July 1993

Multiple Representation Approach to Geometric Model Construction From Range Data

Visa Koivunen
University of Pennsylvania

J. Vezien
University of Pennsylvania

Ruzena Bajcsy
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-93-66.

Multiple Representation Approach to Geometric Model Construction From Range Data

Abstract

A method is presented for constructing geometric design data from noisy 3-D sensor measurements of physical parts. In early processing phase, RLTS regression filters stemming from robust estimation theory are used for separating the desired part of the signal in contaminated sensor data from undesired part. Strategies for producing a complete 3-D data set from partial views are studied. Multiple representations are used in model construction because there is no single representation that would be most appropriate in all situations. In particular, surface triangulation, NURBS, and super-ellipsoids are employed in order to represent efficiently polygonal and irregular shapes, free form surfaces and standard primitive solids. The size of the required control point mesh for spline description is estimated using a surface characterization process. Surfaces of arbitrary topology are modeled using triangulation and trimmed NURBS. A user given tolerance value is driving refinement of the obtained surface model. The resulting model description is a procedural CAD model which can convey structural information in addition to low level geometric primitives. The model is translated to IGES standard product data exchange format to enable data sharing with other processes in concurrent engineering environment. Preliminary results on view registration using simulated data are shown. Examples of model construction using both real and simulated data are also given.

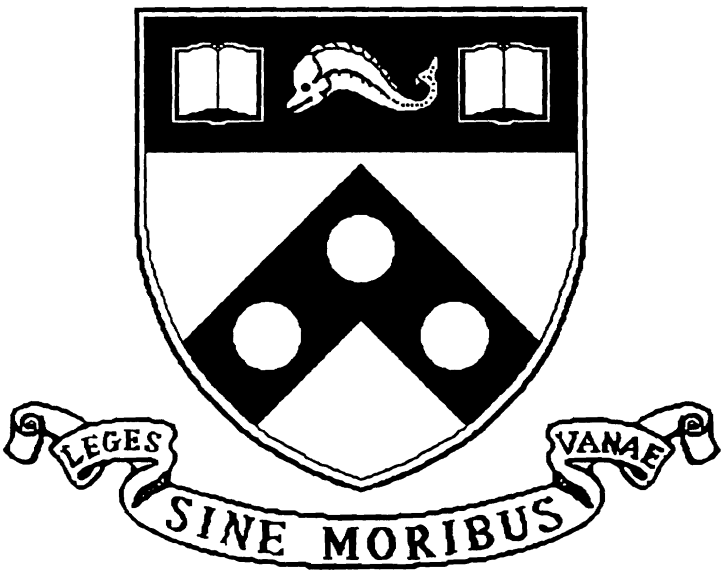
Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-93-66.

Multiple Representation Approach To Geometric Model Construction From Range Data

MS-CIS-93-66
GRASP LAB 352

V. Koivunen
J. Vezien
R. Bajcsy



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

July 1993

Multiple representation approach to geometric model construction from range data

V. Koivunen, J. Vezien and R. Bajcsy

General Robotics and Active Sensory Perception

(GRASP) Laboratory

University of Pennsylvania

300C 3401 Walnut Street

Philadelphia, PA 19104-6228

Abstract

A method is presented for constructing geometric design data from noisy 3-D sensor measurements of physical parts. In early processing phase, RLTS regression filters stemming from robust estimation theory are used for separating the desired part of the signal in contaminated sensor data from undesired part. Strategies for producing a complete 3-D data set from partial views are studied. Multiple representations are used in model construction because there is no single representation that would be most appropriate in all situations. In particular, surface triangulation, NURBS, and superellipsoids are employed in order to represent efficiently polygonal and irregular shapes, free form surfaces and standard primitive solids. The size of the required control point mesh for spline description is estimated using a surface characterization process. Surfaces of arbitrary topology are modeled using triangulation and trimmed NURBS. A user given tolerance value is driving refinement of the obtained surface model. The resulting model description is a procedural CAD model which can convey structural information in addition to low level geometric primitives. The model is translated to IGES standard product data exchange format to enable data sharing with other processes in concurrent engineering environment. Preliminary results on view registration using simulated data are shown. Examples of model construction using both real and simulated data are also given.

