methods discussed in this report will still be sought.

Figure 6.2: Reconstructions of Greek1 model. Each rendering is Gouraud-shaded. (Left) Silhouette model of resolution $R = 7$ after decimation. (Middle) Structured Light model after fairing. The traceable marks of the laser stripes on the object surface are a result of miscalibration and noise. (Right) Fused model at $R = 7$ resolution after decimation. Note the slight blurriness in the fusion due to insufficient depth of traversal in the octree.

Figure 6.3: Reconstructions of Greek1 model continued. Each rendering is Gouraud-shaded. (Left) Silhouette model at resolution 8 after decimation. (Middle) Structured Light model after fairing. (Right) Fused model at resolution $R = 8$. Note the overhanging irregularities beside the nose and below the hair line where the scan lines are too sparse to carve away the volume. The bumpy surface on the base of the fused reconstruction originates from miscalibration which mostly tends to cause misalignment of the scannings to materialize on the top and bottom sections of the fused reconstruction.

Silhouette Reconstruction: The outcome typically lacks cavity information around the face but the inner walls of the hollow parts are captured very well. A range scanner would most likely not detect these parts due to occlusion or even if captured (with more advanced scanners) the acquired data would be scarce and unreliable because of the grazing orientation of the surface normals with respect to the camera view.

Structured Light Reconstruction: Although the details around the face are very well caught the profuse holes in the reconstruction create an unsatisfactory result. Usually hole filling algorithms would be used but the hollow parts of the handles visible in the silhouette reconstruction would fail to be detected.

Fusion Reconstruction: The fusion reaches the quality of the active structured light method while retaining the completeness of the passive silhouette method. However, with the apparent holes in the laser reconstruction the fused model still needs carving around the left handle of the cup. The ellipsoidal shape of the object presents a major problem for a single-striped rotational range scanner, since the laser stripes tend to concentrate more towards the center of the object.

Figure 6.4: Reconstruction of the ellipsoidal shaped Greek2 model. Reconstruction conducted at resolution $R = 8$.

