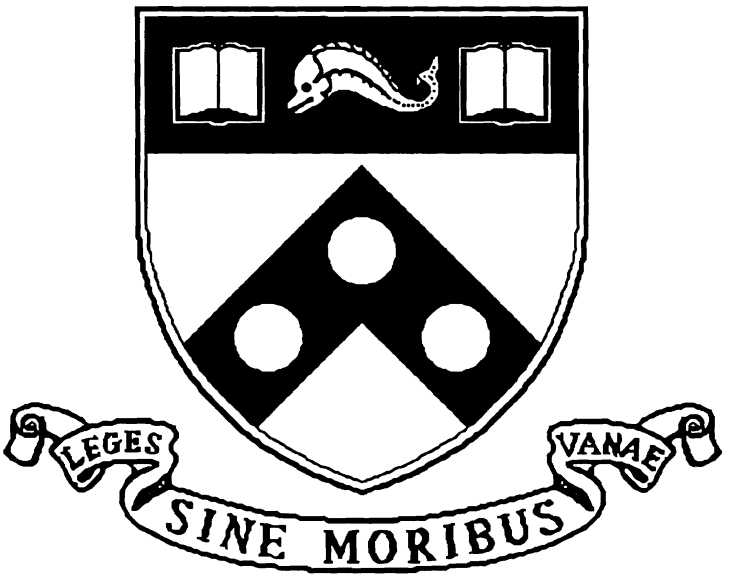
Rapid Prototyping Using Three-Dimensional Computer Vision

MS-CIS-92-70
GRASP LAB 332

Visa Koivunen
Ruzena Bajcsy



University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

August 1992

Rapid Prototyping using Three-Dimensional Computer Vision

Visa Koivunen and Ruzena Bajcsy
General Robotics and Active Sensory Perception
(GRASP) Laboratory
University of Pennsylvania
300C 3401 Walnut Street
Philadelphia, PA 19104-6228

Abstract

A method for building model data for CAD and CAM purposes from physical instances using three-dimensional sensor data is presented. These techniques are suitable for Reverse Engineering of industrial parts, and can be used as a design aid as well. The nature of the reverse engineering task is quantitative, and the emphasis is on accurate recovery of the geometry of the part, whereas the object recognition task is qualitative, and aims to recognize similar shapes. The proposed method employs multiple representations to build a CAD model for the part, and to produce useful information for part analysis and process planning. The model building strategy is selected based on the obtained surface and volumetric data descriptions and their quality. A novel, robust non-linear filtering method is presented to attenuate noise from sensor data. Volumetric description is obtained by recovering a superquadric model for the whole

*Support for Visa Koivunen was provided by the Academy of Finland. The facilities were provided by Navy Grant N00014-92-J-1647, AFOSR Grant 88-0296; Army/DAAL 03-89-C-0031PRI; NSF Grants CISE/CDA 88-22719, IRI 89-06770, and ASC 91 0813; and Du Pont Corporation

data set. A surface characterization process is used to determine the complexity of the underlying surface. A substantial data compression can be obtained by approximating huge amount sensor data by B-spline surfaces. As a result a Boundary Representation model for Alpha_1 solid modeling system is constructed. The model data is represented both in Alpha_1 modeling language and IGES product data exchange format. Experimental results for standard geometric shapes and for sculptured free-form surfaces are presented using both real and synthetic range data.

1 Introduction

Objects in manufacturing industry are usually designed using Computer Aided Design (CAD). In general, CAD-systems are used to design new shapes, and to analyze their structural properties. Computer Vision systems, on the other hand, are typically used to recognize existing objects. These systems aim at representing the shapes in such a way that similar shapes are recognized as the same, i.e. the systems are looking for equivalence relationship with respect to features. The emphasis is on finding characteristic features for the category of shape, and not so much on the accuracy of the representation. On the other hand, *reverse engineering* techniques recover design and manufacturing information from existing physical parts. The information is used, e.g., for representing part geometry, for analyzing structural properties of the part, and for planning the manufacturing process. The reverse engineering requirement is very severe with respect to *accuracy* of the recovery of local and global shape properties and dimensions of the object. This requirement of accuracy has implications both on accuracy for data acquisition, as well as on data interpretation, i.e. fitting models.

We feel that CAD and vision systems can benefit from joining forces. CAD models have been used as a model database for object recognition and visual inspection tasks. On the other hand, solid modeling systems could use geometric models created automatically by a three dimensional computer vision system for different CAX (CAD, Computer Aided Engineering, Computer Aided Manufacturing, Computer Aided Process Planning) purposes. Moreover, both systems should be able to communicate with other automation subsystems using standardized data formats [8].

Manufacturing industry could benefit a lot from applying reverse engineering techniques. They can be useful for producing a new part to replace a broken one. This is especially important if the original manufacturer does not produce the part anymore, or if no design data exists for it. Moreover, if only a very small number of spare parts is needed, redesigning may raise the unit price of the part very high. Similar techniques can also be used as a design aid. The designer could create the a model of the part from clay, and then import the model computed from sensor data to a CAD system. This type of rapid prototyping is especially useful in concurrent engineering paradigm, because the analysis of the part properties and process planning can be started in very early phase of the design process. Automatic data acquisition would reduce the design time especially when designing sculptured free-form surfaces, because the designer should come up with a control point mesh for the surface. It is considered to be a difficult task because the points typically do not lie on the actual surface, and some knowledge about splines is required as well.

In this paper we propose a reverse engineering technique to generate design data and useful information for manufacturing planning starting from an existing component. Three-dimensional computer vision tools are used to produce that information. We have identified two problems in this context:

- automatic complete 3-D data acquisition
- using different techniques from Computer Vision, generate different representations needed for different CAX purposes

The first problem deals with accurate measuring of the complete 3-D geometry of the part. The second one transforms the raw signal data into a model description by reducing the amount of data, and by producing information about the global and local geometric properties of the part. In this report we concentrate on the second subproblem. We present results both on synthetic range images and real parts of varying complexity. We conclude by discussing the feasibility and applications of our approach in geometric modeling and automatic model generation.

2 System Overview

The system we propose consists of the following components:

- Laser range finder.
- Vision system for interpreting 3-D data.
- CAX systems.
- Manufacturing cell.

A laser range finder is used for 3-D measurements because it produces a dense 3-D data set relatively quickly. Dense measurements are especially important in accurate modeling of free-form surfaces. Moreover, this type of data acquisition does not necessarily need any model-based guidance to carry out the measuring, and hence allows the processing to be completely data-driven. It is also possible to measure objects that do not permit contact sensing. A laser range finder sees only one side of the part at the time, and the complete 3-D information have to be merged from a sequence of images. A high level block diagram for the system is depicted in Figure 1. This paper focuses on interpretation of 3-D data and generating useful information for CAX systems.

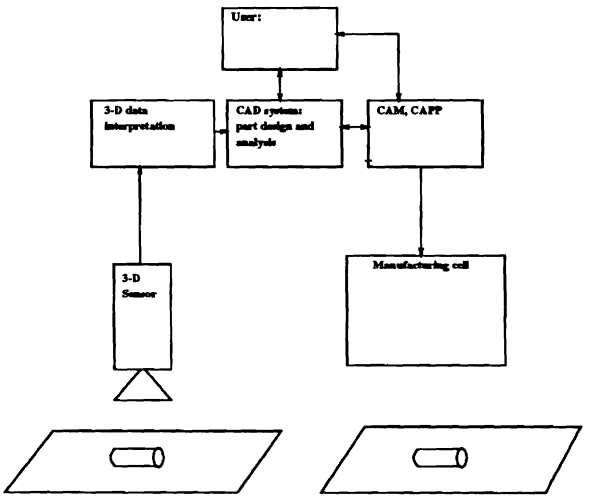


Figure 1: A Block diagram of the proposed reverse engineering system.

The model building process should be able to expose the underlying part geometry from a huge amount of raw sensor data, and simultaneously reduce the amount of data

to be processed. It should recover the dimensions and the shape of the part accurately, reveal possible symmetries and preserve significant surface details like discontinuities. The CAD-model built by the system should be intuitive so that it can be understood and modified by the designer. The system should also provide useful information for analyzing and simulating the part properties, and for process planning. The CAD system we are using is Alpha_1, developed at the University of Utah [1]. Alpha_1 models are defined in modeling language that describes how the part geometry is generated. We chose to produce this type of description, in addition to IGES (Initial Graphics Exchange Specification) file, to be able to associate more semantics for surfaces and space curves, and to be able represent them as design features or primitives like surfaces of revolution, holes and bosses that can be used as a basis for process planning.

The CAX systems can use the obtained model data in several ways. It can be used to visualize the design and to check the manufacturability. The information recovered by vision could be useful for analysis of the mechanical structure or phenomena, for example, for generating finite element mesh for the part. Process planning and NC code generation should be easier if object symmetries can be exposed by vision tools, because the toolpath generation does not have to find out these properties. As a result of process planning, model primitives can be mapped to manufacturing primitives, and NC code can be generated for each processing stage. Other tasks one may want to deal with are determining cost, machining time and material need for the part.

A typical manufacturing cell consists of CNC milling machines and CNC lathes. In the process planning phase for this type of cell, geometric primitives are mapped to different manufacturing stages, and the manufacturing process is a sequence of these stages. Processing parameters, e.g., rotation speed and feed are set, and NC code is generated for each stage. This task can be made automatic in very simple cases. For more complicated parts, designer intervention is often needed. Moreover, the same object geometry can be manufactured in several different ways depending on its functionality. Some sophisticated solid free-form fabrication processes like MD* [34] need only the part geometry to be able to produce the part, and no special tooling and fixturing appears to be required. Furthermore, all the object parts can be manufactured simultaneously, and no separate assembly phase is needed. The shape of the object is

created by disposable masks that are planar cross-sections computed from the CAD-model. MD* sprays each layer with thermal spray using those masks. We are planning to run experiments using both conventional manufacturing process and MD*.