1 Introduction

In this paper we present an approach for integrating an intelligent sensory system into a part of design automation system. Solid modelers could benefit geometric models

constructed automatically and rapidly from 3-D sensory data. Such tools are useful as a design aid, especially for modeling free form shapes which is a very time consuming design task by hand, and requires extensive knowledge about the modeling tools, such as splines. Sometimes no design data exists for an old part and the redesigning could be done by reverse engineering the part from sensor measurements. Customizing is also often needed, and it is desirable to keep the unit price affordable although the number of parts to be produced is small. The analysis of the part and process planning could also be started in very early phase of the design process using the initial model constructed from the sensor measurements.

The two main research problems we are facing in the CAD model construction from 3-D sensor data are:

1. Data acquisition and combination of the partial data sets into a complete 3-D data set.

2. Data interpretation by fitting models.

The first problem requires estimation of relative rotation and translation between data set obtained from different vantage points and combination of all data into one common coordinate frame. The goal of data interpretation is to produce a geometric model of a part to be imported into a solid modeling system. Similarly to Computer Aided Geometric Design (CAGD), there is no single method or representation in Computer Vision that would be appropriate in all situations. Therefore, we employ *multiple representations* in model construction. The produced geometric model should be compatible with common representations in modeling systems in order to analyze and simulate the model and share it with other automation subsystems. The designer should also be able to modify the model because the design typically evolves.

Our approach constructs *procedural CAD models*, which are procedures that generate the part geometry. Procedural models are able to represent low level geometry of the part as well as its overall structure. Structural information is vital for analysis, simulation and process planning and it must be detected by these processes if not provided by the geometric model. Moreover, procedural models are useful in representing intersections of surfaces, for example, in the case of trimmed parametric surfaces. The

intersection is described in the procedure and it can be approximated in the level of required accuracy when it is actually needed. The designer is also able to modify the procedure, if necessary.

The capability to communicate between different subsystems during the design process is a prerequisite for concurrent engineering. The data sharing is provided by standard product data formats, such as IGES [5, 12]. The proposed system is depicted as a part of concurrent engineering environment in Figure 1.

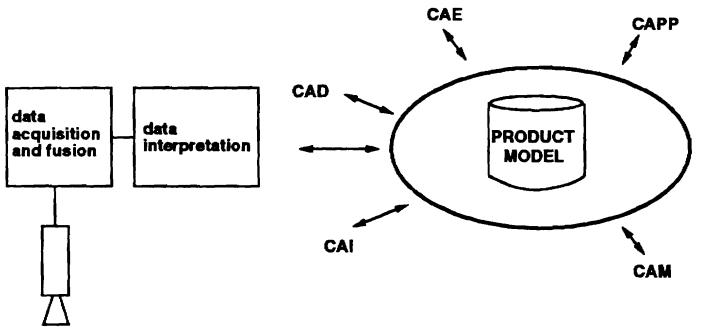


Figure 1: The proposed system as a part of a concurrent engineering environment. The CAX processes are Design, Engineering, Process Planning, Manufacturing and Inspection.

The organization of this paper is as follows. In section 2 we address data acquisition, view registration and integration problems. In section 3 we describe briefly shape representations used in model construction. In section 4 we show some examples using real and simulated range data. Finally in section 5 we summarize and discuss some areas requiring future research.

2 Data acquisition

We chose to use optical non-contact sensors for measuring 3-D shape of the objects. Coordinate Measuring Machines (CMM) were not considered because of their low speed in acquiring data from free form shapes which require dense measurements.

Physical measurements are prone to errors. In the case of range data, the actual noise distribution differs from the nominal one which is often Gaussian. Furthermore, there may occur outliers due to the orientation of material of the surface or because

of other statistical populations present in the processing window. The raw data is filtered using RLTS [13, 14] robust regression filters in order to recover the structure of the underlying signal and reject outliers which may cause incorrect estimates in model building processes.

In general, optical non-contact active range data acquisition techniques provide incomplete 3-D information because the signal does not reach all the surface points if the data is obtained from one viewpoint at the time. A complete 3-D data set has to be merged from a collection of images from different viewpoints. The rotation groups of regular polyhedra, as noted in [4], provide a convenient set for uniformly sampling the observation sphere. Therefore, the scanning procedure should use such a set of evenly distributed viewpoints as a default, if no symmetry is obvious.

