3D RECONSTRUCTION OF CURVED SURFACES FROM SINGLE VIEW

BY

HETAL CHAUHAN

07MCE001

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

MAY 2009

3D Reconstruction of Curved Surfaces from Single View

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

Hetal Chauhan

07MCE001

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May 2009

Certificate

This is to certify that the Major Project entitled "3D Reconstruction of Curved Surfaces from Single View" submitted by Hetal Chauhan (07MCE001), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. S.N. Pradhan
Guide, Professor,
Department Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Prof. D. J. Patel
Professor and Head,
Department of Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Abstract

This project work aims to find 3D model of curved object with surfaces having circular cross section from a single 2D image with minimum human intervention. Problem of generating 3D model is of great interest in many fields like medical science, painting modeling, virtual reality and heritage documentation. Single View Reconstruction appears to be shifting away from Multi view Reconstruction where only single image is available.

Though a single view of a scene does not have enough information about the depth, the problem of 3D Reconstruction can be solved to some extent by imposing constraints or through user interaction.

2D images contain cues about real object. However, their interpretation is ambiguous because depth information is lost when 3D object is projected on 2D image. Multiple images from different viewpoints resolve these ambiguities. However, sometimes it is not possible to obtain such images, and only single photographs or paintings exist or even when working with multiple images, parts of the scene appear in only one image due to occlusions.

Problem of generating 3D surface can be expressed as optimization problem. From Single view Silhouette is extracted. It generates best 3D surface which projects exactly to that silhouette.

Acknowledgements

I am really very grateful to Dr. S. N. Pradhan and to Prof. Swati Jain for sparing their valuable time and providing their useful guidance. They also supported by patiently listening to my search for material and providing their personal opinion which help me a lot.

I am specially thankful to Mukta Prasad, University of Oxford, U.K. for helping me in understanding this project through emails. She helped me a lot in each and every phase of my project where I got confused.

I am also very thankful to other staff members and to my friends for their support.

And at this moment, I would like to express my appreciation to my family members for their unlimited encouragement and support.

Last but not the least, the God, thank you very much for helping me in each and every phase of my life.

- Hetal B. Chauhan

07MCE001

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The aim of this work is to automatically generate the 3D surface of the given 2D image in constrained environment. The increase in demand for 3D content has inspired researchers to devise techniques which allow 3D models to be acquired directly from the images. We are interested in single-view reconstruction: given a single image of a curved object, recover a 3D model with the user input.There should be an automation system whose input is essentially the image, and an ouput is 3D surface of the given image with minimum human intervention.

1.1 Basics of 3-Dimension Views

The reconstruction of 3D objects from 2D images has long been of interest to researchers in computer vision and computer graphics. 2D is 'flat', using the x and y (horizontal and vertical) axis. And if turned to the side, it becomes a line. 3D adds the 'Z' dimension. This third dimension allows for depth. It's essentially the difference between a painting and a model. Because 2D images do not give clear idea about object features like its shape, volume etc., it is necessary to generate its 3D model.

1.2 Motivation

3D model is more valuable rather than 2D image of object in many areas like visualization. When only one view of an object (either real or imaginary) is available, multi-view algorithms can not be applied to construct three-dimensional models of the observed object.

Nowadays cameras of good quality are available. But there are situations where we can not get multiple images, like we have a image of object and we want to visualize it or we have single photograph of heritage and want to document it. Or even when working with multiple images, parts of the scene appear in only one image due to occlusions or lack of features to match between images then need to generate its 3D model from only one image. So, in these kinds of situations, it is necessary to generate 3D surface using only one image

1.3 Scope of Work

The generation of realistic 3D models is of great interest in many fields and has many practical applications. For instance, such 3D models can be used in computerized games, virtual reality, forensic science, Painting Modeling, art history, medical science, architectural drawing, visualization and heritage documentation.

Objective of this study is to reconstruct 3D model from a single image, and in particular to compute a 3D model which projects exactly to the object outline (or apparent contour) in the image. This project work considers object having curved surfaces because real world objects are mainly with curved surfaces

1.4 Outline of Thesis Report

Chapter 2, *Literature Survey*, This chapter describes various techniques used for generation of the 3-Dimensional surface. It also gives an insight details that

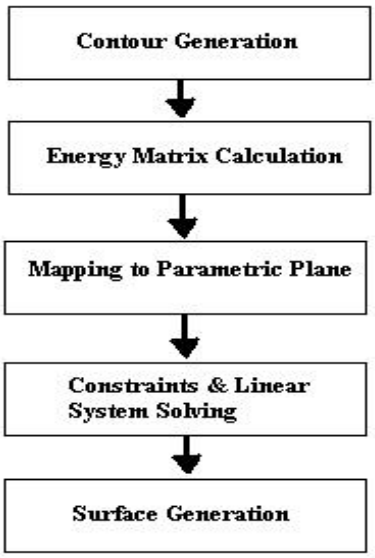

Figure 1.1: Organization Of Work

which technique is useful and why.

Chapter 3, *Problem Definition and Methodology*, This chapter describes the problem that we wish to solve in this study and approach that we want to apply for this problem.

Chapter 4, *Implementation Details*, This chapter describes details of how the proposed methodology can be implemented to find 3-Dimensional mesh surface.

Chapter 5, *Results and Analysis*, This chapter describes output achieved after applying proposed methodology on sample objects and discussion on that output.

Chapter 6, *Conclusion and Future work* ,This chapter describe what the outcome of the study and implementation and also tries to give some future directions.

Chapter 2

Literature Survey

Various techniques have been developed to generate 3D model from 2D images. Some techniques use image features such as texture, shading for generating 3D model, these are known as model based techniques. Constraint-based techniques use geometric properties like co-linearity, coplanar, point, normal etc. The geometric approach deals with geometric relationships between points, lines, planes, etc under imaging. Some techniques share aspects of both constraint and model-based methods [1].

2.1 Approaches for 3D

There are various approaches to find 3D model or reconstructing 3D shape from 2D image. They are classified based on the image features they use to find 3D or technique they apply for 3D reconstruction from 2D image

2.1.1 Stereo Vision

It is technique of generating 3D model from overlapping images like human eyes. Each eye captures its own view and the two separate images are sent on to the brain for processing. When the two images arrive simultaneously in the back of the brain, they are united into one picture. The mind combines the two images by matching

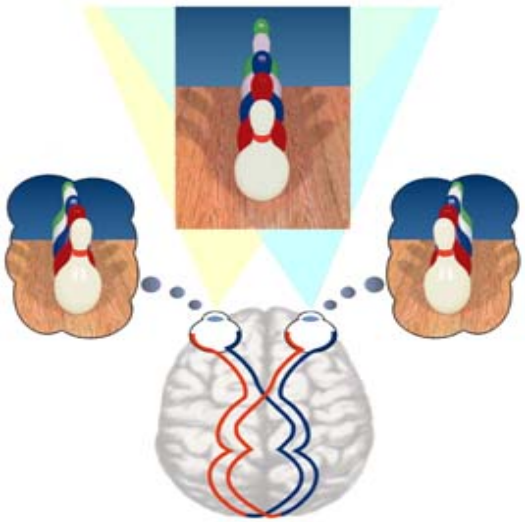

Figure 2.1: Three-dimensional Stereo Picture

up the similarities and adding in the small differences. The small differences between the two images add up to a big difference in the final picture. The combined image is more than the sum of its parts shown in Figure- 2.1

2.1.2 Structure from Motion

Structure from motion refers to the process of finding the three-dimensional structure by analyzing the motion of an object over time. Finding structure from motion presents a similar problem as finding structure from stereo vision. In both instances, the correspondence between images and the reconstruction of 3D object needs to be found. To find correspondence between images, features such as corner points need to be tracked from one image to the next. The feature tracked over time are then used to reconstruct their 3D positions and the cameras motion. Motion of object over time is shown in Figure-2.2.

Figure 2.2: Motion of Object Over Time

2.1.3 Structure from Focus

Sequence of images of the studied object is taken step by step with a constant change of the focusing distance between consecutive images. The composed image is then calculated from this sequence by combining the sharp regions of the raw images. The basic idea of image focus is that objects at different distances from a lens are focused at different distances. Shape from Focus (SFF) is the problem of reconstructing the depth of the scene changing actively the optics of the camera until the point of interest is in focus. The change in the optics is obtained by changing either the lens position or the object position relative to the camera. A focus measure is computed in the small image regions of each of the image frame in the image sequence. The value of the focus measure increases as the image sharpness or contrast increases and it attains the maximum for the sharpest focused image. Thus the sharpest focused image regions can be detected and extracted. This facilitates auto-focusing of small image regions by adjusting the camera parameters (lens position and/or focal length) so that the focus measure attains its maximum value for that image region. Also, such focused image regions can be synthesized to obtain a large image where all image regions are

in focus. Further, the distance or depth of object surface patches that correspond to the small image regions can be obtained from the knowledge of the lens position and the focal length that result in the sharpest focused images of the surface patches. Focusing difference in the same image is shown in Figure- 2.3

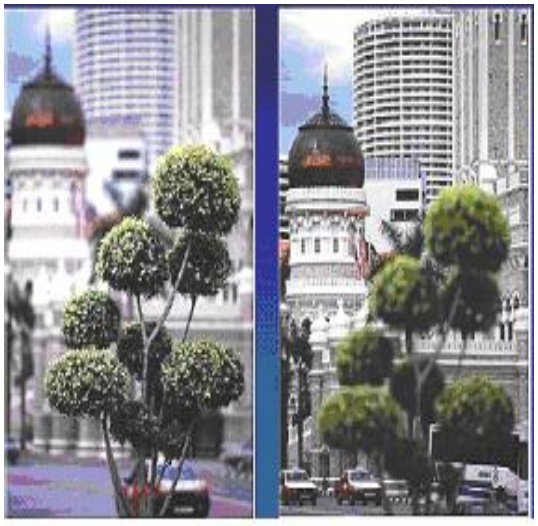

Figure 2.3: Focusing Difference

2.1.4 Shape from Silhouette

It uses image outline of object for 3D generation. Silhouette is the dominant source of shape of object. A silhouette is a view of an object or scene consisting of the outline and a featureless interior, with the silhouetted object usually being black. It can use more than one images to extract silhouette of object. From that silhouette various constraints are given like point constraints, normal constraints, depth constraints, and optimized 3D model can be generated. Outline of object from which silhouette can be extracted is shown in Figure- 2.4

Figure 2.4: Aeroplane with its silhouette

2.2 Approaches for 3D from Single View

2.2.1 Using User Defined Vanishing Point

First Single View Reconstruction(SVR)system proposed by Horry [2] in 1997. In that system, GUI is provided to user for specifying vanishing point.3D geometry of the scene's background is defined as model centering user defined vanishing point. System models the 3D background of the scene using polygons. Foreground objects stand on polygons of the background model.