3 Data Acquisition

Laser range finders are widely used for 3-D data acquisition. Typically they produce a dense set of 3-D measurements reasonably fast, and they produce the range data directly without additional processing. The image acquisition does not necessary need any *a priori* object model to guide what to measure, but the process can be completely data driven. Optical range finding techniques need no contact to the surface and hence it is also possible to measure soft materials like clay models. Coordinate Measuring Machines (CMM) can also measure the dimensions of industrial parts accurately. The measuring is usually based on mechanical contact sensing with a probe. For reverse engineering application, there is no model data or toolpath to guide the probe, and a plan of *what* to measure have to be generated, for example by using visual information. Moreover, it is a very time consuming process to use CMM for producing a dense set of measurements to obtain the geometry of the part. We feel that laser range finder suits well for reverse engineering applications, especially if sculptured surfaces are to be modeled.

The current available range sensors provide incomplete 3-D information, obtained from one viewpoint at the time. The complete data set have to be merged from a sequence of images. Hence, there is a need for overall strategy in the scanning process to get the complete 3-D information. There are two main problems to be solved to get a complete 3-D data set:

- *Occlusion*: the laser light does not reach all the surface points.
- *Range shadows*: all the parts illuminated by the laser are not visible for the camera.

The first problem have to be solved no matter what kind of laser range finder is used. The latter problem is typical for sensors using triangulation measuring principle. Maver

and Bajcsy have developed a scanning strategy to get rid of range shadows [23]. It is also important to be able to focus on important details of the part, for example, joints, and scan them with higher resolution to recover accurate description of the details. The calibration of the sensor is also crucial for measuring the part dimensions accurately in terms of the SI unit of length. Our efforts to solve these problems will be presented in a forthcoming report.

4 Need for Multiple Representations

4.1 3-D Object Representations in Solid Modeling

Widely used representation methods in 3-D CAD-systems are Constructive Solid Geometry (CSG) and Boundary representation (B-rep). CSG represents an object as a binary tree where each leaf represents an instance of a simple volumetric primitive and each node represents a regularized Boolean operation of its descendents. A homogeneous transformation matrix describing rotations and translations of the primitive is attached to each leaf. CSG can represent most conventional objects well but it can not represent sculptured free-form shapes precisely. Figure 2 depicts a sample CAD-model using CSG-representation [29].

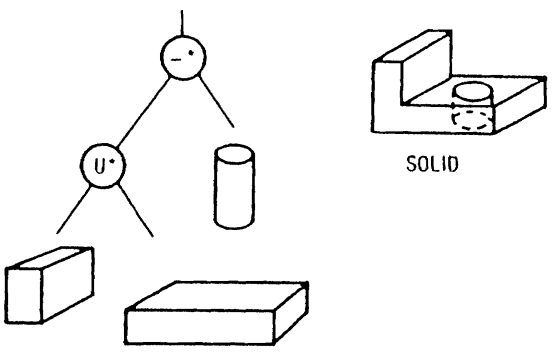


Figure 2: A sample CSG model.

Boundary representation represents a solid by its bounding surfaces. Polyhedral

models are one of the most commonly used and they are best at modeling flat surfaces. Unfortunately, describing free-form surfaces requires a large number of polygons. In order to describe a curved surface splines are often used. Especially B-splines are widely used because of their very good continuity and local control properties. NURB (Non-Uniform Rational B-spline) surfaces can represent precisely both common analytical shapes (conics, quadrics, etc) and free-form surfaces. However, design using B-splines is more challenging because the points on the control mesh typically do not lie on the actual surface. Figure 3 shows the boundary representation for the same object as in Figure 2 [29].

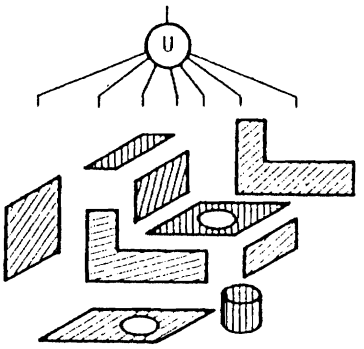


Figure 3: A sample Boundary-representation model

The "Design by Features" paradigm has gained some popularity in design community. It uses manufacturing features to create the part geometry. Typical design features are bosses, ribs and holes. They are abstractions of typical geometric features, e.g., CSG primitives. The features are grouped into a library that can be called from the design system. Boolean operations can be applied on them to create a part.

It seems that there is no single design representation or method that would be the best for every design task. Therefore several CAD-systems employ *multiple representations* [29]. For example, Alpha_1 uses B-spline curves and surfaces as design primitives but it also enables CSG type Boolean operations on solids, as well as design by features [1].

The waterfall design paradigm where design, analysis, prototyping, etc. follow each other sequentially, has been the most employed paradigm in CAD community. It is a very time consuming process because of its sequential nature, and hence it is considered suitable for mass production. Concurrent engineering paradigm where product and process development phases are integrated is proposed to reduce development time and to reduce errors in transition to production. Without product data standards like IGES concurrent engineering is impossible [8]. The capability to communicate with other subsystems during the design process is very important, e.g., because the cost of fixing mistakes later in production phase is much higher than in design phase. Concurrent engineering enables faster customizing and manufacturing at reasonable price in smaller scale. Rapid prototyping is one essential part of the paradigm [24].

4.2 Vision Methods for 3-D Object Representation

CAD based Computer Vision systems have been using CAD models in object recognition problems as a model database [13, 15]. For an introductory article about CAD-based robot vision the reader is referred to [7]. These approaches compute automatically model features for recognition from the CAD model instead of manual training. The design data representation itself is often not unique or rich enough to be used directly for recognition [7]. The computed features can then be used for matching to the sensor data description. Automatic model generation enables rapid changes in product line and inspection criteria because the tedious training phase is avoided. In addition, object recognition strategies, algorithms and model feature indexing may be generated as well [13]. The goal of the object recognition task is to detect if certain type of part occurs in the scene, how many times does it occur, and what is its approximate position and orientation in the space. The nature of the reverse engineering task is more quantitative because we also want to know what is the diameter of the cylindrical part in millimeters or inches, for example.

Different representations used for CAD-based vision can be generally classified into volume, sweep and surface methods. Common volumetric representations are for example voxels and standard analytical shapes like cylinders, spheres and parallelepipeds.

Generalized Cylinders are a typical sweep representation. There is a vast variety of different surface based representations for vision, e.g., planar or second order surface patches, parametric surfaces and superquadrics [3, 2]. Superquadrics can be considered to be a volumetric representation as well, by adding a "less-than" condition to the implicit surface equation. Viewer centered aspect graph descriptions have recently been studied a lot. CAD models have been used for generating aspects [11].

The goal of computer vision based reverse engineering is to build a CAD model from the vision sensor data, and provide useful information for the other CAX processes. The emphasis is upon getting the shape and the dimensions of the part very accurately, preferably within some known tolerance value. The task is not just to convert a huge amount of 3-D measurements into product data exchange format like IGES that a CAD system can read, but also to reveal the structure of the part. The structural information obtained can be used for mapping design primitives to primitive manufacturing processes, and for simulation purposes. The obtained CAD model should also be intuitive, so that the designer can understand it and alter it if necessary. In addition to part geometry, characteristics like functionality of the product to the user, maintenance, repair and cost have to be considered to obtain a good design. Automatic acquisition of CAD models fits very well to the concurrent engineering paradigm, especially if some useful information can be provided for part analysis and process planning.

The situation with vision representations is analog to the situation with CAD representations. There is no single vision representation that would give the best data description for every image. There is indeed a need for *multiple representations.* Instead of using a single method and trying to compensate its shortcomings, we should have a toolbox of representations and just select the appropriate tool for each data set.

5 Building CAD-models from Physical Instances

5.1 Dataflow

Vision tools are used to transfer the raw data set into a model description. We use multiple representations to be able to model efficiently both standard geometric shapes and sculptured surfaces. In this paper we describe a volume based and a surface based representation. A superquadric model is recovered for the entire data set and residuals are computed to evaluate the fit [2]. The obtained superquadric parameters give coarse estimates for position, orientation, size and shape of the parts. If the quality of the fit is good, the superquadric shape parameters can be used to guide the model building process. We use nonuniform B-spline surfaces for representing surfaces. The B-spline surface description is preceded by a surface characterization process to reveal significant surface details and to estimate appropriate size for the control point mesh. A Least Squares fit procedure is employed to extract the control point mesh from a huge amount of raw range data. A tolerance value determined by the user is used to refine the obtained B-spline surface description. Dataflow of the image analysis part is depicted in Figure 4.