In order to combine multiple range views into one complete 3-D data set the *registration*, i.e., the relative rotation and translation between the views, must be estimated, and the *integration* of the views into nonredundant data set in a common coordinate frame performed. Recent overviews of the research on view registration and integration methods are given in [30, 4].

2.1 Registration and integration

2.1.1 Background

The registration estimates the relative transformations between different views and transforms all the data into a common coordinate frame. Typically, methods assume either that the transformation is known, or corresponding features are detected reliably from each view and subsequently the transformation can be solved accurately. In the latter case, the features are a set of *a priori* known reference points from the environment that are visible in different views, or features extracted from object surfaces. This approach is adequate in simple situations where the object consists of relatively few geometric primitives that can reliably detected from different viewpoints. In the case of sculptured free-form shapes, however, it is difficult to establish correspondencies.

We chose to adapt the Iterative Closest Point (ICP) algorithm proposed by Besl and McKay [4] for registering a single view with a known, computer-generated database.

It was chosen because no feature-to-feature correspondencies are required, it is computationnally efficient and independent from data representation, as long as a method for computing point/primitive distances exists. Its main shortcoming is obviously that only a locally optimal displacement is found.

The method matches a collection of points from one set of raw data with a set of primitives from a model. Each point is basically associated to its closest primitive, the type of primitive defining the exact distance measuring function. Given a model $\mathcal{M}= \{m_k\}$ containing M primitives, let $\mathcal{P}= \{\vec{p_i}\}$ be the set of N points. We'll see how to choose those points in the next section. $\mathcal{X}= \{\vec{x_i}\}$ is then defined as the "projection" of $\mathcal{P}$: $\vec{x_i}$ is the closest point of the closest primitive m_{k_i} to $\vec{p_i}$. Once a match $\mathcal{P} \rightarrow \mathcal{X}$ is established, an optimal displacement, in the form of a registration state vector $\vec{q}$, which consists of a rotation quaternion $\vec{q_R}$ and a translation vector $\vec{q_T}$, can be computed. The mean square error function [4]:

$$e(\vec{q}) = \frac{1}{N} \sum_{i=1}^{N} \|\vec{x_i} - R_{\vec{q_R}}\vec{p_i} - \vec{q_T}\|^2$$

is minimized. Let's call $\vec{q_0}$ the state vector solution. $\vec{q_0}$ is applied on the set $\mathcal{P}$ and the closest primitive search is iterated on the displaced set, leading to a new $\mathcal{P} \rightarrow \mathcal{X}$ matching. The consecutive displacements $\vec{q}$ obtained are cumulated into a final solution $(\vec{q_R}, \vec{q_T})$, until the error measure $e(\vec{q})$ is less than a certain threshold.

2.1.2 Constrained solution

The view registration problem in model construction context is more complicated because there is no *a priori* known part model. Hence, we have to incrementally build the internal representation used as the part model, by adding each view after registration instead. On the other hand, we obtain an approximate transformation between the views because the movement of sensors or objects are controllable. Such approximate transformation can serve as a good first estimate for the ICP procedure in order to find a global minima instead of a local one. Furthermore, it provides a way to exclude points from the matching process if they are not visible in the representation registered so far.

The main problem is that, using a given set $\mathcal{P}_k$ extracted from range image k, some

points $\vec{p_i}$ may not appear in the model $\mathcal{M}$ compiled so far (from images $0 \rightarrow k-1$). In that case, some matches in the correspondence $\mathcal{P} \rightarrow \mathcal{X}$ will always be erroneous, leading to a wrong estimation of $\vec{q}$ as pointed out in [4].