Vanishing Point

Parallel lines in the world appear to meet at a single point in the image. This point is known as the vanishing point corresponding to the direction of those parallel lines, and it is the image of a point at infinity at which those parallel lines "intersect".

To give a sense of depth vanishing point is used. As shown in Figure- 2.5, by making parallel lines in the ceiling and on the wall converge to a vanishing point.

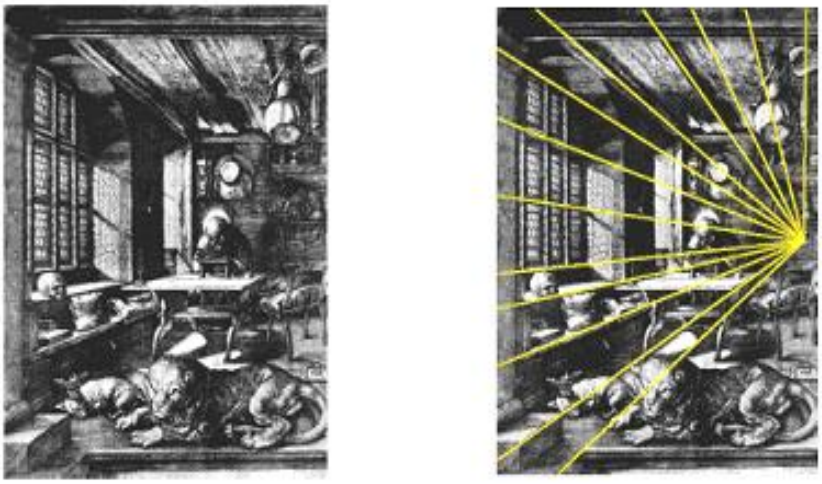

Figure 2.5: Vanishing point

3D Modeling

The main idea of the proposed method by Horry [2] is simply to provide a user interface which allows the user to easily and interactively perform the following operations.

a. Adding "virtual" vanishing points for the scene - The specification of the vanishing point should be done by the user, not automatically

b. Distinguishing foreground objects from background - The decision as to whether an object in the scene is near the viewer should be made by the user

Background Modeling

In order to approximate the geometry of the background scene, several polygons should be generated to represent the background with the vanishing point being on its base. From vanishing point defined by user five 2D rectangles may be deduced as shown in Figure- 2.6 and the rectangles are tentatively called the floor, right wall, left wall, rear wall, and ceiling, respectively. It defines the textures of these 2D rectangles to be taken from the background image.

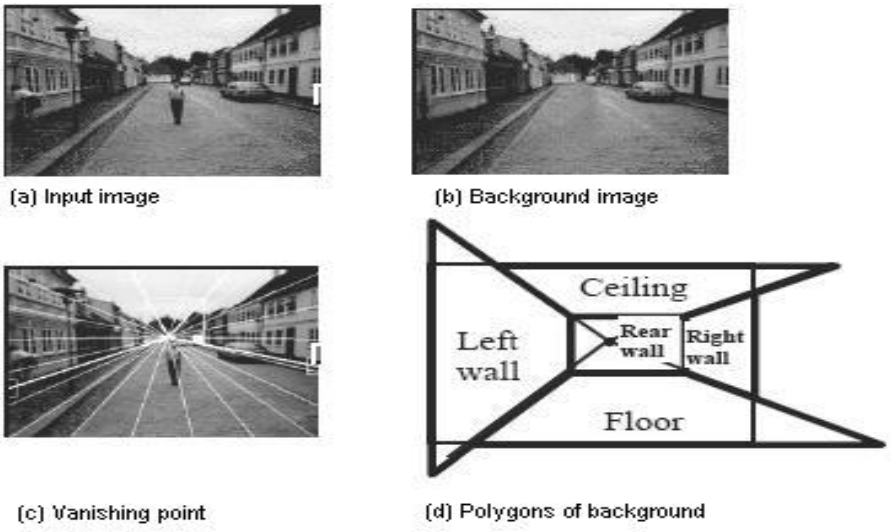

Figure 2.6: 3D Background Modeling

It then defines the 3D background model in 3D space as being these five 3D rectangles, assuming that the following conditions hold:

- Every adjacent 3D rectangle mentioned above is orthogonal to the others
- The 3D rear wall is parallel to the view plane
- The 3D floor is orthogonal to the view up vector
- The textures of the 3D rectangles are inherited from those of the corresponding 2D rectangles

Foreground Modeling

Based on the foreground mask information provided by user, 3D polygonal model for foreground objects in the scene is created. To surround the 2D image of a foreground object, 2D quadrangle in the input image is specified as shown in Figure- 2.7. 3D position of the quadrangle in the 3D background model is derived under the condition that the quadrangle should be perpendicularly put on one of the five 3D regions: floor,

right wall, left wall, rear wall, and ceiling. In the example of figure 2.7, the person is a foreground object to be modeled, and is surrounded by the quadrangle. The quadrangle in the 3D scene is perpendicularly attached to the 3D floor.

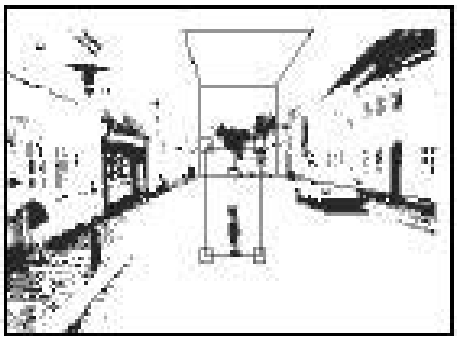

Figure 2.7: Specifying Foreground Object

Foreground object model is hierarchy among the polygons.

- Each model consists of one or more polygons. In particular a single polygon itself is a foreground object model
- For any polygon F1 belonging to the model, another polygon F2 can be added to the model, if F2 is orthogonally attached to F1 so that one side of F2 is on F1. Then F2 is called a child of F1 (or F1 is a parent of F2). This constitutes a hierarchy among the polygons belonging to a foreground object model
- If a polygon of the model is at the highest level in the hierarchy, it is orthogonally attached to one of the five 3D regions of the 3D background. Then only one side of the highest level is only on the region

Figure 2.8 illustrates how to construct the foreground object models. First, two

quadrangles F0 and F1 are defined on the 3D floor. Then F 2 is added to F 1 and F 3 is added to F 2.

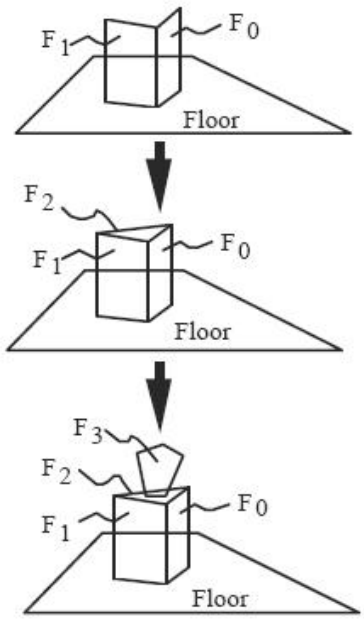

Figure 2.8: Hierarchical Positioning of the Foreground Objects

2.2.2 Vanishing Point Automation

Subsequent systems [3],[4] improved the SVR System. It addresses the problem of extracting three-dimensional geometric information from a single image of a scene. But here vanishing point is not defined by user but automatically estimated by system as shown in Figure 2.9

Steps for Vanishing point Estimation

a. Automatic Canny edge detection and straight line fitting to obtain the set of straight edge segments E

b. Repeat

- Randomly select two segments s1, s2 from edge segments E and intersect them to give the point p;
- The support set Sp is the set of straight edges in E going through the point p;

c. Set the dominant vanishing point with the largest support Sp as the vanishing point

d. Remove all edges in Sp from E and go to 2 for the computation of the next vanishing point

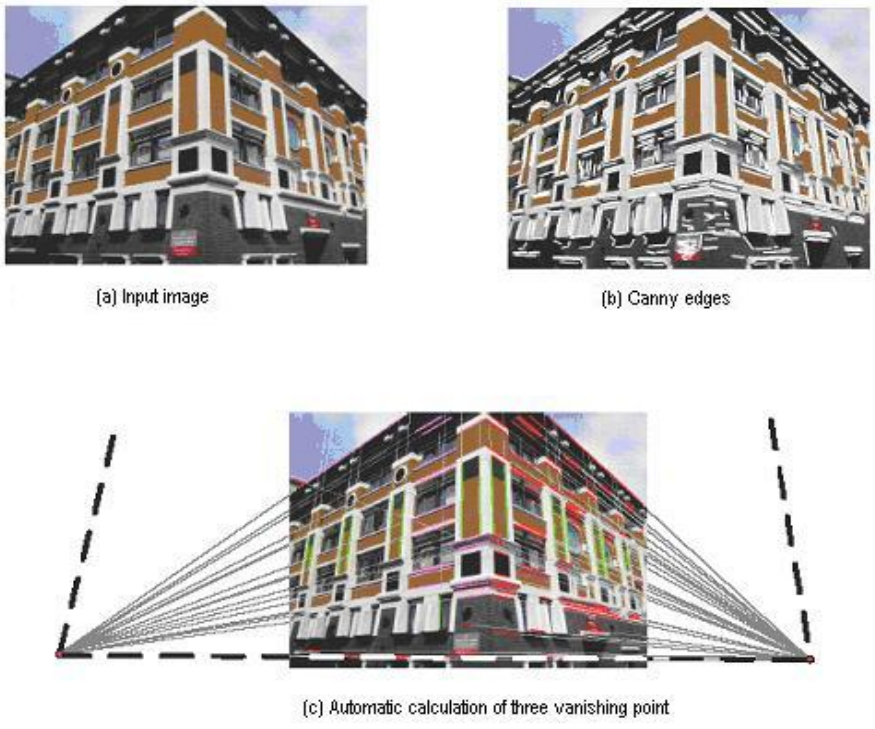

Figure 2.9: Estimating Vanishing Point

2.2.3 3D FOR curved surfaces

Above techniques are restricted to planar surfaces. They do not permit for the object having curvilinear shape