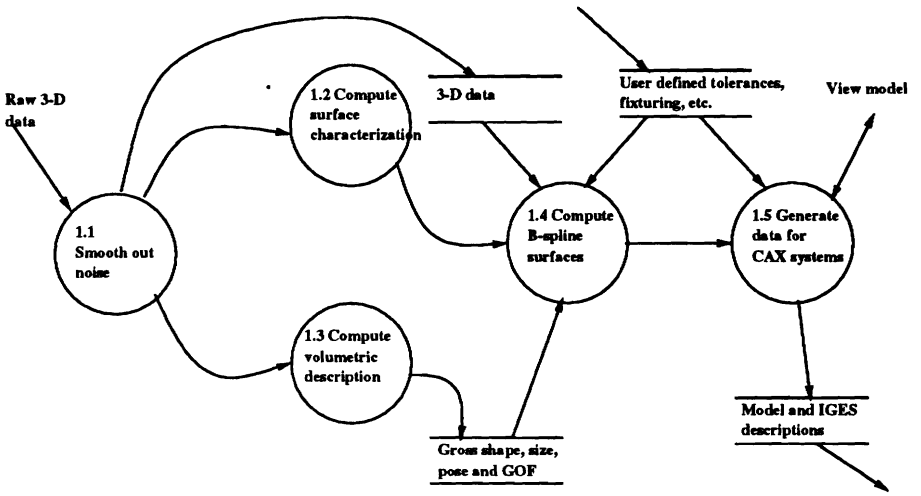


Figure 4: Dataflow of the proposed reverse engineering system.

5.2 Noise Attenuation

Fitting models to data, and hence the estimated parameters can be very sensitive to noise. Range sensors are subject to various noise effects, and the noise is not normally distributed [4]. To get accurate estimates the noise must be smoothed out by filtering, or by using robust estimation methods [18]. The selection of the filtering method depends not only on the noise distribution but also on the future processing of the signal. If it is followed by differentiation, the image is usually filtered with an approximately Gaussian filter or a cubic spline to make the differentiation a well-posed problem [32]. If the processing groups pixels by their similarity, it is desirable that the filtering method would preserve discontinuities between homogeneous regions.

The proposed noise attenuation method is based on robust Least Trimmed Squares (LTS) estimation method [31]. The goal is to smooth out statistical outliers and preserve the geometry of the original noise free signal. The outliers are caused by heavily tailed distributions, or are just bad samples, or members of other data population. The concept of robustness means ability to produce reliable results although the idealized assumptions for which the estimator is optimized are not completely valid. The breakdown point of the estimator describes formally the smallest percentage of outlying points which causes incorrect estimates. Least squares estimation has a breakdown point of 0 %. LTS filtering of the data points minimizes the sum

$$\sum_{m=1}^{h} (r^2)_{m \in n}, \tag{1}$$

where $(r^2)_1 \leq ... \leq (r^2)_h \leq ... \leq (r^2)_n$ are the ordered squared residuals and h is the number of residuals used in summation. The LTS method achieves the maximal breakdown point for $h = [n/2]+[(p+1)/2]$, where n is the number of data points in the neighborhood and p is the number of parameters to be estimated. Moreover, the LTS filter is an efficient estimator under Gaussian distributed inlier noise. If the number of parameters to be estimated is 1, the maximal breakdown point in 5-by-5 neighborhood is obtained with $h = 13$. The median of the neighborhood is used as a reference value for computing residuals. If the filtering is iterated, the output from previous iteration is used as a reference signal for the next iteration. Furthermore, iterative filtering is done using floating point computation to avoid quantization errors introduced after

each iteration. The iteration is continued until good convergence, or the maximum number of iterations is reached. Lee has proposed the Modified Median Filter (MMF) [20] to overcome some edge jittering problems of median filtering. MMF can be used as a reference signal instead of the median.

5.3 Surface Characterization

A surface characterization process is employed to find geometrically homogeneous surface patches, and to determine the complexity of the surface. Differential surface properties are estimated with constant coefficient local window operators. The underlying surface f is assumed to take the parametric form of a polynomial in each neighborhood:

$$f(i,j) = a_1 + a_2 i + a_3 j + a_4 i^2 + a_5 ij + a_6 j^2, \tag{2}$$

where the parameters $a_k (k = 1, 2, ..., 6)$ are to be estimated and i and j are the row and column coordinates in the neighborhood. The second order polynomial above is needed to be able to estimate the second partial derivatives in one pass. First and second partial derivative estimates can be obtained directly from the coefficients of the polynomial [4].

An alternate way to estimate surface parameters employs robust estimators. The LTS scheme above can be used to estimate the coefficients a_k from (2). The method gives reliable results even when there is more than one statistical population present in the neighborhood, and under various noise distributions [18]. In the fitting procedure appropriate order model should be used to get accurate results. Besl proposed a variable order model method where the parameters from the fit order yielding the best fit quality measure are chosen to be the final fit result [5]. In our study we used zero, first and second order models.

Surface coefficients are used to find the gradient, the gradient magnitude and the second directional derivatives, and their directions, and the first partial derivatives in the direction of either extremum of the second directional derivative. A surface type label is given to each pixel according to the spatial properties in the neighborhood of each pixel [14]. Geometrically homogeneous regions are formed by grouping connected

pixels of the same surface type. For detailed experiments on the classification scheme reader is referred to [18].

The complexity of the underlying surface can be inferred from the labeling, and hence the size of the control mesh for the B-spline surface estimated. The mesh size is determined by the maximum number of extracted surface patches in each parameter direction, and by knowing how many control points is needed to describe certain order surface by using B-splines. The maximum order is selected because all entries in same parameter direction must have the same amount of control points. If the control mesh has too few points, it is not able to describe all the degrees of freedom of the surface, and the fit will not converge. If it has too many points, it may hide the structure of the surface and make it difficult to understand and modify the design. It is also important to detect the surface discontinuities to be able to insert multiple control points where the discontinuity occurs. They can be detected using the gradient and the second directional derivatives obtained in surface characterization process, or by using a separate edge detector [10]. Edge data and surface label data are then overlapped to get the complete description.

5.4 B-spline Surface Fit

A non-uniform rational B-spline surface (NURBS) is a more general case of nonrational B-spline surface, and is defined as a function of parameters u and v as follows:

$$S(u,v) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} h_{i,j}B_{i,j}N_{i,k}(u)M_{j,l}(v)}{\sum_{i=1}^{n}\sum_{j=1}^{m} h_{i,j}N_{i,k}(u)M_{j,l}(v)}, \tag{3}$$

where $N_{i,k}$ and $M_{j,l}$ are the basis functions, $h_{i,j}$ are the weights, and the $B_{i,j}$'s are the control points. n and m stand for the number of control point vertices in each direction. The basis functions $N_{i,k}$ of order k are defined recursively as follows:

$$N_{i,1}(u) = \begin{cases} 1 & \text{,if } x_i \le u \le x_{i+1} \\ 0 & \text{,otherwise} \end{cases} \tag{4}$$

$$N_{i,k}(u) = \frac{(u-x_i)N_{i,k-1}(u)}{x_{i+k-1}-x_i} + \frac{(x_{i+k}-u)N_{i+1,k-1}(u)}{x_{i+k}-x_{i+1}}, \tag{5}$$

where x_i's are ordered set of knots from knot vector. A convention $0/0 = 0$ is used for the basis function computation. Basis functions $M_{j,l}$ of order l for parameter v are

computed similarly. We chose to employ 4th order ($k = l = 4$, cubic) B-splines since they allow a point of inflection. Figure 5 depicts a sample B-spline surface and its control point mesh.

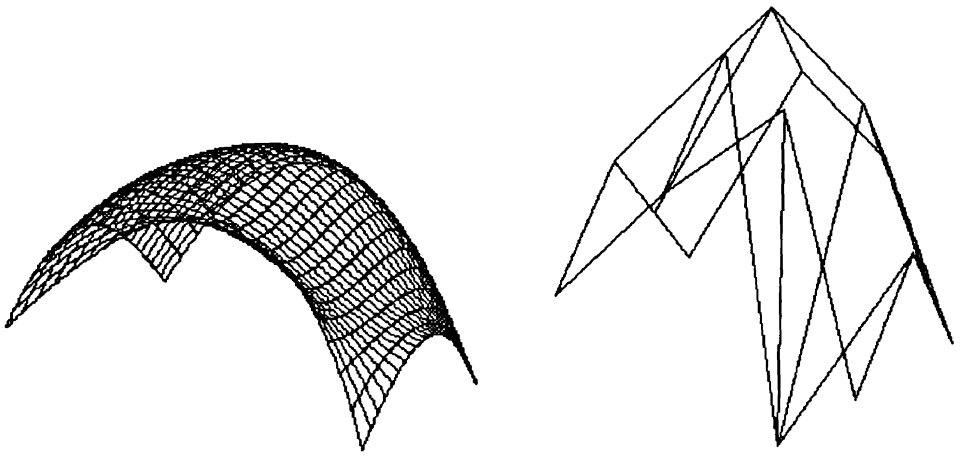


Figure 5: A B-spline surface and its control point mesh

The widely used chord length parameterization for a B-spline in one direction is computed as follows:

$$u_1 = 0, \quad u_i = u_{i-1} + \frac{|P_i - P_{i-1}|}{\sum_{j=1}^{n} |P_j - P_{j-1}|}, u_n = 1.0 \tag{6}$$

where P_i's are data points. Lee has proposed a centripetal method for parameterization and claims that in most cases it results in better shapes than uniform or chord length method [19]. The parameter values are computed by:

$$u_1 = 0, \quad u_i = u_{i-1} + \frac{|P_i - P_{i-1}|^{\frac{1}{2}}}{\sum_{j=1}^{n} |P_j - P_{j-1}|^{\frac{1}{2}}}, u_n = 1.0 \tag{7}$$

Often the parameter values are normalized to $[0, 1]$ range. The number of knots t is related to the number of control points by $t = n + k$ where n is the number of control points and k is the order. An open end condition is used for the spline i.e. the first and last values in the knot vector occur with multiplicity equal to the order of the B-spline. In practice it means that the curve always begins at the first control point and ends at the last control point. A B-spline curve with open end condition and the corresponding control polygon is depicted in Figure 6. This property is utilized

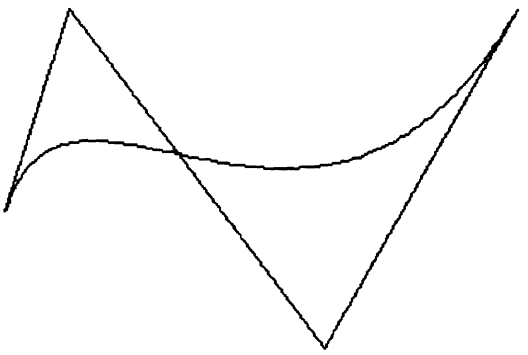


Figure 6: A B-spline curve with open end condition and the corresponding control polygon

also in B-spline subdivision. Other possible end conditions are periodic and floating end condition. The periodic end condition is useful for producing closed curves and surfaces. The knot vector can be adjusted to the distribution of the parameter values. We chose to do it later in the refinement process.