Fortunately, if a reasonable estimate of $\vec{q}$ is known prior to the beginning of the registration procedure, overlapping points will produce relatively small errors, whereas mismatched points often lead to large residual errors, and can then be detected by analyzing the distribution of the errors:

$$\|\vec{x_i} - R(\vec{q}_R)\vec{p_i} - \vec{q}_T\|^2,$$

where $\vec{q}$ is the state vector applied to the data in the previous iteration. To further separate outliers from valid points, we check for the consistency of the normals after displacement:

$$\vec{n_{x_i}} \cdot R_{q_R}(\vec{n_{p_i}}) > cos(\alpha)$$

where $\vec{n_u}$ is the normal estimated at point $\vec{u}$, and α a threshold angle. The normals can be estimated by using local window operators or robust estimation techniques if the image is noisy [13]. Non-valid points display large consistency errors and can be discarded at each iteration: At iteration k, the displacement $\vec{q}$ is recomputed until no point violates the normal consistency constraint. α is kept large to maintain flexibility in the matching process and insure convergence. Also it is clear that such a method will fail if gross outliers (non overlapping data) are predominant in the $\mathcal{X}$ set and the first estimation of $\vec{q}$ prior to refinement is wrong.

Another adaptation concerns the choice of points and primitives. Many types of primitives are possible, depending on the geometry of the scanned surface. Preliminary tests have been performed successfully using large planar patches. We are now investigating the case of free form objects using Delaunay triangulated [11] surface representation addressed later in shape representation section. To run the ICP-based algorithm on triangulated data, we select the set $\mathcal{P}$ as the centers of gravity of the triangles in one image (a low resolution is chosen, so that typically less than a hundred points are selected). Those points are registered against a triangulation of the existing model, constructed with the previous registered images, the first image being the starting point. Eventually a complete 3D data set will be incrementally constructed.

A more thorough description of the constrained registration is given in [31].

2.1.3 Integration

The integration part combines registered and partly overlapping data sets into a complete nonredundant 3-D data set. The methods typically perform a (weighted) averaging of the samples that are in the overlapping parts of the views. The level of integration and representation of the data have to be chosen as well. The approaches are either data driven, such as [29] where the surfaces are represented using low level primitives, e.g., triangular mesh, or model driven, where more elaborate model surfaces are employed and the surface parameters are adjusted to fit to the data. An example of latter approach is given in [6], where initial triangulated ellipsoid or icosahedron is refined to approximate the bounding surface. Other possible approaches are described, for example in [19, 27]. In our case, a data driven integration part is under development.

3 Issues on shape representation

The type of shape representation have to be chosen based on not only the family of shapes we are describing but also on the task where the representation is used. In object recognition from single arbitrary viewpoint, properties such as viewpoint invariance and uniqueness are important. In geometric modeling, however, the representation have to be unambiguous but not necessary unique and the geometry is typically described in object centered coordinate frame.

Constructive Solid Geometry (CSG) and Boundary Representation (B-rep) are widely used in CAGD [23]. CSG models standard primitive solids effectively but modeling of sculptured free form surfaces is difficult. B-rep defines a solid by its bounding surfaces. Polygonal representation is commonly used for modeling flat surfaces. Non Uniform Rational B-splines (NURBS) are widely used for modeling free form shapes, and they are also able to represent conics and quadric surfaces exactly. Sweep methods are often used to design solids that have rotational or translational symmetry [23]. The "Design by Features" paradigm [26] uses manufacturing features to create the

part geometry which is very appealing approach because the design is often done for manufacturing. It seems, however, that there is no single representation or method that would be best for every design task. Therefore, several CAD-systems are hybrid systems, i.e., use multiple representations to provide efficient tools for different design tasks [23].

The representations used for CAD-based computer vision can be generally classified into volumetric, surface and sweep representations [3]. Volumetric methods are actually surface based methods because surface evaluation is required to recover volumetric description. Such methods can represent only closed surfaces, hence they are not suitable for describing partial information. Surface based methods represent the part geometry typically by a set of bounding low order surface patches. Generalized cylinders (GC) are a typical sweep representation. It seems, as in the design, that there is no single representation that would recover an appropriate description from sensor data in all situations.

In order to model different shapes we are employing multiple representations. An optimal triangulation is generated for modeling polygonal and irregular shapes and surfaces of arbitrary topology. Triangulation includes very little structural information about the part but it can be used as a worst case representation, if no other method is suitable. NURBS are used for modeling free form surfaces because of their continuity and local control properties. Furthermore, they are included in IGES product data exchange standard which facilitates data sharing and concurrent engineering. Superellipsoid models are used to detect overall part structure which allows us to use more efficient model primitives that are helpful in part analysis and process planning.