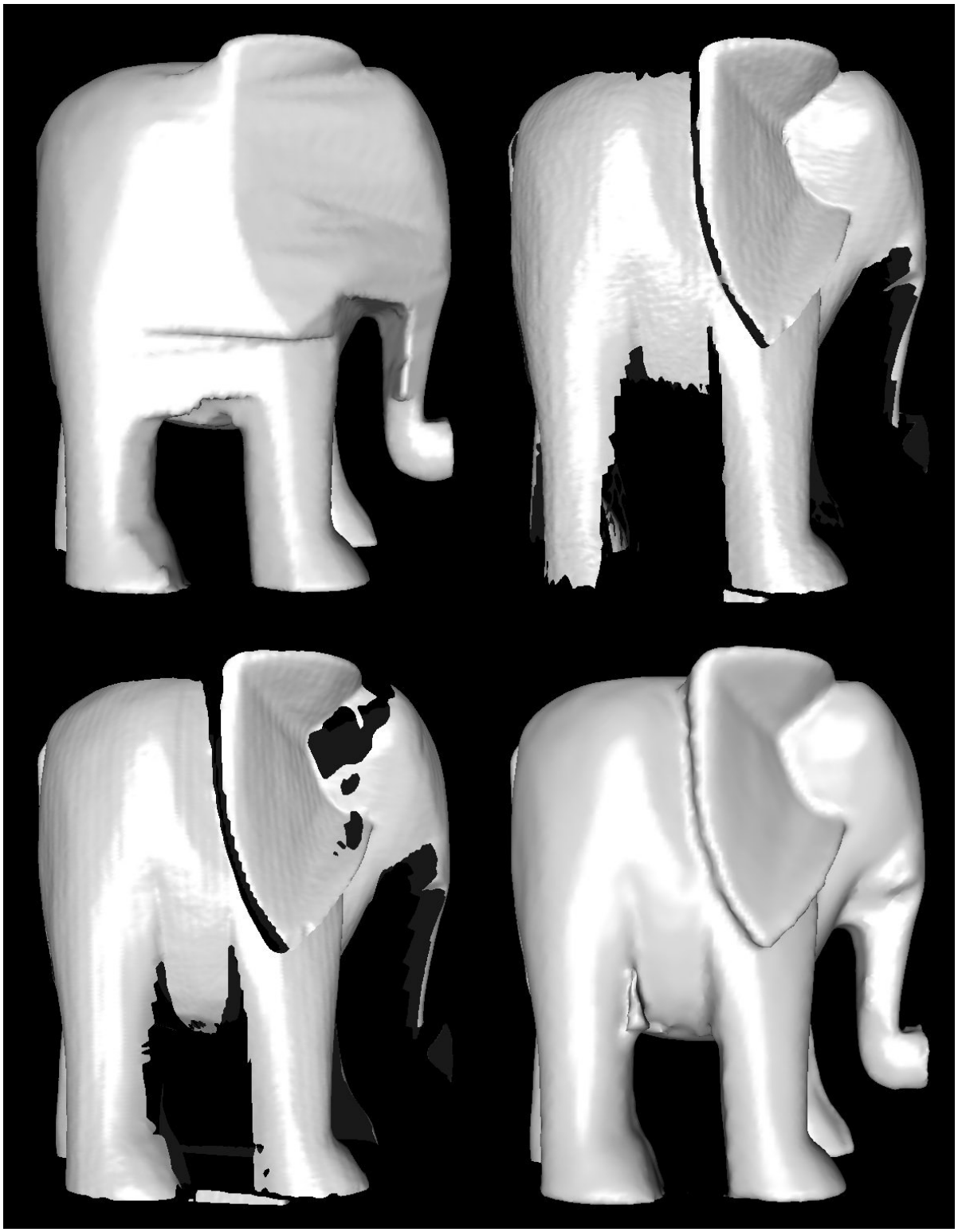

Figure 6.5: Reconstruction of the Elephant model. From top left to bottom right: Decimated silhouette ($R = 8$), first and second laser, and decimated/faired fused reconstructions are depicted respectively. The ears that are missing from the silhouette reconstruction and the interior parts of the legs and the trunk that are missing from the structured light reconstruction are all visible in the fusion. Some artifacts at the belly are a result of deficiency of range data around that area. Note that these parts could be picked out by better range scanners.

Figure 6.6: Faired reconstructions from three separate structured light scannings of the hand model. For each scanning the laser projector is positioned differently to increase the chance of a richer sampling. Scanning on a rotational course, the range scanner used for the acquisition produces very poor results for this model. Towards the centre of the object (the palm) the range samples converge where the laser planes from consecutive images intersect. As a result, little amount of useful information can be drawn from a single scanning that is inadequately distributed on the object.

Figure 6.7: Reconstructions of the hand model continued. (Left) Faired silhouette reconstruction at resolution $R = 7$. The palm is not perceived by the silhouettes and needs to be carved by the range data. (Middle) Final reconstruction at resolution $R = 7$ formed by fusing the silhouette and three scannings from the range scanner. Although the carving is successful, the reconstruction is blurry with respect to the structured light results. (Right) Fused reconstruction at resolution $R = 8$. The reconstruction is sharper than the $R = 7$ counterpart but the carving fails due to scarceness of the available range data.

Figure 6.8: Three faired reconstructions of the hand model obtained from separately fusing a silhouette reconstruction of resolution $R = 7$ displayed in Figure 6.7 with the three structured light scannings displayed in Figure 6.6. Each fused reconstruction shows signs of insufficient carving especially around the palm.

Table 6.2: Statistics for the reconstruction of Greek1, Greek2, Elephant, and Hand models. Number of triangles obtained with SfS (Shape from Silhouette), Shape from Structured Light, and fusion at various resolution levels before and after decimation.

Model	**SfS**	**SfS after decimation**	**Laser 1**	**Laser 2**	**Laser 3**	**Fusion**	**Fusion after decimation**
Greek1	(R = 7) 67K (R = 8) 272K	(R = 7) 43K (R = 8) 175K	136K	-	-	(R = 7) 69K (R = 8) 280K	(R = 7) 44K (R = 8) 180K
Greek2	(R = 8) 208K	(R = 8) 143K	104K	-	-	(R = 8) 202K	(R = 8) 140K
Elephant	(R = 8) 145K	(R = 8) 100K	113K	143K	-	(R = 8) 162K	(R = 8) 105K
Hand	(R = 7) 12K (R = 8) 48K	(R = 7) 8K (R = 8) 31K	101K	98K	96K	(R = 7) 12K (R = 8) 49K	(R = 7) 8K (R = 8) 31K

Figure 6.9: More views of the reconstruction of the Greek1 model. Starting from the top left: Silhouette reconstruction ($R = 7$), Silhouette reconstruction ($R = 8$), Optical triangulation, Fusion ($R = 7$), and Fusion ($R = 8$) are depicted.