2.5 Dimensional Reconstruction

Given a set of user-specified constraints on the shape of the scene, a smooth surface that satisfies the constraints is generated [5]. This problem is formulated as a constrained optimization problem. Measure of surface smoothness is thin plate energy function.
Grapchics interface is provided for user-specified constraints like surface positions, normals creases and discontinuity. As each constraint is specified, the system recalculates and displays the reconstruction in real time

a. Point constraints specify the position or the surface normal of any point on the surface

b. Surface discontinuity constraints is a curve across which surface depth is not continuous, creating a tear in the surface

c. Crease constraints specify curves across which surface normals are not continuous

d. Planar region constraints determine surface points that lie on the same plane

e. Fairing curve constraints allow users to control the smoothness of the surface along any curve in the image

Using given constraints and thin plate function as objective function linear system is formed and iterative optimization technique is used to find solution of linear system. In this approach, subset of surface that is visible from single image is modeled and represented as f(u,v,z(u,v)). So generated surface is 2.5D rather than 3D, because it represents just one side of 3D model.

Three Dimensional Reconstruction

Given a single photo of a curved object, recover 3D model with minimum of user input which projects to object silhouette[6]. Like former approach[5], smoothest surface satisfying user defined constraints is generated. But in this approach, generated 3D surface projects to object silhouette. 3D Surface is represented as $r(u,v) = [x(u,v), y(u,v), z(u,v)]^T$ where u,v are value between 0 to 1.

Energy Function: Surface smoothness is defined in terms of an energy on the surface, and it is thin plate energy

$$E(r) = \int_0^1 \int_0^1 ||r_{uu}||^2 + 2||r_{uv}||^2 + ||r_{vv}||^2 dudv \tag{2.1}$$

Constraints: Various kind of constraints are defined at (u,v) points.

a. Position constraints are of the form $r(u_k, v_k) = p_k$ for known values of u_k, v_k, p_k

b. Normal constraints require the surface normal at (u_k, v_k) to equal a supplied normal n_k. The normal to r at a point is the unit vector along $r_u * r_v$. they are of the form

   1 $n_k \cdot r_u(uk, vk) = 0$

   2 $n_k \cdot r_v(uk, vk) = 0$

c. Partial position constraints act on just one component of r, for example $z(uk, vk) = zk$ constrains only the z coordinate at (uk, vk)

d. Silhouette constraint forces that contour generator of the 3D surface must project to object silhouette in given 2D image of that object.Contour generator is the set of 3D points on the surface at which the viewing direction is in the tangent plane.

e. Inflation constraints provides constraints for z values.Wihout inflation,generated surface is 2D surface and it projects to given silhouette for which z(u,v)=0 for all u,v

This approach represents surface as $r(u, v) = [x(u, v), y(u, v), z(u, v)]^T$ and finds depth at (x,y) pixel using mapping between (x,y) and (u,v) rather than representing surface as $r(u, v) = [u, v, z(u, v)]$ like 2.5D reconstruction [5].

Chapter 3

Problem Definition and Methodology

Given single image of curved object with special class of curved surfaces, surfaces of revolution, 3D mesh surface with minimum user intervention is generated. Surface of revolution is the surface generated by rotating two-dimensional curves about an axis. The resulting surface therefore always has symmetry. 3D surface is generated using object silhouette. The problem of 3D Surface generation is considered as constrained optimization problem [6].

3.1 Problem Definition

Our aim in this work is generating 3D mesh surface by revolving curves taken from single 2D image of curved object. **Single View Reconstruction:** When any object is projected as an image, its depth information is lost. So it is difficult to interpret shape of object from given image of object. Reconstruction is the process of computing the three dimensional shape of an object from its images. In single view Reconstruction, data is minimal that only single image of that object. **Surface of Revolution:** It is the special class of curved surfaces, which is generated by rotating 2D curve about axis [7]. Examples of surfaces generated by rotating curves are

cylindrical, conical, toroidal, spherical surfaces etc. A cylinder is shape derived by rotating two parallel lines as shown in Figure- 3.1. A sphere is defined as the set of all points that are located at constant distance from center as shown in Figure- 3.2.

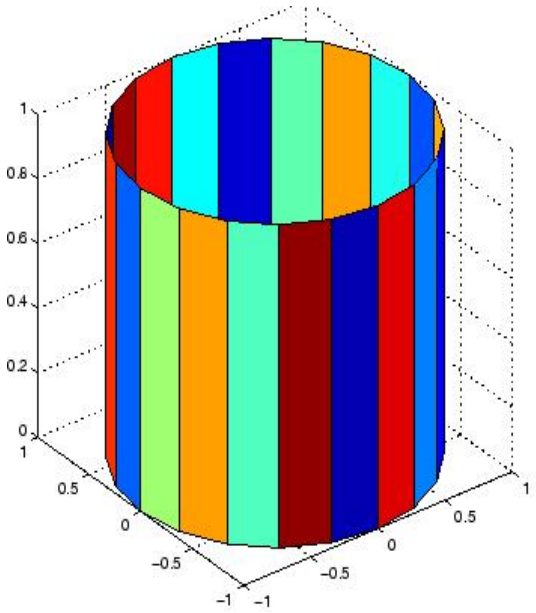

Figure 3.1: Cylinder

3.2 Methodology

we try to find 3D surface using object silhouette. So user defined silhouette need to be extracted from given image. 3D surface is generated which projects exactly to this silhouette. **Silhouettes** (alternatively referred to as apparent contour, occluding contour, profile or outline) are often a dominant image feature, and can be extracted relatively easily . They provide rich information about the shape of an object, and are indeed the only information available in the case of smooth texture less surfaces[8]. Optical rays which are tangent to the surface from the viewing direction touch the surface along a smooth curve known as the **contour generator** shown in Figure-3.3. The contour generator separates the visible part from the occluded part of the

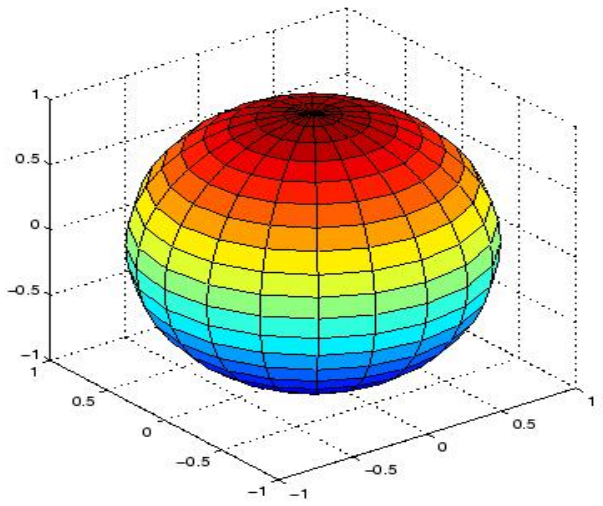

Figure 3.2: Sphere

object

The silhouette is the projection of contour generator on image plane as shown in Figure- 3.4. In contrast to the features arising from corners, edges and surface markings, which are viewpoint independent, silhouettes are inherently viewpoint dependent. In general, two silhouettes of an arbitrary smooth object observed from two distinct viewpoints are the projections of two distinct curves

3.2.1 Energy Minimizing surface

The 3D model is obtained by minimizing a surface smoothness objective function subject to constraints from the apparent contour[9].

Surface smoothness is measured using **bending energy**. It is also known as the Willmore energy. The bending energy measures the total curvature of a surface. Intuitively, this corresponds to the work required to deform a surface from a homogenous flat sheet (which has zero bending energy) to its current shape[10].

There are various ways of representing surface like function of two vaiables such as f(x,y,(x,y)) using implicit notation or three variables like f(x,y,z) as explicit notation.

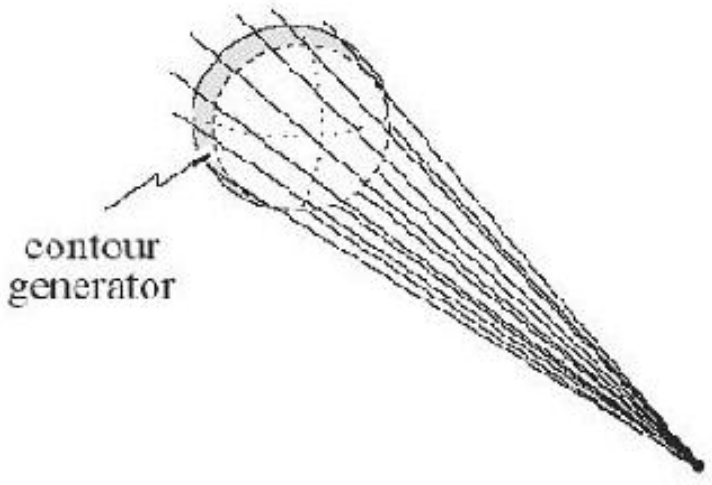

Figure 3.3: Contour Generator

But in this kind of representation, controling each and every point of surface is difficult.

Here Surface is represented using **parametric representation**. It is the most general way to specify a surface in various surface optimization problems. Parametric surface be given by the equation $\vec{r} = r((\vec{u}, v)$ where $\vec{r}$ is a vector-valued function of the parameters (u, v) and the parameters vary within a certain domain D in the parametric uv-plane [11]. Advantages of parametric surface reprentation are, it is easy to enumerate points on surface and best for freeform surfaces . So, using parametric representation, 3D surface is denoted as $r(u, v) = x(u, v), y(u, v), z(u, v)^T$ and domain for u and v is from 0 to 1.