3.1 Triangulation

In order to describe polygonal and irregular surfaces and surfaces of arbitrary topology we construct a collection of triangles (triangulation) describing the surface. We require that the distance between any point of the object surface and its projection onto the triangulation along the surface normal is less than a user defined tolerance value. Triangulation methods are widely used in approximation theory, finite element analysis,

and CAD [27].

A *k/n-triangulation* is a collection of k-simplices in n-dimensional space. A triangle is a 2-simplex and a tetrahedron is a 3-simplex, for example. In general, triangulations where triangles are nearly equilateral are desirable. In particular, we are generating a Delaunay triangulation which satisfies the property that the interior of the minsphere of a k-simplex contains no vertex of any k-simplex. The minsphere is the smallest $(n-1)$-dimensional sphere which passes through $k+1$ vertices of a k-simplex. In two-dimensional case, the minsphere property means that the minimum angle is maximal over all triangulations.

The method applied here follows the Generalized Delaunay Triangulation procedure presented in [11] with $k = 2$ and $n = 3$. The method starts with one k-simplex and new triangulation points are inserted one at the time. Different insertion operations are executed based on the location of insertion point relative to the triangulation. The process continues until all points have been inserted or or the remaining points cannot be inserted without violating the empty minsphere property.

The triangulation is refined based on Euclidean distances between points and triangles by adding and removing points to meet the user defined tolerance value *tol*. For point removal as well as point addition, all is needed is to compute the distance between a point and a triangle in 3D. A point is candidate for addition if the distance d to the nearest triangle T is $>$ *tol*. The list of points to be added is obtained by sampling the input data at various resolution, starting at a coarse level, and interating until no point can be added. Between each resolution, the triangulation vertex list is scanned for possible removals: For each point p in the triangulation, let $\mathcal{T}$ be the set of 2-simplices sharing p. p is tentatively removed, and $\mathcal{T}$ is retriangulated into $\mathcal{T}'$. p is definitively removed only if the distance of p to $\mathcal{T}'$ is lesser than *tol*.

Given a triangle $T = (\vec{p_1}, \vec{p_2}, \vec{p_3})$, the distance between a point $\vec{p}$ and primitive T is given by:

$$d(\vec{p}, T) = \min_{a_1+a_2+a_3=1} \|a_1\vec{p_1} + a_2\vec{p_2} + a_3\vec{p_3} - \vec{p}\|$$

with the weighting coefficients $a_1 \in [0,1]$, $a_2 \in [0,1]$, $a_3 \in [0,1]$. In practice, one projects $\vec{p}$ on the plane embedding T, and the projection $\vec{p_\perp}$ should be inside T. When the triangulation is generated, one can easily compute the equation of the edges of T.

Let's then call $d_i(\vec{v})$ the algebraic distance between any point $\vec{v}$ and edge i (i is the edge which doesn't contain $\vec{p_i}$). To perform the interiority check, one has only to compute $d_i(\vec{p_\perp})$ ($i \in \{1,2,3\}$), then tests:

$$0 < d_i(\vec{p_\perp}) < d_i(\vec{p_i}) \quad \text{for} \quad i \in \{1,2,3\}$$

provided that $d_i(\vec{p_i}) > 0$ (otherwise, reverse the sign of the inequalities). The $d_i(\vec{p_i})$ are already computed and stored during the triangulation process.

3.2 Spline approximation

B-spline surfaces have several desirable properties for geometric modeling and high quality surface approximation, such as local control and continuity. B-spline surfaces lie within the convex hull formed from the control point mesh. It is important to have sufficient number of control points to be able to describe all the degrees of freedom of the underlying surface. If there are too few control points the fit is not likely to converge and on the other hand, the fewest number of control points tends to yield the fairest surfaces [25]. An initial estimate for the B-spline control point mesh size is computed by using the maximum number of geometrically homogeneous patches in each parameter direction. The surface patches are detected by a local characterization process. If the order of B-splines is three (degree=2), we need at least three control points in each parameter direction to be able to describe each second order surface patch.

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

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

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 identify the number of control point vertices in each direction. The basis functions $N_{i,k}$ of order k are defined recursively as follows:

$$N_{i,1}(u) = \begin{cases} 1 & \text{, if } x_i \le u \le x_{i+1} \\ 0 & \text{, otherwise} \end{cases} \tag{2}$$

$$N_{i,k}(u) = \frac{(u - x_i)N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}, \quad (3)$$

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 maximum of 4th order ($k = l = 4$, cubic) B-splines to make the approximation less sensitive to small local variations.