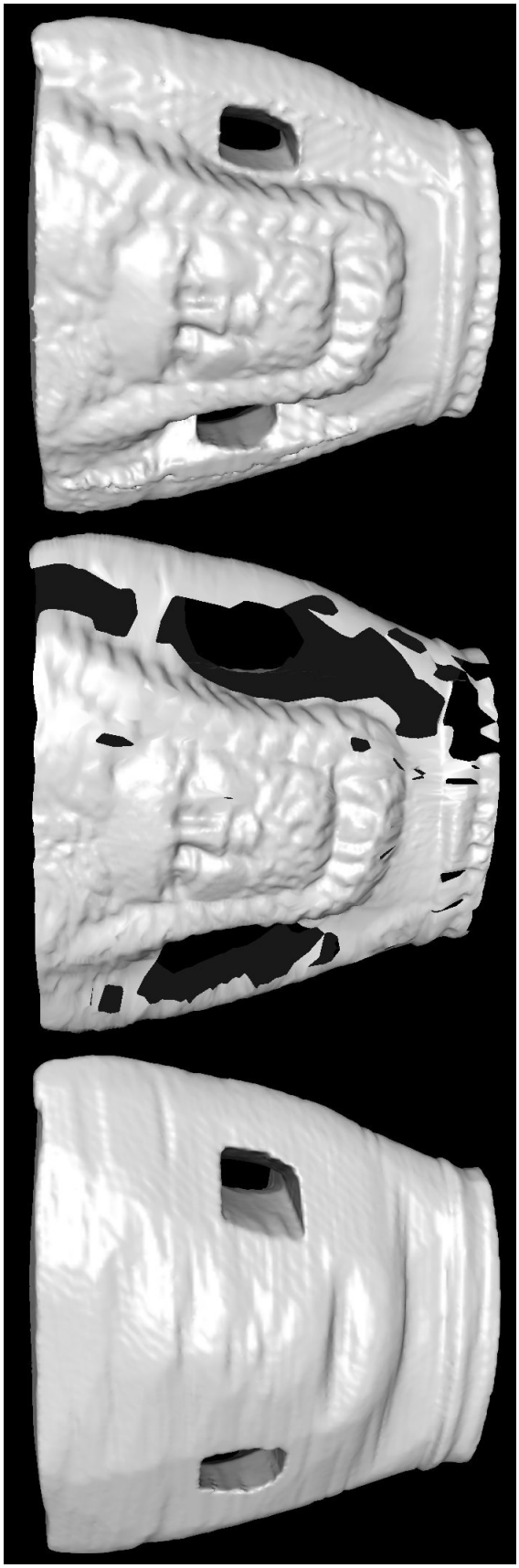

Figure 6.10: More views of the faired reconstructions of the Greek2 model. (Left) Silhouette reconstruction ($R = 8$). (Middle) Optical triangulation. (Right) Fused reconstruction ($R = 8$).