Bending energy function using (u,v) representation is,

$$E(r) = \int_0^1 \int_0^1 ||r_{uu}||^2 + 2||r_{uv}||^2 + ||r_{vv}||^2 dudv \tag{3.1}$$

Here r_{uu} is the second order partial derivative of surface with respect to u ,r_{vv} is second order partial derivative of surface with respect to v and r_{uv} is second order partial derivative of surface with respect to u and then v. This energy can be optimized

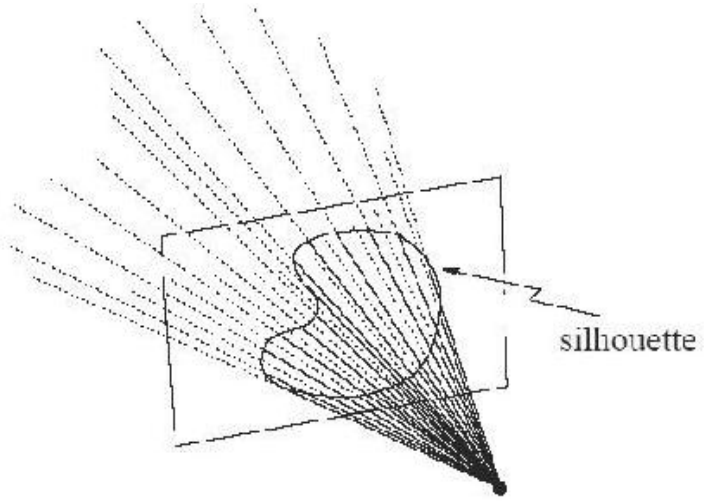

Figure 3.4: Apparent Contour

using gridded discretization means representing r by three $m * n$ matrices, X, Y, Z. Stack these matrices into a single vector $g = [x^T y^T z^T]^T$ where lowercase x represents the reshaping of X into a column vector. Central difference approximations are used for the first and second derivatives on the surface [9]. Thus the first derivative term x_u at a point (i, j) is discretely approximated as

$$X_u(i,j) = X(i+1,j) - X(i-1,j) \tag{3.2}$$

and matrix C_u is computed as, $X_u = C_u$ x. Same way second order derivatives are computed and denoted as C_{uu}, C_{uv} and C_{vv}. In terms of these, the bending energy Equation 3.1 , can be expressed in discrete form as a quadratic function of g as,

$$\varepsilon(x) = x^T(C_{uu}^T C_{uu} + 2C_{uv}^T C_{uv} + C_{vv}^T C_{vv})x \tag{3.3}$$

$$E(g) = \varepsilon(x) + \varepsilon(y) + \varepsilon(z) = g^T C g \tag{3.4}$$

where C is energy matrix. If we take X,Y and Z matrix of size $m * n$ then size of matrix C should be $3mn * 3mn$

3.2.2 Apparent Contour Constraints

Apparent contour is the image outline of a smooth surface S results from surface points at which the imaging rays are tangent to the surface. So, it is 2D curve. The contour generator (CG) is the set of points on S at which rays from the viewer are tangent to the surface and denoted as $c(t)|0 \leq t \leq 1$. Here t ranges in the domain of (u,v) as shown in Figure- 3.5. The image of the contour generator is called the apparent contour, and is the set of points s which are the image of contour generator denoted as $s(t)|0 \leq t \leq 1$ assuming it is parameterized by the same t as the contour generator

Contour Constraint is,

- The candidate 3D model must project to the 2D apparent contour.

Simplest of the parallel projections is the **Orthographic projection**, commonly used for engineering drawings. It accurately show the correct or 'true' size and shape of an object.[12]

The matrix for orthographic projection onto z=0 plane is, $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$

Assuming orthographic projection along z axis, In terms of (u,v) points at surface, constraint is,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} r(u(t), v(t)) = s(t) \tag{3.5}$$

Consider the left contour l, where $(s_j, t_j)$are the point co-ordinates of the apparent contour where j varies from 1 to n points . So,in the discrete setting, the constraints in Equation 3.5 translate to the following equations

$$X(i_l, j) = sj, \;\; where \;\; j = 1..n \tag{3.6}$$

$$Y(i_l, j) = tj, \;\; where \;\; j = 1..n \tag{3.7}$$

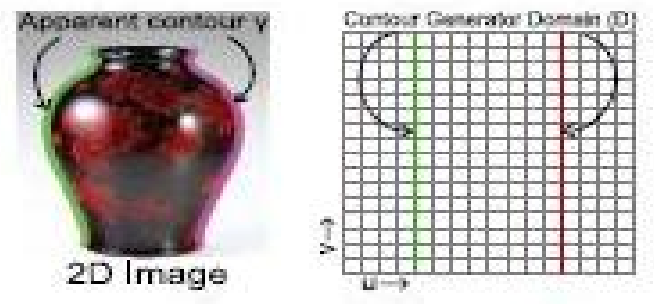

Figure 3.5: Parameterization of Contour Generator

This set of equations can be created for each contour. Putting these equations together, we get a linear system of equations of the form Ag = b. Now the minimization of Equation 3.4 can be re-written as the following Lagrange

$$L = g^T C g + \lambda.(Ag^T - b) \tag{3.8}$$

3.2.3 Inflation Constraints

Apparent contour constraints work only on x and y co-ordinates. Thus on solving the system, without inflation constraint the result is a flat surface which projects to the given image, but does not have depth. In other words, it gives the trivial solution z(u, v) = 0 for all u,v

To avoid the trivial solution, we need to inflate the surface. This is done by specifying constraints for z. They are of the form $r(u_k, v_k) = r_k$ and it will give additional terms in Equation3.8

3.2.4 Linear System

Linear system is formed with bending energy function as objective function subject to apparent contour point constraints and inflation constraints. Linear system is solved using numerical method to find vector g that includes X,Y,Z coordinates. So, 3D mesh surface is generated using X,Y and Z matrices. Using Lagrangian, Linear System having the form

$$L = g^T C g + \lambda^T (Ag - b) \tag{3.9}$$

And, it is optimized as,

$$[C \;\; A^T; A \;\; 0][g \;\; \lambda] = [0; b] \tag{3.10}$$

Above equation is solved to find vector g, from which X, Y and Z matrices are retrieved. Surface is generated using that X, Y and Z matrices.

Chapter 4

Implementation Details

This work aims to generate 3D mesh surface which projects to user specified apparent contour of the curved object having circular cross section. The 3D surface r : $[0, 1]^2 \rightarrow R^3$, is as smooth as possible while obeying user-specified constraints [9]. I have used MATLAB to implement proposed methodology.

4.1 Steps

a. From given image of curved object, its contours are extracted .

b. Using bending energy equation, energy matrix is calculated.

c. Contours are mapped to parametric plane (u,v).

d. Contour point constraints are given.

e. Inflation constraints are given.

f. Form linear system and get x,y,z coordinates by solving it.

g. 3D mesh surface is generated using x,y and z coordinates.

4.2 Apparent Contour

The approach used to retrieve contour points is suitable for simple images having clear distinction between object and its background without any kind of reflection. First object of interest from given image need to be extracted. I have applied following steps to find object of interest from image.

- Convert given image into binary image
- Convert binary image into edge image
- Find connected components from image using pixel connectivity, it will identify objects in given image.

For the given image of flower vase after applying above steps, extracted object is shown in Figure 4.1

After object extraction, contours are retrieved. Following steps are applied to find contours for cylinder topology.

- Find y value for topmost and bottommost points of object and say it yt, yb respectively.
- Find x value for leftmost,rightmost and middle points of object and say it xl, xr and xm respectively.
- Starting from yt to yb, retrieve all the points between xl and xm into array which are the points for left contour.
- Starting from yt to yb, retrieve all the points between xm and xr into another array which are the points for right contour.

For flower vase, contours which can be used to generate its 3D shape is both of its sides from boundary. So, pixel coordinates of both sides have been retrieved as shown in Figure 4.2

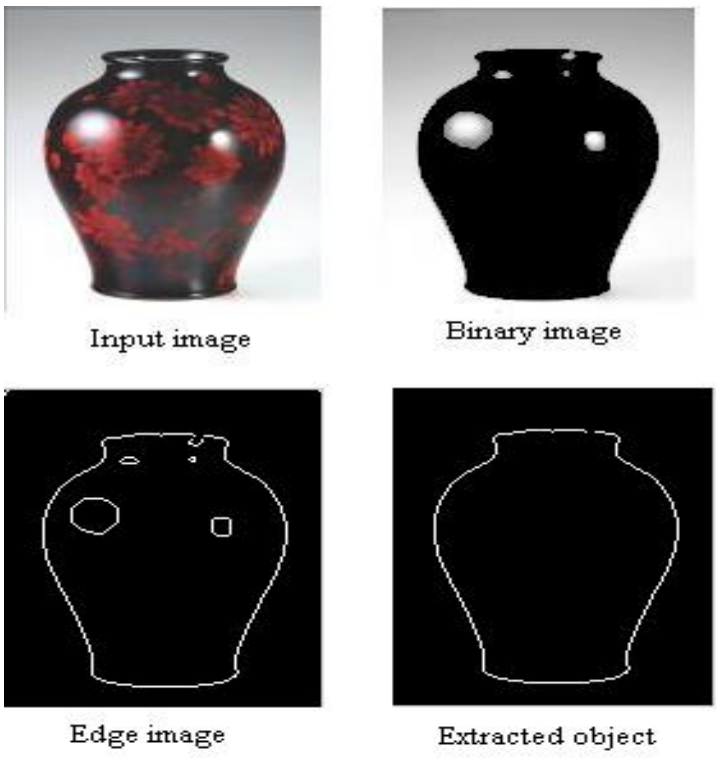

Figure 4.1: Object Extraction for flower vase

4.3 Energy Matrix

We try to generate smoothest surface while satisfying silhouette and inflation constraints. We use bending energy as a measure of surface Smoothness. So, objective is minimizing bending energy. Bending energy is,

$$E(r) = \int_0^1 \int_0^1 ||r_{uu}||^2 + 2||r_{uv}||^2 + ||r_{vv}||^2 dudv \tag{4.1}$$

4.3.1 Energy Matrix Discretization

To optimize this, we have to discretize the surface, means surface should be represented as three matrices X,Y, and Z of $m * n.$ size. So, in this discrete form,bending