Chord length parameterization is employed. The parameter values are normalized to $[0, 1]$ range. An open end condition is used to force the spline to begin exactly from the first control point and end at the last control point.

The locations of the control points of the approximating B-spline surface are computed by minimizing errors in least squares sense. Now we have to solve $B_{i,j}$'s from equation (1), and $S(u, v)$'s are the measured data points. All the weights are originally set to 1.0 because $S(u, v)$'s are physical measurements. The weights $h_{i,j}$ of the control points can be adjusted later in surface refinement [20]. Using matrix representation the solution is:

$$[B] = [[C]^T[C]]^{-1}[C]^T[S], \quad (4)$$

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 point mesh.

The accuracy of the approximation should meet a user given tolerance value. The error of the approximation is defined as the Euclidean distance between the measured and the approximated surface with same (u, v) parameter values. There are different approaches for controlling the shape the B-spline surface, see [8, 17, 20, 25]. We start with a good estimate of the appropriate control point mesh size and add knots if the distance exceeds the given tolerance value. Adding a certain number of knots has a consequence of adding the same number of control points. Curve or surface discontinuities can be introduced by inserting a knot with multiplicity equal to the order of the B-spline.

Tensor product B-spline surfaces require a rectangular parametric grid which maps the coordinates in (u, v) parameter space to physical (x, y, z) coordinates. Resampling may be necessary to get a rectangular arrangement of the data. In the case of scattered samples, for instance, the points are typically organized into triangular faces and then

the contours, e.g., isoparametric lines, are interpolated [19].

It is not possible to describe certain surfaces of arbitrary topology with single nondegenerate B-spline. Designers can, however, can introduce degeneracies into the mesh by reducing the control points forming an edge of the mesh into a single point or use non-tensor product patches [16]. The surfaces can then be represented, for example, by using Gregory patches or Rational Boundary Gregory (RBG) patches [7] which are joined together. The continuity properties at the joints are relaxed into geometric tangent plane (G^1) continuity. In order to transfer such data to CAD systems using IGES [12] product data exchange format the representation have to be converted to NURBS.

A simple engineering solution for situations where the rectangular arrangement of tensor product surfaces is not appropriate 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" [10]. In our approach, the boundaries of the surface are used to compute trimming curves which divide it into invalid and valid parts. The intersection curves of parametric surfaces are computed using subdivision based techniques [8]. The fit procedure is run using a bounding box for the object, and the parts of the surface which are not on the object surface are declared invalid. Trimmed surfaces are included in IGES standard and advanced solid modelers [12, 1].

3.3 Superellipsoid model recovery

Superellipsoids are a subclass of superquadrics [2] that can represent shapes ranging from ellipsoids to cuboids and cylindroids. A superellipsoid surface in nonparametric implicit form is defined as follows [28]:

$$f(x,y,z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\varepsilon_1}} = 1, \tag{5}$$

where a_1, a_2, and a_3 define the size in x-, y- and z-axis direction. ε_1 and ε_2 are the shape (squareness) parameters in the latitude and in the longitude plane, respectively. Additional parameters are employed to describe global deformations. The Levenberg-

Marquartd method [22] is used for minimizing the following expression [28]:

$$\min \sum_{i=1}^{N} [\sqrt{a_1 a_2 a_3}(F(x_i, y_i, z_i) - 1)]^2, \tag{6}$$

where the function $F(x, y, z) = f(x, y, z)^{\varepsilon_1}$ determines the locus of a point relative to superellipsoid surface and N is the number of samples. The range of the shape parameters is constrained to $0.1 \leq \varepsilon_1, \varepsilon_2 \leq 2.0$. The goodness of fit is defined as $GOF = \sqrt{a_1 a_2 a_3}((\sum_{i=1}^{N} |F(x, y, z) - 1|)/N)$.

The superellipsoid model recovery is used to detect typical primitive solids shapes and overall part properties such as symmetry. Rotationally symmetric shapes are constructed using surface of revolution design primitive whereas translationally symmetric shapes can be generated by extrusion. The primitive solids [18] and the approximating superellipsoid shape parameters are depicted in Table 1. The nature of the obtained

Table 1: Primitive solids, the corresponding CAD model parameters, and the approximating superellipsoid shape parameters. "-" means that in general the primitive can not be recovered using the superellipsoid model we employ.