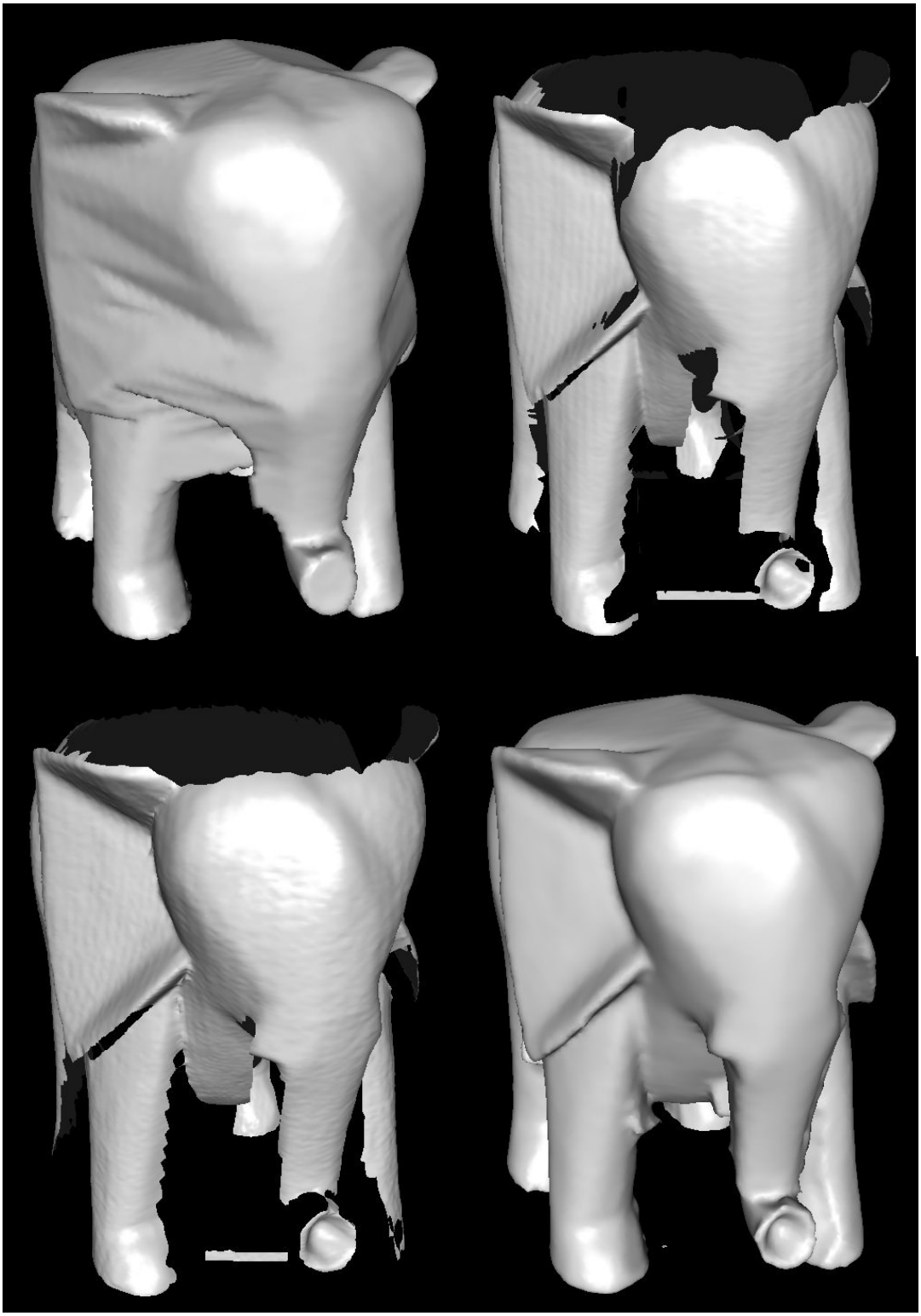

Figure 6.11: More views of the reconstruction of the Elephant model. Starting from the top left: silhouette reconstruction ($R = 8$), optical triangulation from the first laser projector, optical triangulation from the second laser projector, fusion ($R = 8$) are depicted.