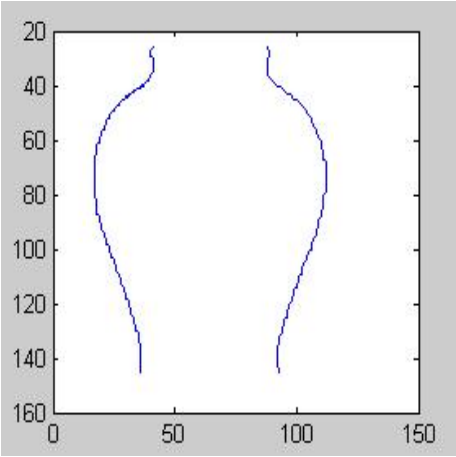

Figure 4.2: Contours for flower vase

energy is,

$$\varepsilon(x) = x^T(C_{uu}^T C_{uu} + 2C_{uv}^T C_{uv} + C_{vv}^T C_{vv})x = x^T \cdot c1 \cdot x \tag{4.2}$$

$$\varepsilon(y) = y^T(C_{uu}^T C_{uu} + 2C_{uv}^T C_{uv} + C_{vv}^T C_{vv})y = y^T \cdot c2 \cdot y \tag{4.3}$$

$$\varepsilon(z) = z^T(C_{uu}^T C_{uu} + 2C_{uv}^T C_{uv} + C_{vv}^T C_{vv})z = z^T \cdot c3 \cdot z \tag{4.4}$$

$$E(g) = \varepsilon(x) + \varepsilon(y) + \varepsilon(z) = g^T C g \tag{4.5}$$

Here, x, y, z vectors store the values of the points on the mesh discretizing the object. The vecotr g is basically a concatenation of x, y and z.
So, Equation 4.5 can be represented as

$$E(g) = [XYZ]^T C [XYZ] \tag{4.6}$$

So, energy matrix C is formed by concatenating matrices c1,c2 and c3 which are energy matrices in x,y and z directions respectively. Matrices c1,c2 and c3 must be formed accordingly where u and v represents index of the matrices.

So, Equation 4.6 can be simplified as

$$E(g) = [XYZ]^T[C1 \cdot C2 \cdot C3][XYZ] \tag{4.7}$$

Using Equations 4.2, 4.3 and 4.4, we can say that

$$C1 = C2 = C3 = C_{uu}^T C_{uu} + 2C_{uv}^T C_{uv} + C_{vv}^T C_{vv} \tag{4.8}$$

4.3.2 Energy Matrix Formation

Observing Equation 4.8, it is clear that energy matrix is same for all three dimensions x,y and z. So, to calculate matrix C, we have to find matrix using relation $C_{uu}^T C_{uu} + 2C_{uv}^T C_{uv} + C_{vv}^T C_{vv}$ and concatenate it two more times.

Matrix C_{vv}

C_{vv} is second order partial derivation with respect to v. We use gradient square to find partial derivation. Gradient square is,

$$X_v(i,j) = (X(i,j+1) - X(i,j+1))^2 \tag{4.9}$$

First order partial derivation with respect to v can be derived using following relation.

$$x_v = C_v \cdot x \tag{4.10}$$

where x_v and x is column vector of X_v and X respectively.
Similarly, second order derivation can be calculated as,

$$x_{vv} = C_{vv} \cdot x \tag{4.11}$$

and

$$X_{vv}(i,j) = X(i,j+1) + X(i,j-1) - 2X(i,j) \tag{4.12}$$

For, m=n=3, then using Equation 4.12,

X_{vv}(1,1)=X(1,2)+X(1,0)-2X(1,1)

X_{vv}(1,2)=X(1,3)+X(1,1)-2X(1,2)

X_{vv}(1,3)=X(1,4)+X(1,2)-2X(1,3)

X_{vv}(2,1)=X(2,2)+X(2,0)-2X(2,1)

X_{vv}(2,2)=X(2,3)+X(2,1)-2X(2,2)

X_{vv}(2,3)=X(2,4)+X(2,2)-2X(2,3)

X_{vv}(3,1)=X(3,2)+X(3,0)-2X(3,1)

X_{vv}(3,2)=X(3,3)+X(3,1)-2X(3,2)

X_{vv}(3,3)=X(3,4)+X(3,2)-2X(3,3)

Now using relation 4.11, Matrix C_{vv} is represented as shown in Table- I

(i,j)	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
(1,1)	-2	1	1	0	0	0	0	0	0
(1,2)	1	-2	1	0	0	0	0	0	0
(1,3)	1	1	-2	0	0	0	0	0	0
(2,1)	0	0	0	-2	1	1	0	0	0
(2,2)	0	0	0	1	-2	1	0	0	0
(2,3)	0	0	0	1	1	-2	0	0	0
(3,1)	0	0	0	0	0	0	-2	1	1
(3,2)	0	0	0	0	0	0	1	-2	1
(3,3)	0	0	0	0	0	0	1	1	-2

Table I: C_{vv} for $3 * 3$ matrix for one component

Now, this matrix is of size $mn * mn$ means $9 * 9$ for m=n=3. Concatenating same matrix two more times, resultant matrix is of size, 3mn*3mn, means $27 * 27$. It is as shown as Figure 4.6

Matrix C_{uu}

Same way C_{uu} means second order partial derivation with respect to u. So, all above Equations from 4.9 to 4.12 can be modified for index j rather than for i.

$$X_u(i,j) = (X(i+1,j) - X(i-1,j))^2 \tag{4.13}$$

$$x_u = C_u \cdot x \tag{4.14}$$

where x_u and x is column vector of X_u and X respectively.

$$x_{uu} = C_{uu} \cdot x \tag{4.15}$$

where x_{uu} is,

$$X_{uu}(i,j) = X(i+1,j) + X(i-1,j) - 2X(i,j) \tag{4.16}$$

For, m=n=3, then using Equation 4.16,

X_{uu}(1,1)=X(2,1)+X(0,1)-2X(1,1)

X_{uu}(1,2)=X(2,2)+X(0,2)-2X(1,2)

X_{uu}(1,3)=X(2,3)+X(0,3)-2X(1,3)

X_{uu}(2,1)=X(3,1)+X(1,1)-2X(2,1)

X_{uu}(2,2)=X(3,2)+X(1,2)-2X(2,2)

X_{uu}(2,3)=X(3,3)+X(1,3)-2X(2,3)

X_{uu}(3,1)=X(4,1)+X(2,1)-2X(3,1)

X_{uu}(3,2)=X(4,2)+X(2,2)-2X(3,2)

X_{uu}(3,3)=X(4,3)+X(2,3)-2X(3,3)

Now using relation 4.15, Matrix C_{uu} is represented as shown in Table- II.

Now, again this matrix is of size $mn * mn$ means $9 * 9$ for m=n=3. Concatenating same matrix two more times, resultant matrix is of size, 3mn*3mn, means $27 * 27$. It is as shown as Figure 4.7

(r,c)	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
(1,1)	0	0	0	0	0	0	0	0	0
(1,2)	0	0	0	0	0	0	0	0	0
(1,3)	0	0	0	0	0	0	0	0	0
(2,1)	1	0	0	-2	0	0	1	0	0
(2,2)	0	1	0	0	-2	0	0	1	0
(2,3)	0	0	1	0	0	-2	0	0	1
(3,1)	0	0	0	0	0	0	0	0	0
(3,2)	0	0	0	0	0	0	0	0	0
(3,3)	0	0	0	0	0	0	0	0	0

Table II: C_{uu} for $3 * 3$ matrix for one component

Matrix C_{uv}

C_{uv} means partial derivation with respect to u first and then with respect to v. And,

$$x_{uv} = C_{uv} \cdot x \tag{4.17}$$

where x_{uv} and x is column vector of X_{uv} and X respectively.

We have $X_u(i,j) = (X(i+1,j) - X(i-1,j))$ and
$X_v(i,j) = (X(i,j+1) - X(i,j+1))$.
Using these, X_{uv} is,

$$X_{uv}(i,j) = X(i+1,j+1) - X(i+1,j-1) - (X(i-1,j+1) - X(i-1,j-1)) \tag{4.18}$$

and simplifying it,

$$X_{uv}(i,j) = X(i+1,j+1) - X(i+1,j-1) - X(i-1,j+1) + X(i-1,j-1) \tag{4.19}$$

Using above relation, matrix C_{uv} is represented in Table- III. and after concatenating for all three components ,it is as shown in Figure 4.8

(i,j)	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
(1,1)	1	-1	0	-1	1	0	0	0	0
(1,2)	0	1	-1	0	-1	1	0	0	0
(1,3)	-1	0	1	1	0	-1	0	0	0
(2,1)	0	0	0	1	-1	0	-1	1	0
(2,2)	0	0	0	0	1	-1	0	-1	1
(2,3)	0	0	0	-1	0	1	1	0	-1
(3,1)	-1	1	0	0	0	0	1	-1	0
(3,2)	0	-1	1	0	0	0	0	1	-1
(3,3)	1	0	-1	0	0	0	-1	0	1

Table III: C_{uv} for $3 * 3$ matrix for one component

Steps for Energy Matrix-C

- Find matrix C_{vv} using relation 4.11
- Find matrix C_{uu} using realtion 4.15
- Find matrix C_{uv} using relation 4.17
- Find matrix C using $C_{uu}^T C_{uu} + 2C_{uv}^T C_{uv} + C_{vv}^T C_{vv}$ relation.

4.3.3 Mapping to Parametric Plane

We use parametric representation of surface. Domain of parametric variables u and v is [0,1]. So, contours should be mapped on this interval. keep u constant and vary v for mapping.

For cylinder topology, we have two contours namely, left and right contours.