As a result of the data acquisition we have a huge amount of raw range data. We use least squares fitting to be able to generate a control point mesh for a B-spline surface that approximates accurately the original one. Now we have to solve $B_{i,j}$'s from equation (3), and $S(u,v)$'s are the measured data points. All the weights are originally set to 1.0, and the surface degenerates to a nonrational B-spline surface. The weight values $h_{i,j}$ can be later adjusted in surface refinement [25]. Using matrix representation the solution is:

$$[B] = [[C]^T[C]]^{-1}[C]^T[S], \tag{8}$$

where elements of C are $C_{i,j} = N_{i,k}M_{j,l}$, S is the matrix of data points, and B is the obtained control points mesh [30]. We employed also Singular Value Decomposition to solve the control point mesh. We prefer it for very large number of samples and control points because it is very stable algorithm, and almost never fails [27].

Surface parameterization depends very much on scanning procedure. For cylindrical scan procedure it does not cause any problem. If the samples are scattered arbitrarily, one may have to do resampling to get a rectangular grid of samples for the fit procedure.

However, the sampling does not need to be uniform. It is not possible to describe certain surfaces of arbitrary topology, e.g., surfaces with handle, with single nondegenerate B-spline. Proper surface segmentation helps avoiding the problem. Loop and DeRose describe generalizations of B-spline surfaces that are capable of capturing such surfaces [21]. However, the work was done using uniform B-splines, although the authors claim that it can be expanded to non-uniform and rational cases as well.

A simple and straight forward solution to run a fit for any surface topology is to employ trimmed surfaces. A trimmed B-spline surface is essentially a regular B-spline surface where certain parts of the surface are marked "invalid" [12]. The boundaries of the object can be used to compute trimming curves that divide the surface into valid part, which is the object surface, and the invalid part, which is the background. The fit procedure can be run using a rectangular grid of points that exceeds the object boundaries, and then the part of the surface that is not object surface is then declared invalid. Most solid modelers have trimming operations included because they have to be able to deal with surface intersections anyway. Trimmed surfaces are included in IGES standard and in Alpha_1 system as well [16, 1]. An example of a trimmed surface is depicted in Figure 7 [12].

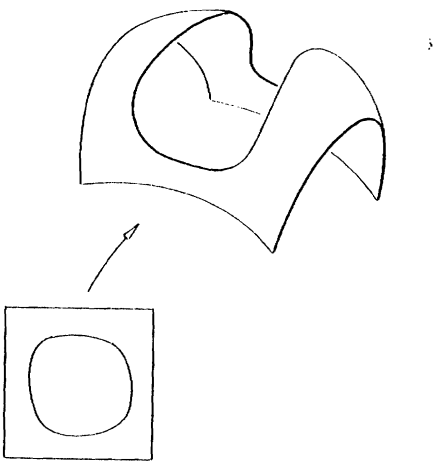


Figure 7: A sample trimmed surface

The B-spline refinement process is guided by a user given tolerance value. The error of the approximation is defined as Euclidean distance between the actual mea-

sured surface and approximated surface with same (u, v) parameter values. We use B-spline subdivision to deal with discontinuities. Curve or surface discontinuities can be represented, if a knot with multiplicity equal to the order of the B-spline is inserted. Control point refinement is done by moving old control points and inserting knots to get a more accurate approximation of the underlying surface [25]. Adding a certain number of knots has a consequence of adding the same number of control points as well. The Oslo algorithm is useful if one wants to insert knots to add degrees of freedom, but not to change the actual curve or surface [9].

5.5 Volumetric Representation

Superquadrics are a family of parametric surfaces that can represent shapes ranging from cylinders and parallelepipeds to ellipsoids. A superquadric model was used to recover a volumetric description for the entire range data set. The model is recovered in object centered coordinate system. An implicit equation for superquadric surface is defined as follows:

$$f(x,y,z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1, \tag{9}$$

where a_1, a_2, and a_3 define the superquadric size in x-, y- and z-axis direction. ε_1 and ε_2 are the superquadric shape (squareness) parameters in the latitude and in the longitude plane, respectively. Figure 8 depicts some typical superquadric shapes.

The shape family can be augmented by applying a set of global deformations like tapering and bending to superquadrics. A volumetric representation is recovered instead of surface representation by replacing "=" condition by "$\leq$" in the implicit surface equation. The superquadric model was modified by Solina with additional ε_1 to improve the recovery of cylindrical objects [28]. The inside-outside function defined as $F(x,y,z) = f(x,y,z)^{\epsilon_1}$ determines where a point lies relative to superquadric surface. To find the smallest superquadric that fits to a set of range data the following expression is minimized [28]:

$$\min \sum_{i=1}^{N} [\sqrt{a_1 a_2 a_3}(F(x_i, y_i, z_i) - 1)]^2 \tag{10}$$

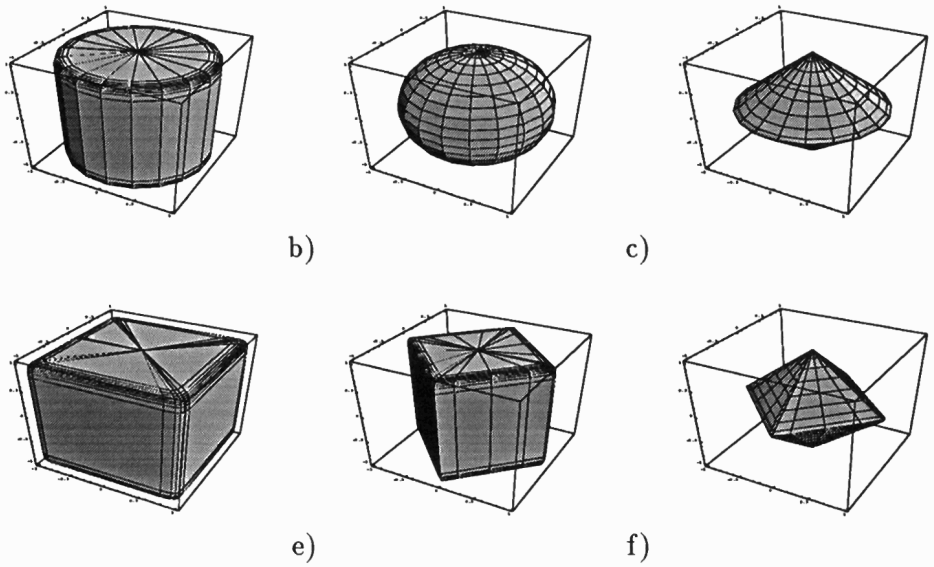


Figure 8: Sample superquadric primitives with different shape parameters: a) $\varepsilon_1 = 0.1$, $\varepsilon_2 = 1.0$, b) $\varepsilon_1 = 1.0$, $\varepsilon_2 = 1.0$, c) $\varepsilon_1 = 2.0$, $\varepsilon_2 = 1.0$, d) $\varepsilon_1 = 0.1$, $\varepsilon_2 = 0.1$, e) $\varepsilon_1 = 0.1$, $\varepsilon_2 = 2.0$ and f) $\varepsilon_1 = 2.0$, $\varepsilon_2 = 2.0$, respectively.

The Levenberg-Marquartd method is used for non-linear least squares minimization [27]. As a result coarse shape, size, position and orientation parameters are recovered. To be able to find the global minima in the fit procedure, and to recover the parameters accurately, it is crucial to have good first estimates for the fit procedure.

The quality of the fit can be used to evaluate how well the data fits to the model. If it is good the shape parameters can expose object symmetries, and they can be used to guide the model building process. Hence, the superquadric primitives have to be divided into categories based on the recovered shape parameters. For example, primitives with $\varepsilon_2 = 1.0 \pm \delta$, where δ is a given threshold value, are considered rotationally symmetric objects, and primitives with $\varepsilon_1 = 0.1 \pm \delta, \varepsilon_2 = 0.1 \pm \delta$, or $\varepsilon_1 = 0.1 \pm \delta, \varepsilon_2 = 2.0 \pm \delta$ are both parallelepipeds. Superquadric representation is not unique, for example in the case of parallelepipeds (Figure 8 d) and e) above), but by categorization the problem can be avoided.