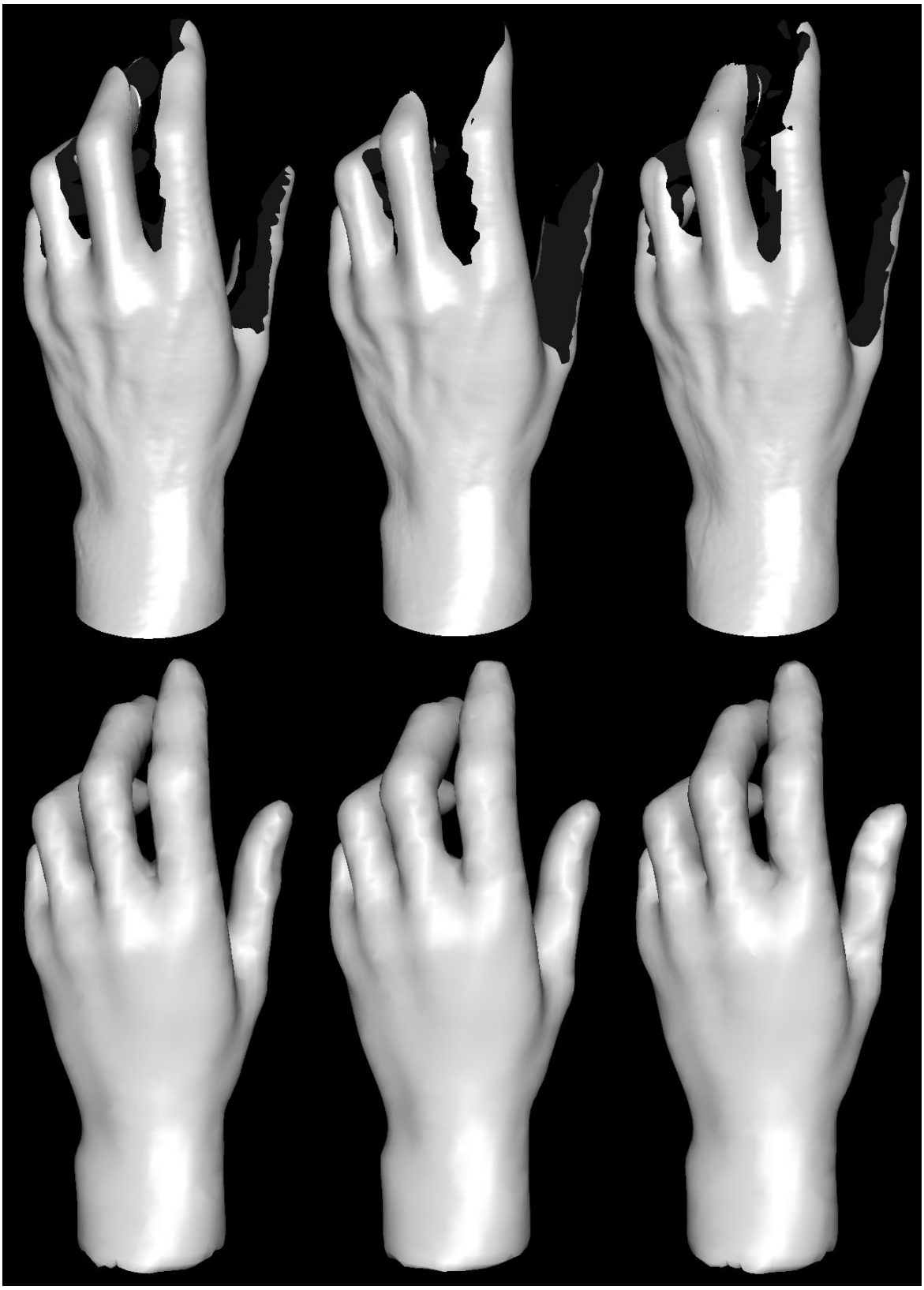

Figure 6.12: More views of the reconstruction of the Hand model. (Top row) First, second and third optical triangulations. (Bottom row) fusion of the silhouette reconstruction ($R = 7$) with the first, second and third optical triangulations respectively.

Figure 6.13: More views of the reconstruction of the Hand model (continued). Starting from the top left: silhouette reconstruction at resolution $R = 7$, fusion at resolution $R = 7$ using the combination of three optical triangulations, silhouette reconstruction at resolution $R = 8$, and finally the fusion at resolution $R = 8$ using the combination of three optical triangulations.

Chapter 7

CONCLUSIONS AND FUTURE WORK

This work has described a novel 3D modeling scheme based on volumetrically fusing two models obtained from Shape from Silhouette and Shape from Structured Light techniques. The aim was 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 method is the ability to construct cavity-sensitive and hole-free models.

There are several ways in which the system described in this report could be improved. The easiest options involve software changes. In order to reduce user intervention the binarization of the input silhouette images and the extraction of the range data can be automated. Secondly, texture can be integrated as explained in [2] and [21] to complete the models description, which is relatively easy. Thirdly, although the resultant surface models are huge in size, they can be further simplified for efficient visualization and transmission by employing progressive modeling schemes [3].

The foremost restraining factor in the overall system performance was found to be hardware related. As the obtained results show, the fixed single-striped range scanner used in our system can perform inadequately in delivering well distributed range samples. This defect is the main cause for the uncarved sections apparent in some of the reconstructions. This, in effect, also restricts the use of higher resolutions of the octree than the ones conducted in the experiments, since the higher the resolution of the octree the smaller the

cubes get and ultimately, the greater the number of cubes that are going to remain uncarved. It was observed that using range scanners that produce full range images as discussed in Section 5 can greatly enhance the distribution of the range data and permit carving at higher octree resolutions, which would produce even sharper results and maintain the object's correct shape.