- Left contour is mapped on u=1/4 and v=[0,1]
- Right contour is mapped on u=3/4 and v=[0,1]

If m=n=8 then take 8 points from left contour and 8 points from right contour. So, for $8*8$ grid, left contour points are mapped to $i = 8*1/4 = 2$ and j=[1,8]. Same way

right contour points are mapped to $i = 8 * 3/4 = 6$ and j=[1,8]. Contour mapping is shown in Figure 4.3

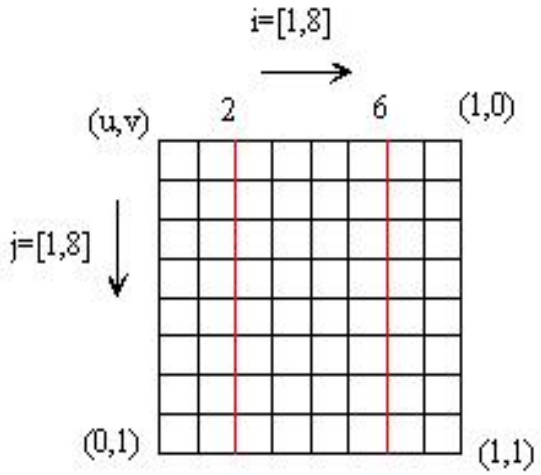

Figure 4.3: Mapping Contours on $8 * 8$ Grid

4.4 Silhouette Point Constraints

Silhouette point Constraint states that 3D model must project silhouette points.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} r(u(t), v(t)) = s(t) \tag{4.20}$$

Consider the left contour l. where (lx_j, ly_j) are the point co-ordinates of the apparent contour where j varies from 1 to n points . So,in the discrete setting, the constraints in Equation 4.20 translate to the following equations

$$X(i_l, j) = lx_j, where j = 1..n \tag{4.21}$$

$$Y(i_l, j) = ly_j, where j = 1..n \tag{4.22}$$

Same way for right contour r, if point coordinates are (rx_j, ry_j),then constraints for right contour is,

$$X(i_r, j) = rx_j, wherej = 1..n \tag{4.23}$$

$$Y(i_r, j) = ry_j, wherej = 1..n \tag{4.24}$$

Here (i_l, j) and (i_r, j) are integer grid coordinates that we have retrieved from mapping done in previous step. Now, left hand side of above equations forms **Matrix A** and right hand side forms **vector b**.

Matrix A

Left hand side of Equations 4.21 to 4.24 represents coefficient of unknowns for mesh surface. If n sample points have been taken form contours, then number of rows in matrix A is $4 * n$. Because 2 equations for 2 contours and n points from each contour. Number of columns of matrix A is $2 * m * n$ if grid is of size $m * n$. Because number of points are mn and we need to find 2 dimensions, x and y for each point.And values for matrix A is coefficients which are 1 only.

If grid is of size $8 * 8$,then number of unknowns for x is 64 and y is also 64. So, first 64 column in matrix A stores x coordinate coefficients, where left contour coordinates starts from 2 and right contour coordinates starts from 6. Then for subsequent coordinates it is incremented by 8. Next 64 columns in matrix A means from 65 to 128, store y coordinates coefficients, where left contour coordinates starts from 2+64=66 and right contour coordinates starts from 6+64=70. Then for subsequent coordinates, it is incremented by 8.

Vector b

Right hand side of Equations 4.21 to 4.24 represents contour point coordinates. So, if n sample points have been taken form contours, then number of rows in matrix b is $4*n$, Because Because 2 equations for 2 contours and n points from each contour. And it is having only one column. Values of b are x and y dimensions of both contours.

For grid of size $8 * 8$, number of rows in vector b is, 32. First 16 rows having x and y coordinates of left contour and next 16 rows having x and y coordinates of right contour.

4.5 Inflation Constraints

Without inflation, flat surface is generated from contour points. To make it 3D following constraint is given for cylinder topology.

$$r(0, v) = r(1, v) \ \ for \ \ all \ \ v. \tag{4.25}$$

It means left and right sides of parametric-(u,v) plane need to be joined as shown in Figure 4.4 [13]. To do so we need to bend (u,v)plane from middle and use radial

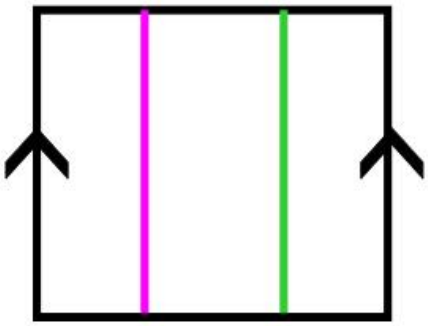

Figure 4.4: Cylinder Topology

distance to find z, as shown in Figure 4.5. Inflation adds terms into both A and b matrices. In matrix A, 2n rows and mn columns are added and values are 1 only that is coefficient for unknown Z. In b vector 2n rows are added with values equal to radial distance that is equal to x dimension.

For grid of size $8 * 8$, it adds z coordinate coefficients in matrix A from 129 to 192 columns, and approximated z coordinates for left and right contours are added in

Figure 4.5: Radial Distance for Inflation

vector b from 33 to 48 rows.

4.6 Linear System and Surface generation

After solving energy function and giving all these constraints, we have A,C matrices and vector b of following size.

Matrix	Number of Rows	Number of Columns
A	6n	3mn
C	3mn	3mn
b	6n	1

Table IV: Size of Matrices

Using Lagrangian, Linear System having the form

$$L = g^T C g + \lambda^T (Ag - b) \tag{4.26}$$

So, it is optimized as,

$$[C \;\; A^T; A \;\; 0][g \;\; \lambda] = [0; \;\; b] \tag{4.27}$$

In above equation, left hand side matrices $[C \;\; A^T; A \;\; 0]$ are stacked into one matrix

l and right hand side of equation is stacked into matrix r. Then solve linear system using MATLAB's black slash operator,

$g = l \setminus r$

Now, g is vector has X, Y and Z coordinates for mesh surface.For $m * n$ size grid, then first mn rows of vector g has x coordinates. Next mn rows contain y coordinates and last mn rows contain z coordinates. So, retrieve X, Y and Z matrices from vector g. MATLAB's surf or mesh functions can be used to represent 3D surface.

surf(X,Y,Z)

mesh(X,Y,Z)

-2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	-2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	-2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	-2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	-2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	-2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	-2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	-2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	-2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	-2	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	-2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-2	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-2	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-2	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-2

Figure 4.6: C_{vv} for 3*3 matrix

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	-2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	-2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	-2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	-2	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	-2	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	-2	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-2	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-2	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-2	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.7: C_{uu} for 3*3 matrix

1	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	0	1	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	-1	0	1	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	1	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-1	1	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	-1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	-1	0	1	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	-1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1	1	0	-1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	-1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	-1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1	1	0	-1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	-1	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	-1	1	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1	1	0	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	1	-1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	1	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	-1	0	1

Figure 4.8: C_{uv} for 3*3 matrix

Chapter 5

Result and Analysis

The approach discussed in previous chapter has been applied to create reconstructions of different curved objects having circular cross section.This chapter includes what the output of implementation and discussion on that output.

5.1 Apparent Contour

Input image of flower vase and contours used to generate 3D surface for that image are shown in Figure 5.1 for different size of grid.

Figure 5.1: contours of vase for different size of grid

5.2 Flat and 3D surfaces from contours

Using contours retrieved in previous section, reconstructed flat surfaces without inflation constraints and 3D surfaces with inflation for different size of grid are shown. Figure 5.2 represents surfaces for $4*4$ grid. For $8*8$,$28*28$ and $60*60$ grid surfaces are shown in Figures 5.3, 5.4 and 5.5 respectively.

Minimum size of grid that we can use is $4*4$. Surfaces generated using that grid does not match with actual shape of given object.So, surface generated using $4*4$ grid may be considered as worst case. As we increase grid dimensions from four to eight, generated surfaces some what matches with given object. And for $28*28$ grid surface approximates given object. After that even if we increase size, it makes little changes in the output.

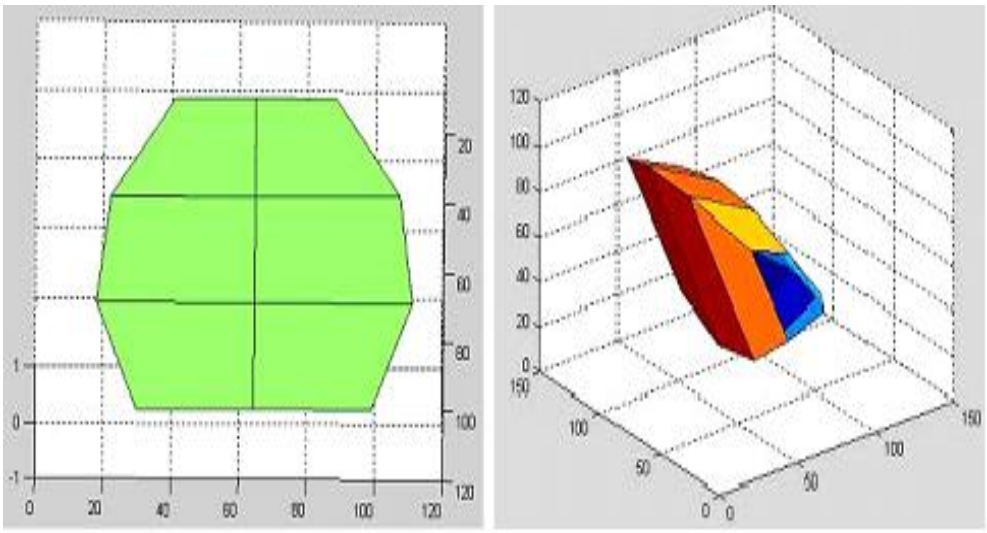

Figure 5.2: Flat and 3D surfaces using $4*4$ grid

While observing **Top view** of 3D surfaces in Figures 5.6 and 5.7 , it is clear that reconstructed surface is not exactly circular. And as grid size increases, it approximates to circular shape. But it is difficult to reconstruct exactly circular shape because we have limitation of contour points also. It means that we can take as much points as number of points in contours.