5.6 Building the Model

The model building tools are selected depending on the parameters acquired from the fit procedure, and the goodness of the fit. If rotational object symmetry has been detected, a surface of revolution is employed as a model primitive. If any other type of symmetry is discovered, a sweep surface is a natural choice. This kind of structural information is very helpful for process planning because these properties do not have to be discovered in the planning phase. The symmetry and the axis of symmetry must be verified because of the coarseness of superquadric parameters. If no symmetry is present, or the quality of the fit is poor, the surface is modeled as a set of surface patches sewed together. The surface patches may also be trimmed surfaces to be able to deal with surfaces of arbitrary topology. Alpha_1 includes specific operations for declaring adjacencies, i.e. sewing surfaces [1].

6 Experimental Results

The set of experiments was chosen to show the ability to construct a model for both standard geometric shapes and sculptured free form surfaces. We made experiments using both real and synthetic range data. A synthetic Cylindrical Pin test data was generated by constructing a Alpha_1 CAD model of an object, and the range image was produced from it by using Z-buffer algorithm. The dimensions of the part are relative because the data is not actual measured but synthetic data. Test data for the Fan Blade was scanned using GRASP-laboratory laser range finder which is based on triangulation principle [33]. The resolution in x and y direction for the sensor is $1.0\,mm$ ($\triangle x = \triangle y = 1.0\,mm$) and the depth resolution $1.5\,mm$. The Face Mask image from NRCC range image library was chosen to demonstrate capability to model very complex sculptured surfaces. The x- and y-resolution for the sensor is $1.0\,mm$, and the least significant value for z-coordinate is $10\,\mu m$. The test images are depicted in Figure 9.

Least trimmed squares filtering was employed for noise attenuation. Gaussian distributed noise with zero mean and $\sigma = 10.0$, and random bit error noise with probability $p = 0.002$ was added to the synthetic "Wedding Cake" test image to find out the per-

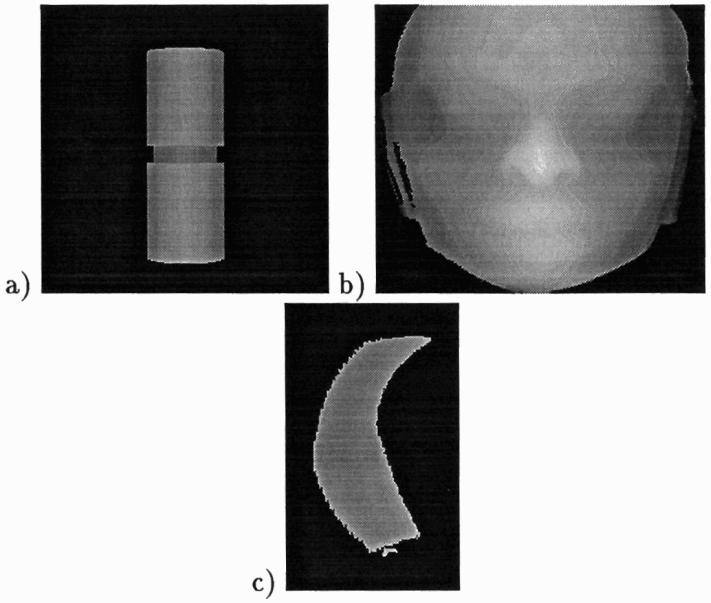
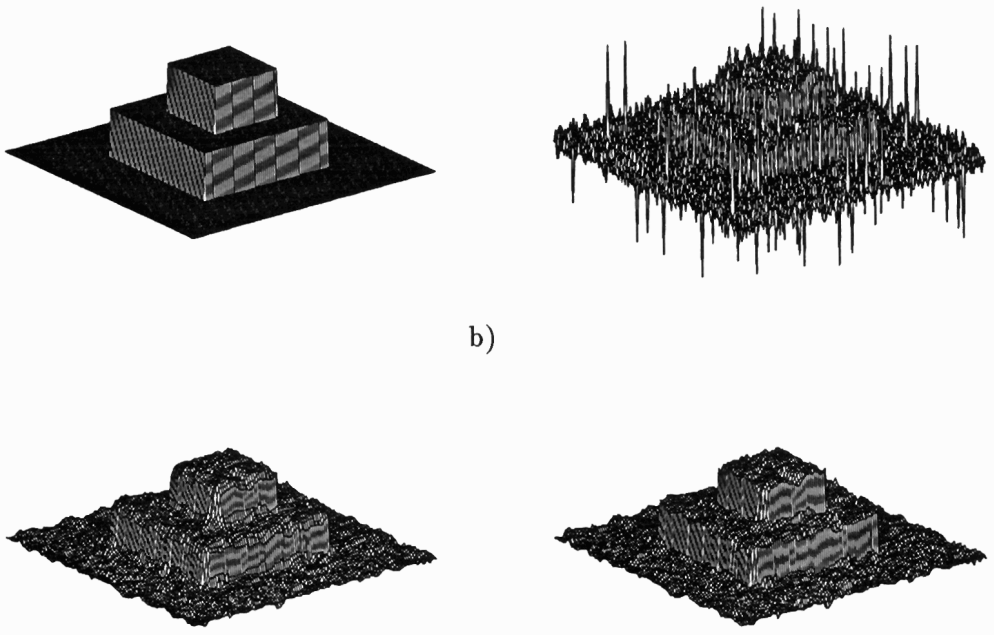


Figure 9: Test images: a) The Cylindrical Pin is a synthetic range image produced from a CAD model, b) the Face Mask is a real range image from NRCC range image library, c) and the Fan Blade is a real range image produced using GRASP-laboratory laser range finder

formance of different filtering schemes under various noise distributions and relatively high noise. The height difference between different levels of the cake is 50. For the Cylindrical Pin Gaussian the noise parameters were $\mu = 0$ and $\sigma = 4.0$, and the random bit error probability $p = 0.001$. The evaluation of the filter performance depends also on the final goal of the processing. For example, many edge detection algorithms smooth out noise with Gaussian type filtering or cubic splines to make the surface differentiable, and to get rid of multiple responses [32]. As a result they also smear sharp edges. We discuss here only how well the method attenuates different types of noise, and how well it preserves the original geometry of the image. LTS and median filtering were considered here. The original noise-free image, the noisy image, and the images after filtering are depicted in Figure 10. The RMS errors between the original noise-free image and the filtered images are shown in Table 1. Both filters smooth out the impulse noise and preserve discontinuities, although LTS filter performs slightly better. Furthermore, LTS filtering is also efficient under Gaussian distributed inlier noise. The corners appear to suffer from this type of non-linear filters, and hence rather small



a) b)

c) d)

Figure 10: Noise attenuation results for the "Wedding Cake": a) original data, b) noisy data, c) median filtered data, and d) LTS filtered data.

neighborhoods should be employed. The LTS filtering result for the Cylindrical Pin is show in Figure 11.

Surface characterization was used to detect homogeneous surface patches and to find out the complexity of the underlying surface. The results can be used to determine how many control points are needed in the control mesh to be able to describe all the degrees of freedom of the underlying surface. For each patch, a control point mesh size equal to the order of the B-spline is needed. We used second order surface patches, and hence we need three control points in each parameter direction for defining the patch. The neighborhood size for the surface coefficient estimation, and a zero threshold value for the surface classification have to be determined. Constant coefficient window operators were used for estimating surface parameters. The neighborhood size was 7-by-7 and the zero threshold value was set to 1.5. Surface characterization results for test images are depicted in Figure 12. For noisy real range data the surface characterization tends

Table 1: RMS errors for noise attenuation.

Noise dist/ Filtering method	Gaussian $\mu = 0, \sigma = 10.0$	Bit reversal $p = 0.002$	Bit rev. & Gaussian
median	3.83	1.89	3.85
LTS	3.65	1.82	3.68

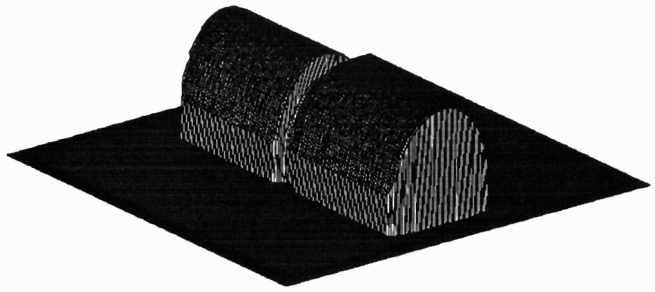


Figure 11: Noise attenuation results for the Cylindrical Pin using LTS filtering.

to fragment the surfaces into slightly more patches than there actually exists because of noise and local support for the computation. However, this property is less severe than too few patches because the control point mesh with too many points is able to describe the shape of the surface which is not possible with too few control points. The surface stays fair as long as the number of control points is much lower than the number of sample points. Furthermore, B-spline curves have a variation diminishing property which means that the curve does not oscillate cross any straigh line more than its control polygon. For surfaces this property is not known. A method for decreasing the number of control points by knot removal has been proposed by Lyche and Mørken [22]. They use a tolerance value to examine whether a knot can be removed from knot vector.