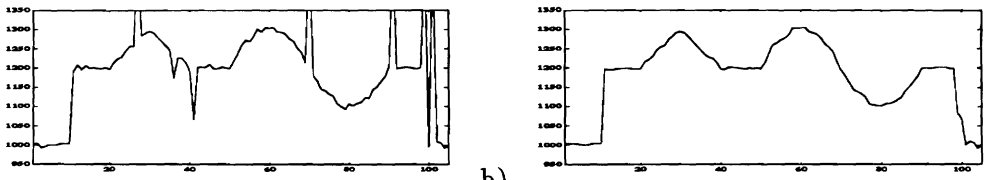
Primitive solid	Model parameters	Shape parameters/remarks
sphere,ellipsoid	radius, major and minor axis	$\varepsilon_1 = 1.0, \varepsilon_2 = 1.0$
parallelepiped	length, width, height	$\varepsilon_1 \ll 1.0, \varepsilon_2 \ll 1.0$ or $\varepsilon_1 = 2.0, \varepsilon_2 \ll 1.0$
cylinder	height, radius, normal plane	$\varepsilon_1 = 1.0, \varepsilon_2 \ll 1.0$
cone	base circle, vertex point	cylinder + tapering
torus	major and minor radius, initial plane	-, supertori
wedge	length, width, height	only a subset, parellelepiped + tapering
fillet	length, width, height	-
polyhedron	vertex points	-
any collection of halfspaces	regularized intersection of surfaces	-

superellipsoid parameters is qualitative and they can be used as a hypothesis to invoke the appropriate model building procedure. If the shape parameters indicate that the part is a natural quadric, a more accurate description is obtained by fitting quadric model to the data [21]. In the case of a surface of revolution, the actual model building process fits conic sections along the assumed axis of symmetry to recover the rotation axis and the profile NURBS curve accurately.

4 Experimental results

In the experimental part we show preliminary results on view registration using simulated data. Moreover, model construction results are shown using simulated and real sensor data emphasizing the need for multiple representations.

The images are filtered using robust RLTS filtering to recover the structure of the original signal and reject outliers. Very deviating observations would cause serious errors in model construction which employs least squares error norm in fitting. Figure 2 shows filtering results using a 5 point processing window for a simulated sample profile where Gaussian noise with $\mu = 0$ and $\sigma = 5.0$ and random bit error with probability $P = 0.015$ are added to the noise-free signal. A 12 bit quantization is used.



a) b)

Figure 2: Filtering results for noisy signal: (a) The noisy signal, and (b) the RLTS filtered signal, respectively.

Registration estimates the relative rotation and translation between the views. In our preliminary experiments, the ICP algorithm is used to minimize the distances of the points to planar surface patches. A region-growing algorithm uses surface normal consistency to generate a planar approximation of the part. Small patches that occur typically in the vicinity of C_0 and C_1 discontinuities are discarded as unstable and irrelevant. The centers of gravity of the remaining planar areas are taken as reference points $\vec{p_i}$, to be matched against the planar primitives extracted in a contiguous viewpoint, using the ICP based registration scheme. A simulation using a computer-generated polyhedral object in eight different poses is shown in Figure 3.

The final registration is shown with cross-section along the x,y and z axis of the object (Figures 4 and 5). The algorithm was here able to recover the right displacement in all cases with good accuracy: 1/4 pixel accuracy in pixel to pixel registration between views. Apparently, less accurate results are obtained using real data. Furthermore, one

has to take account the actual amount of overlap between views. The more complicated the data, the more occlusions, and the higher the number of viewpoints is required to obtain a complete data set.