Changes in the hardware would also allow a more accurate detection of range data by the use of better triangulation schemes such as spacetime analysis [33]. Range scanning errors on curved and discontinuous surfaces may thus be mostly prevented. Moreover, using multiple cameras would both alleviate the range scanner problem and allow the scanning of objects scaled larger than the ones experimented on so far. Finally, real time processing of the acquired range data can help detect the uncarved sections of the object volume and aid in next view planning. Such a scheme would call for an efficient way to display the intermediate results which could be handled with techniques utilizing low cost primitives. Splatting [32], is a good candidate for this task and has recently started to gain interest.

BIBLIOGRAPHY

[1] A. Y. Mülayim, Y. Yemez, F. Schmitt and V. Atalay, Rotation Axis Extraction of a Turntable Viewed by a Fixed Camera, Vision Modelling and Visualization VMV'99, Germany, Erlangen, (1999).

[2] Y. Yemez, and F. Schmitt, 3D Reconstruction of Real Objects with High Resolution Shape and Texture, Image and Vision Computing, (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, (2002).

[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] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, Decimation of Triangle Meshes, ACM Computer Graphics, SIGGRAPH '92 Proc., Vol. 26, (1992), 65-70.

[10] B. Curless and M. Levoy, A Volumetric Method for Building Complex Models from Range Images, SIGGRAPH '96.

[11] S. Rusinkiewicz, O. Hall-Holt, M. Levoy, Real-Time 3D Model Acquisition, SIGGRAPH '02, (2002), http://www-graphics.stanford.edu/papers/rt_model/.

[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] M. Proesmans, L. Van Baar, and F. Defoort, Reading Between the Lines – A Method for Extracting Dynamic 3D Texture, Proc. ICCV, (1998).

[17] K. L. Boyer and A. C. Kak, Color-Encoded Structured Light for Rapid Active Ranging, Trans. PAMI, Vol. 9, No. 1.

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

[19] C. Loop, Smooth Subdivision Surfaces Based on Triangles, Master's Thesis, Dept. of Mathematics, University of Utah, (1987).

[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 3DIM'03, (2003), Vol. 96, No. 3, 367-392.

[22] M. Kaas, A. Witkin, D. Terzopoulos, Snakes: Active Contour Models, International Journal of Computer Vision 1, (1988), 321-332.

[23] J. A. Sethian, Level Set Methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science (2nd ed.). Cambridge University Press, (1999).

[24] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares, Efficient Implementation of Marching Cubes' Cases with Topological Guarantees, Journal of Graphics Tools, (2003), 1-15.

[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] R. Y. Tsai, A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lens, IEEE J. Robotics Automation, Vol. RA-3, No. 4, (1987), 323-344.

[29] J. Bouguet, Camera Calibration Toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/calib_doc/

[30] Kaidan, Feasterville, PA 19053, http://www.kaidan.com

[31] N. Dyn, J. A. Gregory, and D. Levin, A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control, ACM Trans. on Graphics 9 (1990), 160-169.

[32] S. Rusinkiewicz and M. Levoy, QSplat: A Multiresolution Point Rendering System for Large Meshes, Proceedings of ACM SIGGRAPH'00, (2000).

[33] B. Curless, M. Levoy, Better Optical Triangulation through Spacetime Analysis, IEEE Int. Conference on Computer Vision '95, (1995), 987-994.

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

[35] StockerYale Inc, Salem, NH 03079, http://www.stockeryale.com

[36] H. H. Chen and T. S. Huang, A Survey of Construction and Manipulation of Octrees, Comp. Vision, Graphics and Image Process., Vol. 43, No. 3, (1988), 409-431.

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

[38] A. Bottino, A. Laurentini, The Visual Hull of Piecewise Smooth Objects, Proceedings of British Machine Vision Conference 2004, Kingston-upon-Thames (UK), (2004).

[39] P. Fua and Y. Leclerc, Object-centered Surface Reconstruction: Combining Multi-image Stereo and Shading, International Journal of Computer Vision 16, (1995), 35-56.

[40] Qilong Zhang and Robert Pless, Extrinsic Calibration of a Camera and Laser Range Finder, Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), (2004).

VITA

Can James Wetherilt was born in Swindon, England on May 14, 1981. He graduated from Burak Bora Anadolu Lisesi, Istanbul in 1998. He received his B.S. degree in Computer Engineering from Marmara University, Istanbul in 2002. In September 2002, he joined the M.S. Program in Electrical and Computer Engineering at Koç University, as a research and teaching assistant.