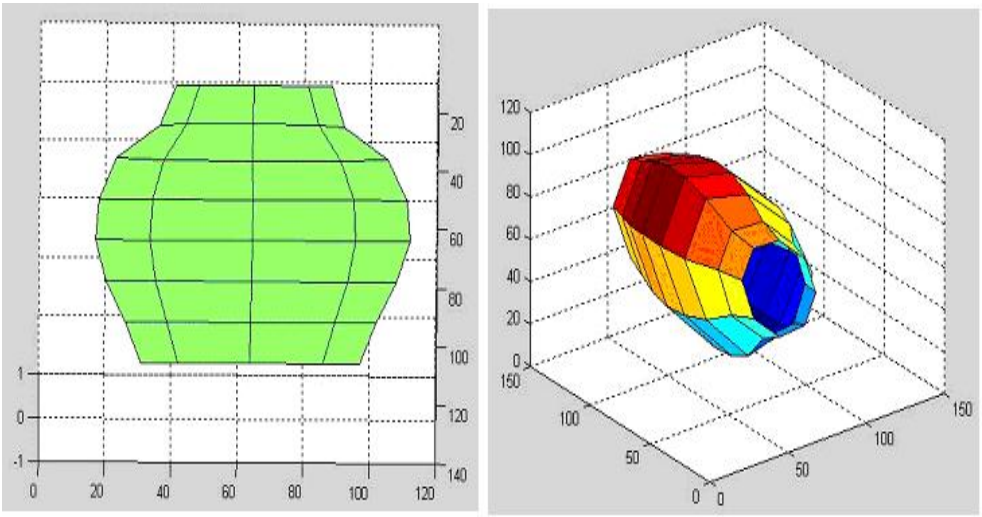

Figure 5.3: Flat and 3D surfaces using $8 * 8$ grid

We can say that reconstructed surface can be represented from all the sides rather than representing just one side of 3D model which is visible in given image like other single view reconstruction approaches.

This approach has been applied on other object (clay pot) also.Input image and boundary extracted from that image are shown in Figure 5.8-(a) and 5.8-(b) respectively. It is observed that there are breaks in boundary points. So, some points are missing also in the contours extracted from that boundary. Reconstructed flat and 3D surfaces are shown in Figures 5.9 and 5.10. Accuracy of these surfaces depends on contour points also.

Results of applying this approach on other clay pot are shown in Figures 5.11,5.12 and 5.13.

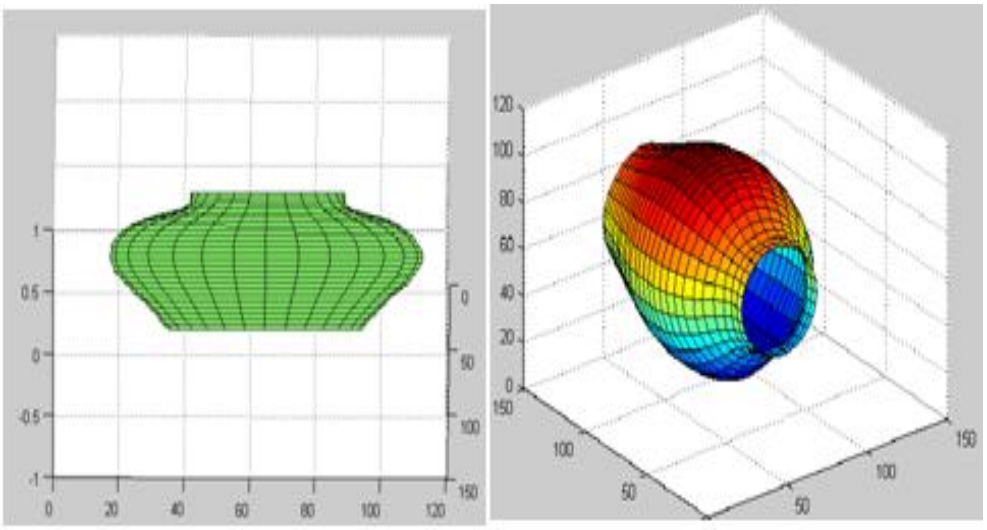

Figure 5.4: Flat and 3D surfaces using $28 * 28$ grid

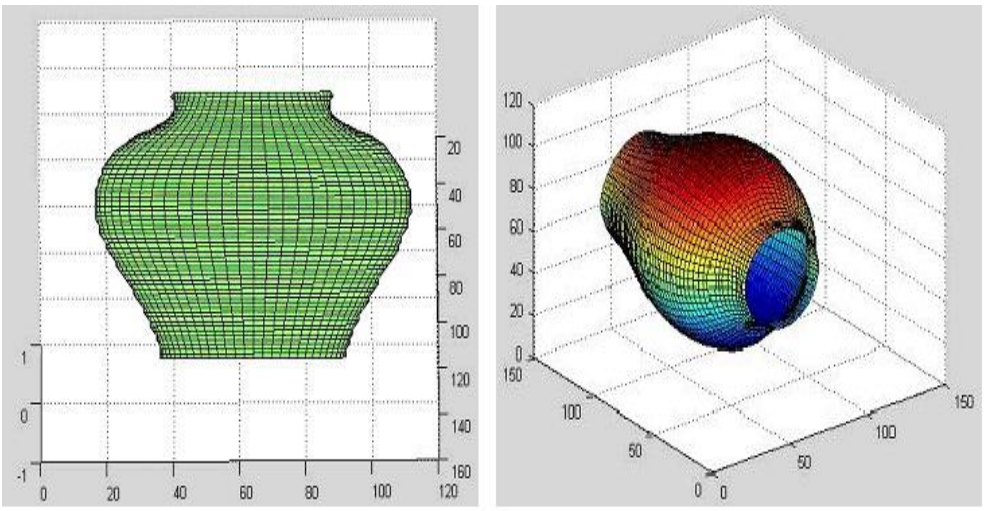

Figure 5.5: Flat and 3D surfaces using $60 * 60$ grid

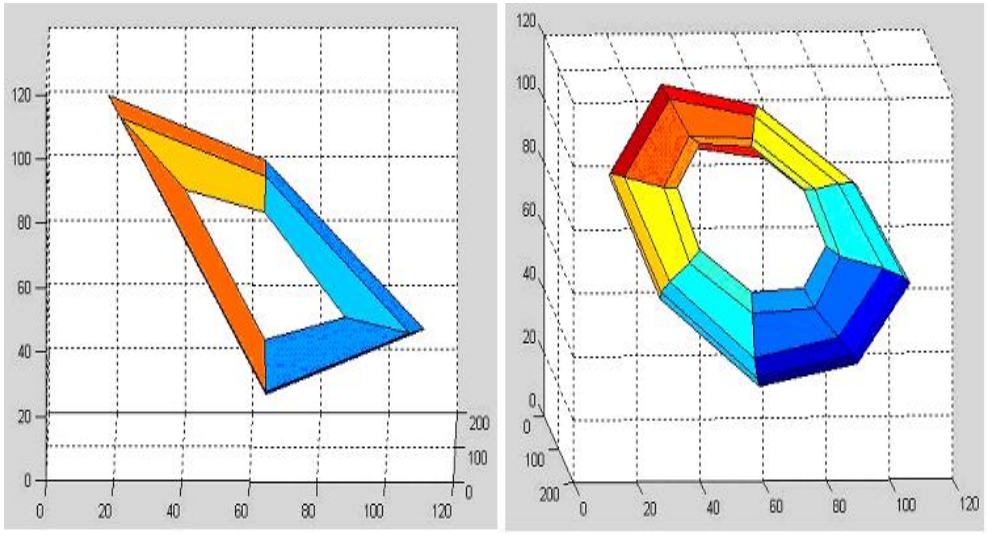

Figure 5.6: 3D surface top view (a) using $4 * 4$ grid and (a) using $8 * 8$ grid

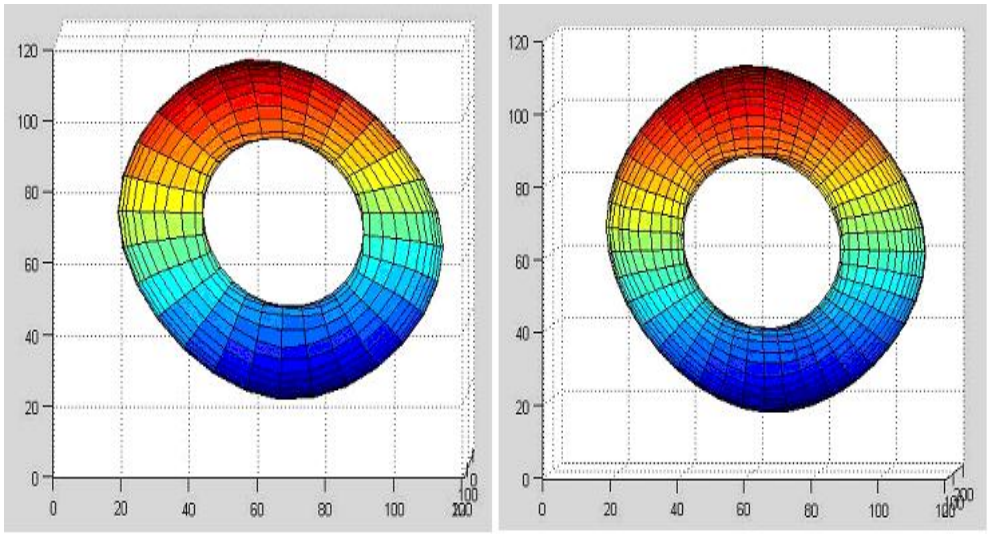

Figure 5.7: 3D surface top view (a) using $28 * 28$ grid and (a) using $30 * 30$ grid