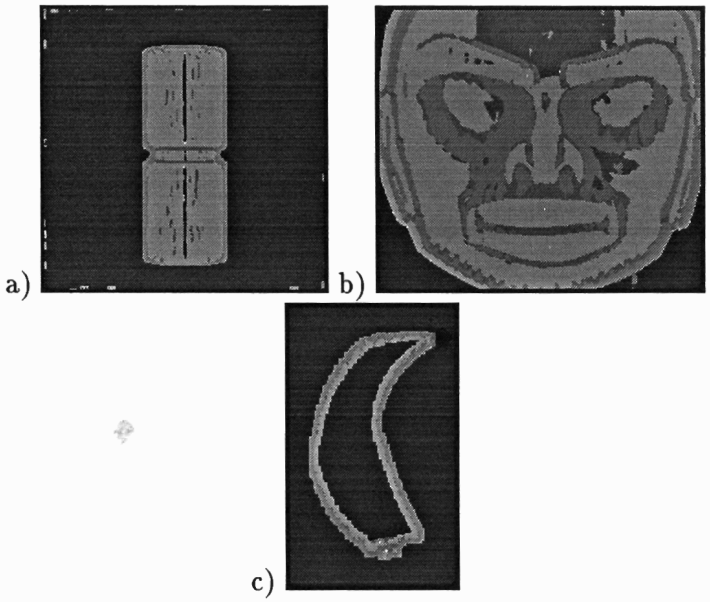


Figure 12: Surface classification results for the Test images: a) The Cylindrical Pin, b) the Face Mask and c) the Fan Blade. Different surface types are shown in different gray scale values.

Robust estimation methods produce more accurate surface characterization results but the computational complexity is significantly higher. If the estimation is done by using robust window operators, no filtering prior to the estimation is required. A detailed description on experiments on LTS estimation based surface classification can be found in [18]. The current implementation uses variable order fit method proposed by Besl [5]. It provides more accurate results than only second order fit because it is able to select the appropriate model for the fit.

A superquadric model was recovered for LTS filtered test images to find out gross object shape, size, position and orientation parameters. The model recovery is done to the whole data set, and the goodness of fit measure [28] can be used to determine if the obtained parameters are useful for the model building. A threshold value for the goodness of fit have to be set. The superquadric primitives are divided into shape categories, e.g., circularly symmetric shapes and parallelepipeds, to be able select an appropriate model building strategy. If the quality of the fit is good, the main axis of the superquadric primitive can be used as the axis of symmetry, and the shape parameters can be used to select appropriate CAD model primitive. However, the

symmetry and the axis of symmetry must be verified because of the coarse nature of the superquadric parameters. For verifying a rotational symmetry we used Direct Least Squares fit along the assumed symmetry axis [26]. The fit gives the radius and the center point of the circle and a quality of the fit measure. The line along the center points is used as the axis of symmetry. The superquadric models are shown in Figure 13. The corresponding shape and size parameters and goodness-of-fit measures

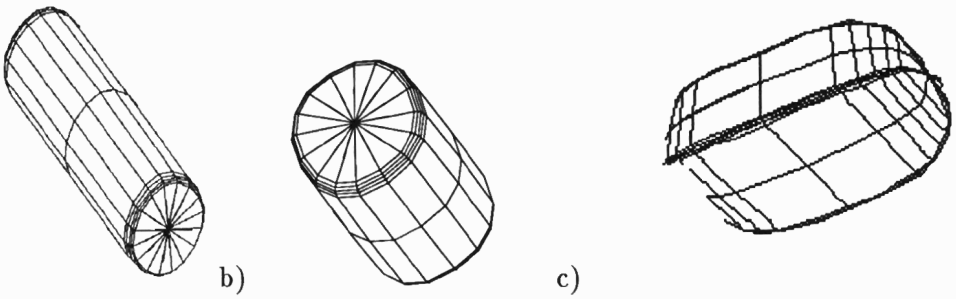


Figure 13: Recovered superquadric primitives for Test objects: a) Cylindrical Pin, b) Face Mask, and c) Fan Blade.

are displayed in Table 2. In general, the parameter recovery works well for convex

Table 2: Superquadric model recovery results

Parameters/ Test image	Shape	Goodness of fit χ^2
Cylindrical Pin	$\varepsilon_1 = 0.10, \varepsilon_2 = 1.08$	0.06
Fan Blade	$\varepsilon_1 = 0.59, \varepsilon_2 = 0.12$	0.20
Face Mask	$\varepsilon_1 = 0.15, \varepsilon_2 = 0.87$	0.15

symmetric objects. The number of superquadric parameters can be extended to deal with global deformations like cavity, bending and tapering. However, in the case of Fan Blade the quality of the fit did not get better despite the deformations.

Uniform, chord length and centripetal parameterization methods were compared for B-spline curve and surface fits. LTS filtering gets rid of statistical outliers that would cause high errors in the Least Squares fit procedure. Similar knot vector and the same number of control points were used to find out how well each method can

capture the shape of the underlying surface and how large errors they produce. The obtained B-spline curves for a sample profile from Face Mask image using different parameterization methods, and no refinement, are plotted in Figure 14. The corresponding distances between the measured and the recovered data are shown as well. Maximum and RMS error for different parameterization methods are shown in Table 3 for a profile from Face Mask image. The results indicate that both the centripetal and

Table 3: Error measures using different parameterization methods for profiles from the Face Mask and from the Cylindrical Pin. No refinement was done.

Parameterization method	RMS error (Face Mask)	Max. error (Face Mask)	RMS error (Cylindrical Pin)	Max. error (Cylindrical Pin)
Uniform	0.39	1.58	0.56	2.67
Centripetal	0.40	1.52	0.49	2.29
Chord length	0.39	1.43	0.51	2.41

the chord length parameterization provide good accuracy for fit purposes. The chord length method gives more predictable results because the maximum and RMS errors appear to decrease almost linearly as the number of control points increases. The accuracy of the uniform parameterization is not sufficient for rapidly changing surfaces or scattered data. In general, the accuracy is good if the surface is smooth. Larger errors are caused by rapid changes in the surface shape. Surface discontinuities are the worst case because of the continuity property of the B-splines. Hence, the refinement by subdivision is vital for getting accurate results. An example of errors introduced by discontinuities using different parameterization methods for a sample profile from the Cylindrical Pin are depicted in Figure 15. The fit results could be made better by reflecting the distribution of parameter values in the knot vector. We chose to do this later in the spline refinement process.

The B-spline fit produces surfaces that are guaranteed to have C^{k-2} continuity, where k is the order of the B-spline. Therefore they smooth out curve and surface discontinuities. To be able to represent discontinuities, the B-spline curve or surface

have to be refined by subdividing it where the discontinuity occurs. Multiple knots are inserted to the point where the discontinuity was detected to force the spline to pass through that point. The multiplicity is equal to the order of the B-spline. Figure 16 depicts the effects of B-spline subdivision for the same profile used in Figure 15.

The B-spline refinement process takes place after subdivision. It is driven by a user given tolerance value. A knot, and hence a control point is added to a point where local error maxima exceeding the tolerance value occurs. If the RMS error exceeds the tolerance value the number of control points is increased by one for each patch extracted in the characterization phase in both parameter directions. Refinement results for the profile from the Face Mask image are shown in Figure 17 in which knots were inserted where local error maxima greater than the given tolerance value occurred. A tolerance value 1.0 mm was used for the profile.

Preliminary tests using trimmed surfaces indicate that much larger number of control points have to be used. The knot vector should be quite dense especially near the object boundaries. The local control property of B-splines isolates the errors introduced by the surface boundary. We used also lower order splines because they cause less oscillation by the boundary.

Different B-spline fit strategies were applied for different type of objects. For the Cylindrical Pin the superquadric shape parameters revealed the cylindrical shape, and the final B-spline surface was created by finding the axis of symmetry of the object, and then creating a surface of revolution by using a cross section of the object. The quality of the superquadric description for the Fan Blade was not good, and hence the part was modeled as a set of adjacent surfaces. Because the sample grid was not perfectly rectangular a resampling process was run. For each scanline on the object surface a B-spline curve was fitted, and the resampling was done by interpolating the needed number of samples from the B-spline curve. For the Face Mask, the B-spline surface fit is the very natural choice because of the sculptured shape of the object, and there is no sharp edges, either.

The Alpha_1 solid modeling software we use is running on Sun-4 workstation. The obtained model is is produced both in Alpha_1 modeling language called *r-lisp* and in IGES format. A part of the model description in r-lisp for the Cylindrical Pin is

shown in the Figure 18. The solids are decribed using *Shell* objects in Alpha_1, and *B-spline Solids* in IGES format. Figure 19 shows the obtained Alpha_1 models for the test images. For the Cylindrical Pin multiple control points are inserted where the discontinuities occur. For the Face Mask a 18-by-27 control point mesh was recovered.

7 Conclusion

In this paper we proposed a method for building model data for different CAD and CAM purposes using Three-Dimensional Computer Vision. These techniques are suitable for Reverse Engineering of industrial parts, or can be used as an aid for the designer, especially for designing sculptured free-form surfaces. Rapid prototyping allows also the part analysis and the planning of the manufacturing process in early phase of the desing process, and hence helps reaching the goals of concurrent engineering paradigm.

The proposed method employs surface and volumetric representations to build a CAD model for the part, and to produce useful information for other CAX processes. The nature of the task is quantitative because the emphasis is in the accuracy of the recovered dimensions and shape, whereas object recognition task is interested in classifying an object to be a member of certain category.

We introduced a robust, detail preserving filtering method for noise attenuation. LTS filter produces reliable results even if there is statistical outliers present, or if the noise distribution is not normal. We employed *multiple representations* in the model building phase to be able to deal with both standard geometric shapes and sculptured free-form surfaces. Surface and volumetric representations were used to find out the complexity of the underlying surface, and to reveal potential symmetries. The model building strategy was selected depending on the obtained surface and volumetric primitives and their quality. As a result the method creates a B-Rep CAD model and produces useful information for CAM and CAPP processes. We used B-spline surfaces for CAD model representation. Considerable data compression was gained by determining a relatively small control point mesh that describes a huge amount of

sensor data accurately. The spline description is refined to make the maximum error smaller than a user specified tolerance value. The obtained model data was imported into the Alpha_1 solid modeling system. An IGES file for the model is produced to be able to communicate with other automation subsystems in a standardized way. The experimental results using both real sensor data and synthetic data show that our approach is feasible for reverse engineering industrial parts, both standard geometric shapes and sculptured free-form surfaces.