Figure 5.8: (a) Input image (b) Extracted boundary

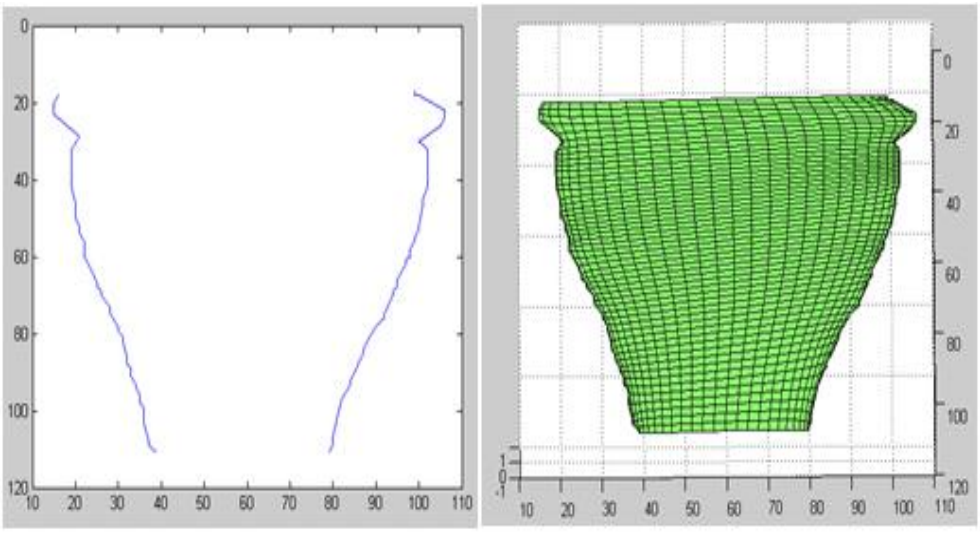

Figure 5.9: (a) Contours (b) Flat surface

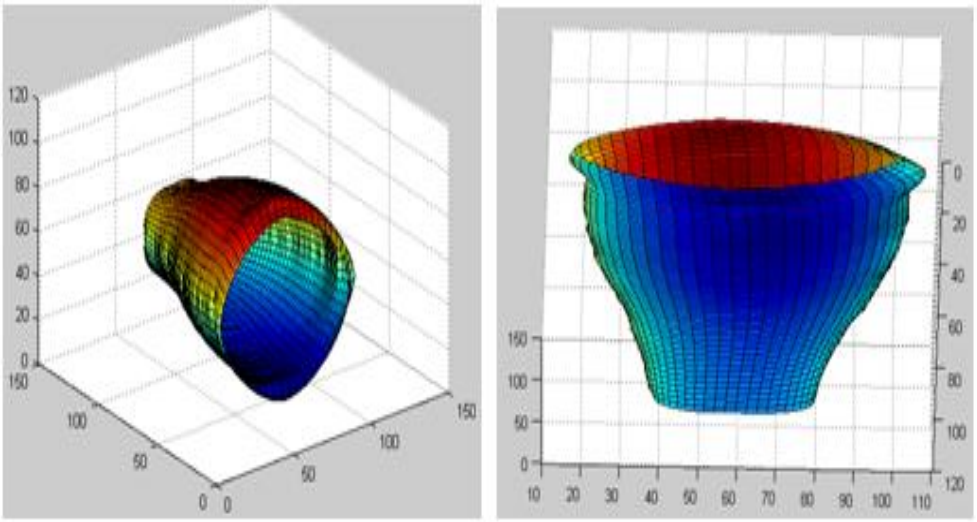

Figure 5.10: 3D surfaces from different views

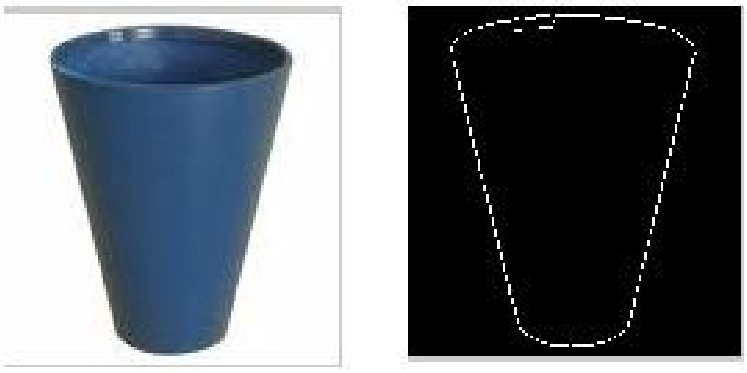

Figure 5.11: (a) Input image (b) Extracted boundary

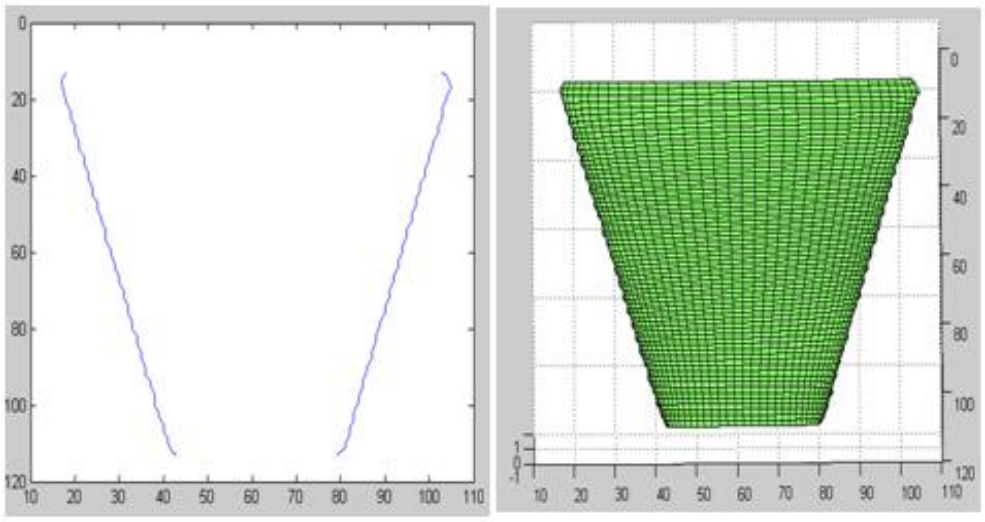

Figure 5.12: (a) Contours (b) Flat surface

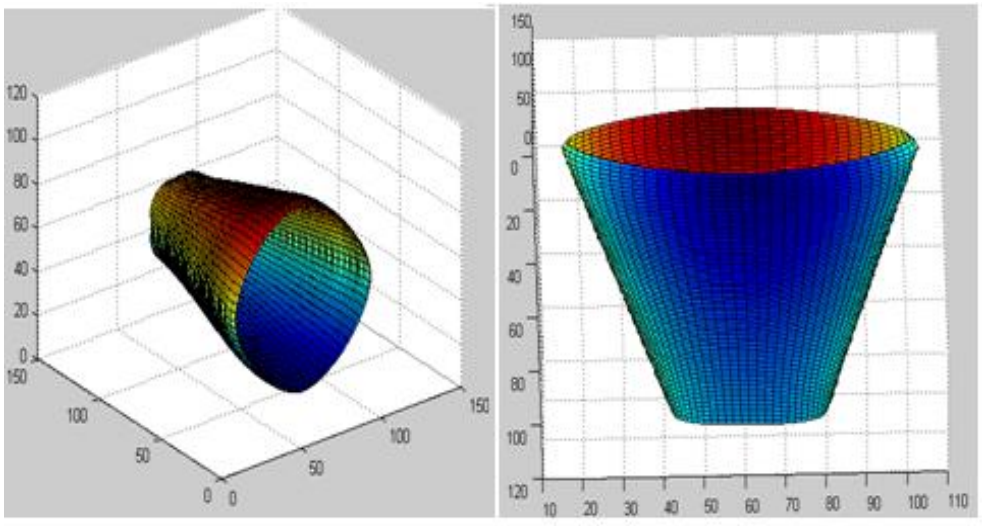

Figure 5.13: 3D surfaces from different views

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

In this study, we proposed approach for reconstructing 3D surface of curved object with circular cross section using object silhouette and constraints.

Problem of 3D reconstruction from single image has minimal information because only one image is available instead of multiple images from different views. Generating 3D from single image is the problem of optimization and generated 3D model is approximation rather than actual model. But single view reconstruction is the only way when multiple images are not available or even multiple images are available, part of image is visible only in one image because of occlusion

From results we can say that contours specifying the shape of the object should be visible in given image. If required contours are not visible in given image, than reconstructed surface does not match with the actual surface of given object. Quality of 3D surface depends on grid size. Surface reconstructed using $4 * 4$ grid is the worst case. Quality of generated surface is improved by increasing grid size. Grid size is again limited by number of points in contours. So, quality of 3D surface depends on accuracy of contour points also.

Observing results it is clear that this approach is limited to the objects which are

symmetric (circular) in cross section and reconstructed surface can be represented from all the sides rather than just representing one side of 3D model which is visible in given image like other single view reconstruction approaches.

6.2 Future Scope

The approach can be extended for complex topologies like torus as well as for complex objects having discontinuous contour generator means crease in contour. The approach used to retrieve contour points can only be applied on the image having clear distinction between object and its background. So, it can be extended to improve accuracy of contour points as well as for all kind of images including reflection also. Texture mapping technique can also be included in 3D reconstruction approach.

References

[1] E. Grossmann, *Maximum Likelihood 3D Reconstruction From One or More Uncalibrated Views Under Geometric Constraints*. PhD thesis, Instituto Superior Tecnio, 2002.

[2] Y. Horry, K. Anjyo, and K. Arai, "Tour into the picture: Using a spidery mesh interface to make animation from a single image," in *proc ACM SIGGRAPH*, pages 225-232, 1997.

[3] I. R. A. Criminisi and A. Zisserman, "Single view metrology," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 123-148, Nov.2000.

[4] A. Criminisi, "Single view metrology: Algorithms and applications," *Pattern Recognition : 24th DAGM Symposium, Zurich, Switzerland*, September 16-18, 2002., vol. 2449/2002, Sept. 2002, pp. 224-239.

[5] J. S. L. Zhang, G. Dugas-Phocion and S. Seitz., "Single view modeling of free-form scenes," in *proc CVPR*, pages 990-997, 2001.

[6] A. W. F. M. Prasad and A. Zisserman, "controllable 3d modelling from silhouette," in *Eurographics Short Papers*, 2005.

[7] "Surface of revolution,http://mathworld.wolfram.com/surfaceofrevolution.html,"

[8] W. C. Kwan Yee Kenneth Wong, "Structure and motion from silhouettes," in *Eurographics Short Papers*, 2001.

[9] A. W. F. M. Prasad and A. Zisserman, "Single view reconstruction of curved surfaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, Pages: 1345-1354,vol. 2, June 2006.

[10] P. Joshi, "Minimizers of curvature-based surface energy," 2006.

[11] Thomas and Finney, *Claculus and Analytic Geometry*. Narosa, sixth ed.

[12] D. F. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics*. TATA McGraw-Hill, second ed., 2002.

[13] "Cyliner geometry,http://mathworld.wolfram.com/cylinder.html,"