8 Acknowledgements

We want to thank Dr. Beth Cobb and Prof. Rich Riesenfeld for providing us Alpha_1 system, and for their help and hospitality while studying it. The Academy of Finland is gratefully acknowledged for financial support.

References

[1] "Alpha_1 User's Manual", University of Utah, USA, 1992.

[2] Bajcsy R., Solina F.,"Three-Dimensional Shape Representation Revisited", 1st International Conference on Computer Vision, London, 1987.

[3] Barr A., "Global and Local Deformations of Solid Primitives" Computer Graphics, Vol. 18, No. 3, pp. 21-30, 1984.

[4] Besl P., "Surfaces in Range Image Understanding", Springer-Verlag, 1988.

[5] Besl P., Jain R., "Segmentation Through Variable-Order Surface Fitting", IEEE Transactions on Pattern Analysis and Machine Intelligence", PAMI-10, No. 2, pp. 167-192, 1988.

[6] Besl P., "Geometric Modeling and Computer Vision", Proceedings of the IEEE, Vol. 76, No. 8, pp. 936-958, 1988.

[7] Bhanu B., Ho C., "CAD-Based 3-D Object Representation for Robot Vision", IEEE Computer, August, pp. 19-35, 1987.

[8] Carver G., Bloom H.,"Concurrent Engineering through Product Data Standards", in Control and Dynamic Systems, Vol. 45: Manufacturing and Automation Systems: Techniques and Technologies, Academic Press, pp. 31-109, 1992.

[9] Cohen E., Lyche T., Riesenfeld R.,"Discrete B-Splines and Subdivision Techniques in Computer Aided Geometric Design and Computer Graphics", Computer Graphics And Image Processing, 14, pp.87-111, 1980.

[10] Deriche R.,"Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector", Int. J. of Computer Vision, pp. 167-187, 1987.

[11] Dickinson S., Pentland A., Rosenfeld A.,"From Volumes to Views: An Approach to 3-D Object Recognition", CVGIP: Image Understanding, Vol. 55, No. 2, pp. 130-154, 1992.

[12] Farin G., "Curves and Surfaces for Computer Aided Geometric Design", Academic Press, 1990.

[13] Flynn P., Jain A., "CAD-Based Computer Vision: From CAD Models to Relational Graphs", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 2, pp. 114-132, 1991.

[14] Haralick R., Watson L., Laffey T.,"The Topographic Primal Sketch", Int. J. of Robotics Research, Vol. 2, No. 1, pp. 50-72, 1983.

[15] Hansen C., Henderson T., "CAGD-Based Computer Vision", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 11, pp. 1181-1193, 1989.

[16] "Initial Graphics Exchange Standard, Version 5.1", IGES/PDES Organization, September 1991.

[17] Kak A., Vayda A., Cromwell R., Kim W., Chen C., " Knowledge-Based Robotics", Int. J. Production Res., Vol. 26, No. 5, pp. 707-734, 1988.

[18] Koivunen V., Pietikäinen M., "Evaluating Quality of Surface Description Using Robust Methods", 11th International Conference on Pattern Recognition, The Hague, Netherlands, 1992.

[19] Lee E., "Choosing the Nodes in Parametric Curve Interpolation", Computer Aided Design, Vol. 21, No. 6, pp. 363-370, 1989.

[20] Lee Y., Tantaratana S., "Decision-Based Order Statistic Filters", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 38, No. 3, pp. 406-420, 1990.

[21] Loop C., DeRose T., "Generalized B-spline Surfaces of Arbitrary Topology" Computer Graphics, Vol. 24, No. 4, pp. 347-356, 1990.

[22] Lyche T., Mørken K.,"Knot Removal for Parametric B-spline Curves and Surfaces", Computer Aided Geometric Design, Vol. 4, pp. 217-230, 1987.

[23] Maver J., Bajcsy R., "How to Decide From the First View Where to Look Next", TR MS-CIS-90-39, GRASP Laboratory, University of Pennsylvania, 1990.

[24] Mettala E., "Manufacturing Automation and Design Engineering", DARPA, June, 1992.

[25] Piegl L.,"Modifying the shape of rational B-splines. Part 2: Surfaces", Computer Aided Design, Vol. 21, No. 9, pp. 538-546, 1989.

[26] Pratt V., "Direct Least-Squares Fitting of Algebraic Surfaces", Computer Graphics 21,No. 4, pp. 145-152, 1987.

[27] Press W., Flannery B., Teukolsky S., Vetterling W., "Numerical Recipes in C", Cambridge University Press, 1988.

[28] Solina F., "Shape Recovery and Segmentation with Deformable Part Models", Ph.D. Thesis, MS-CIS-87-111, GRASP Laboratory, University of Pennsylvania, 1987.

[29] Requicha A., Voelcker H., "Solid Modeling: Current Status and Research Directions", IEEE Computer Graphics and Applications, Vol. 3, No. 7, pp. 25-37, 1983.

[30] Rogers D., Fog N., "Constrained B-spline curve and surface fitting" Computer Aided Design, Vol. 21, No. 10, pp. 641-648, 1989.

[31] Rousseeuw P., Leroy A., "Robust Regression & Outlier Detection", Wiley, 1987.

[32] Torre V., Poggio T., "On Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 2, pp. 147-163, 1986.

[33] Tsikos C., "Laser Range Imaging System User's Guide", GRASP Laboratory, University of Pennsylvania, 1989.

[34] Weiss L., Prinz F., Siewiork D., "A Framework for Thermal Spray Shape Deposition: The MD* System", Solid Freeform Fabrication Symposium, Austin, Texas, 1991.

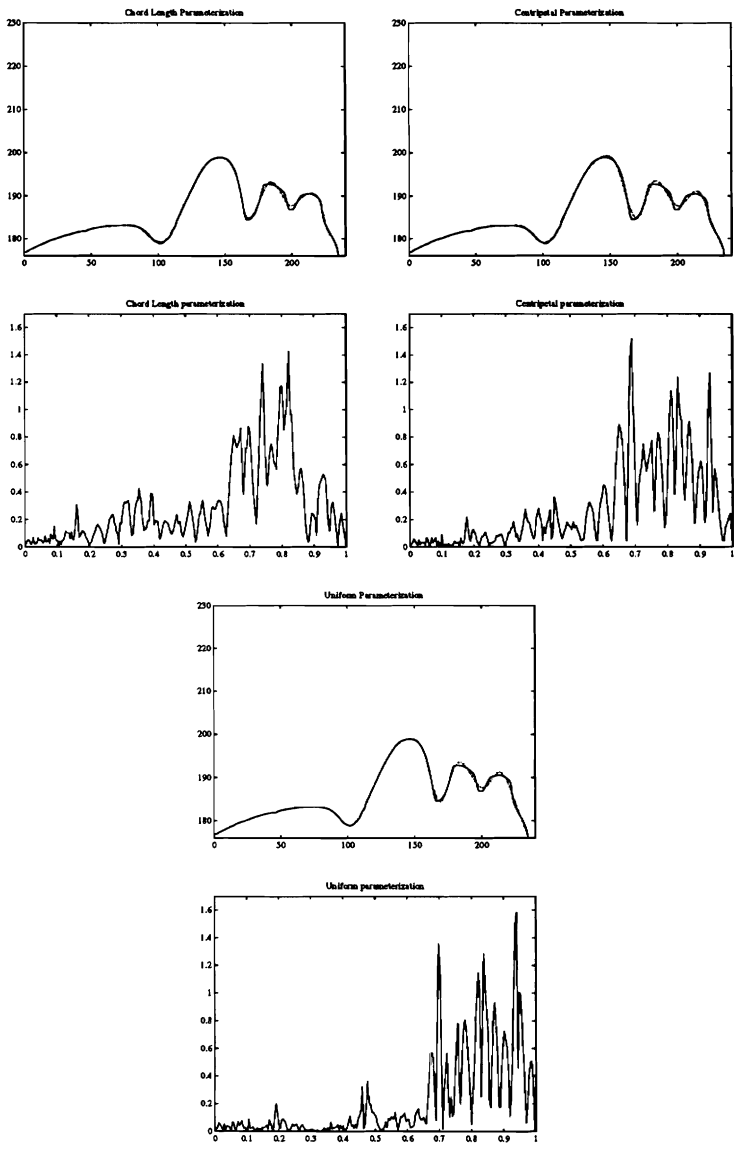


Figure 14: The original range data and the recovered B-spline (dotted line) from the Face Mask image using different parameterization methods: a) Chord Length, b) Centripetal and c) Uniform parameterization. The corresponding Euclidean distances are plotted below each profile.

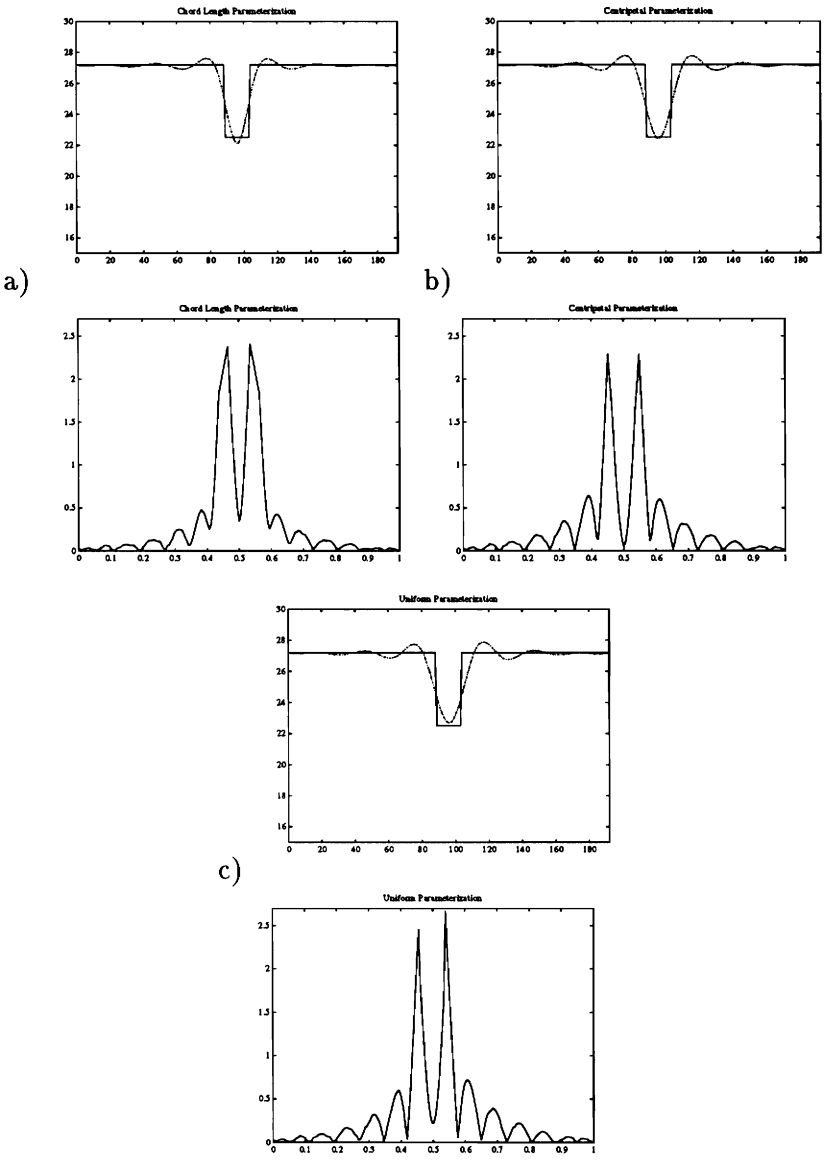
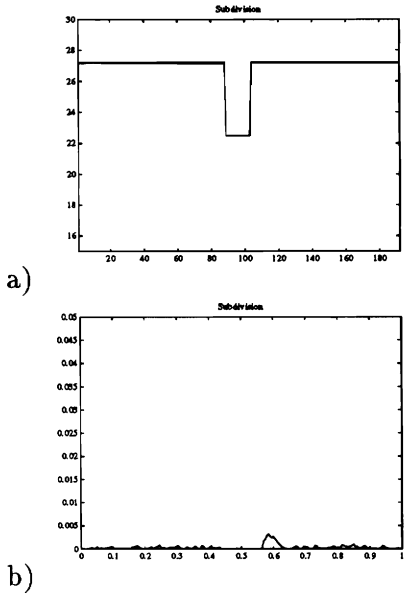


Figure 15: Errors introduced by discontinuities for different parameterization methods without subdivision: a) Chord Length method, b) Centripetal method and c) Uniform parameterization. The corresponding error distances are shown below each profile.



a)

b)

Figure 16: B-spline subdivision allows representing discontinuities: a) The obtained profile from the Cylindrical Pin after subdivision, and b) the corresponding error distances.

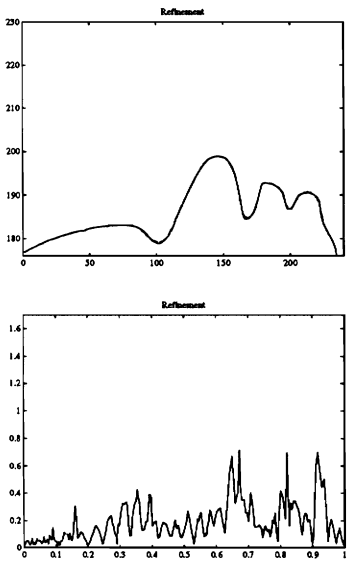


Figure 17: The obtained B-spline for the profile from the Face Mask and corresponding error distances after the refinement process.

```
                    .
                    .
                    .
p12 := projPt(0.000000,80.61196,27.174069,1.0);
p13 := projPt(0.000000,100.26872,27.123658,1.0);
p14 := projPt(0.000000,120.01186,27.189411,1.0);
p15_1 := projPt(0.000000,129.88000,27.140000,1.0);
p15_2 := projPt(0.000000,129.88000,27.140000,1.0);
p15_3 := projPt(0.000000,129.88000,27.140000,1.0);
p15_4 := projPt(0.000000,129.88000,27.140000,1.0);
p16 := projPt(0.000000,129.88000,0.00000,1.0);
};

{
Revcurv := curve(parminfo(cubic,ec_open,kv_uniform),
list( p0, p1_1, p1_2, p1_3, p1_4, p2, p3, p4, p5_1, p5_2,
p5_3, p5_4, p6_1, p6_2, p6_3, p6_4, p7, p8, p9, p10_1,
p10_2, p10_3, p10_4, p11_1, p11_2, p11_3, p11_4, p12, p13, p14,
p15_1,p15_2,p15_3,p15_4, p16));
RevSurf := srfofRevolution(Yaxis,Revcurv,nil,nil);
Modsolid := shell(RevSurf);
};
```

Figure 18: A part of the model description for the Cylindrical Pin using Alpha_1 modeling language.

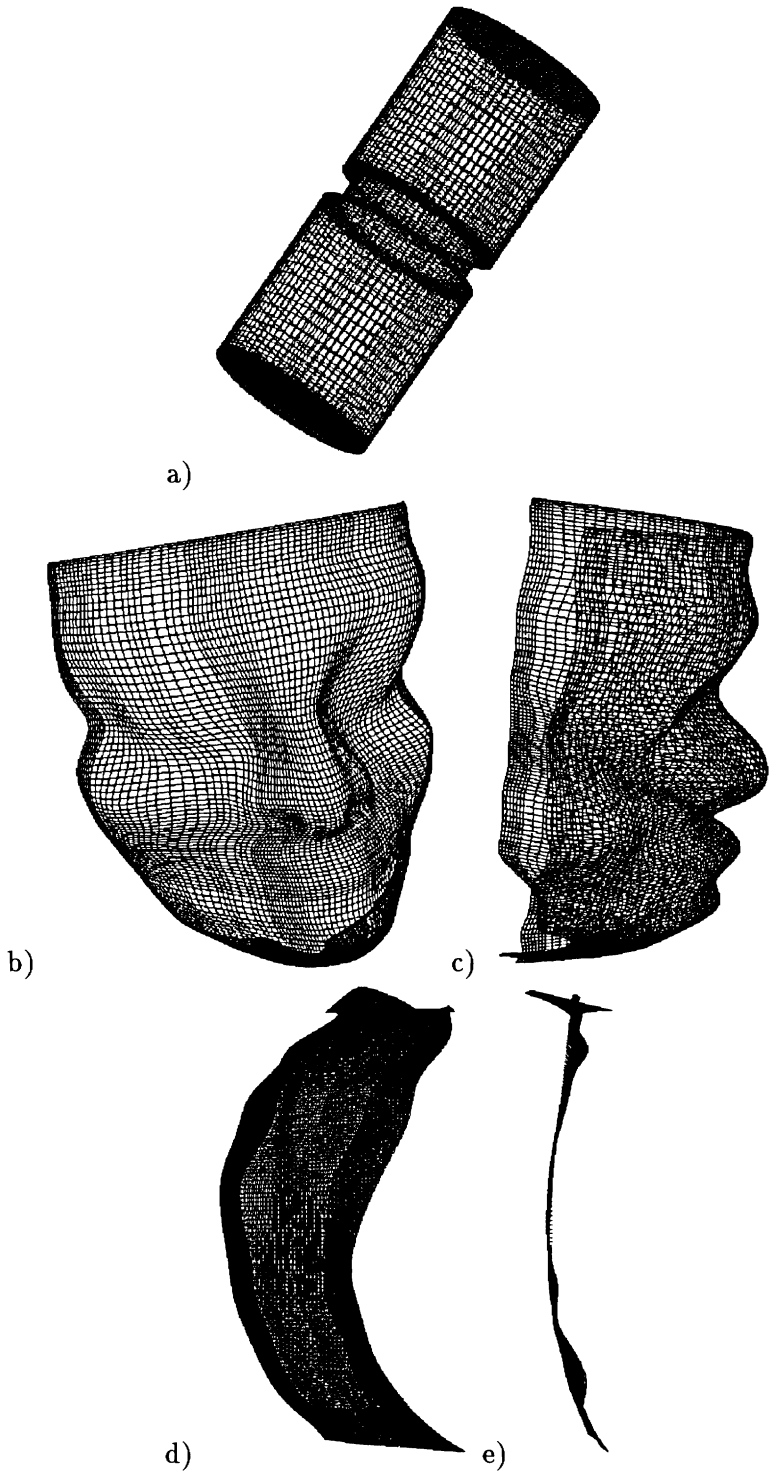


Figure 19: Graphical model data for a) the Cylindrical Pin, b) the Face Mask , and c) its side view, d) for the Fan Blade, and e) its side view, respectively.