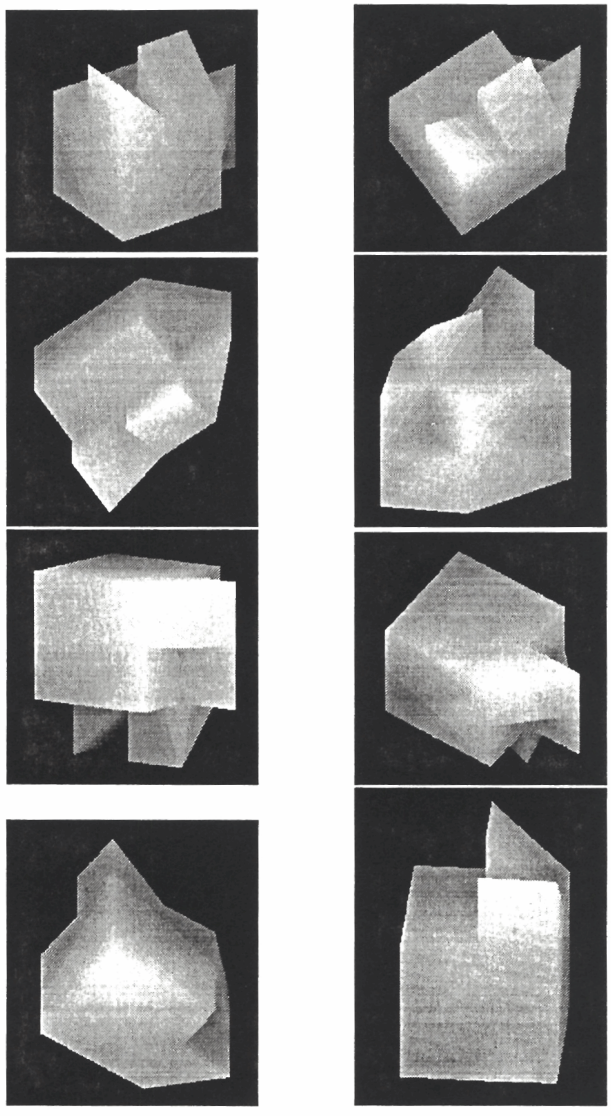


Figure 3: Eight views of the same simulated object.

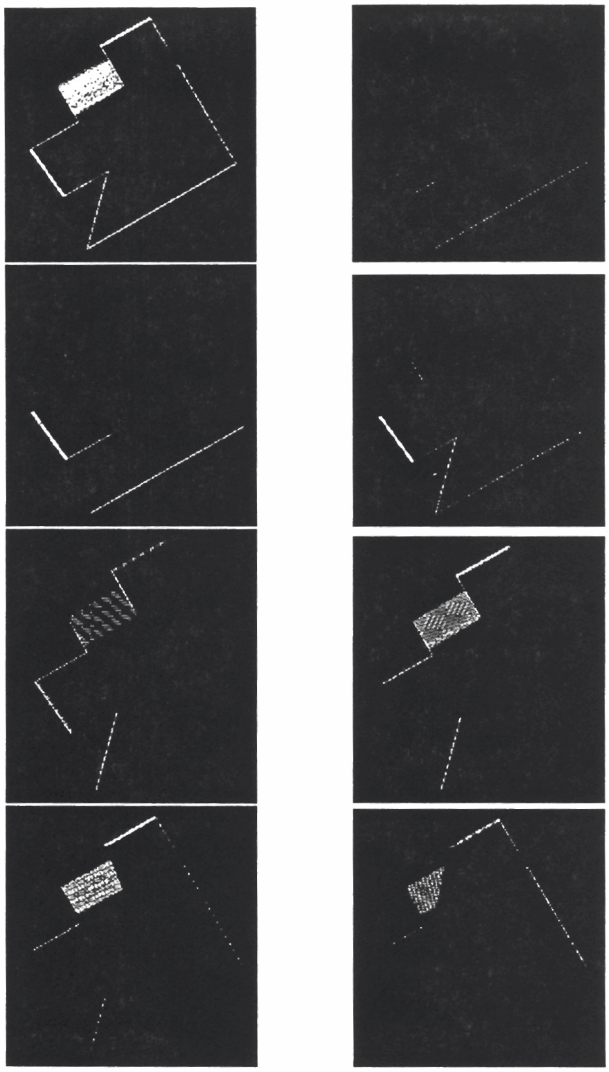


Figure 4: The registration result is illustrated by taking a cross-section of the object by keeping X coordinate (= 1.3) constant. At the top is the complete data set, the other images represent the contributions of each view. The white area is due to the cutting plane grazing of one of the object surfaces.

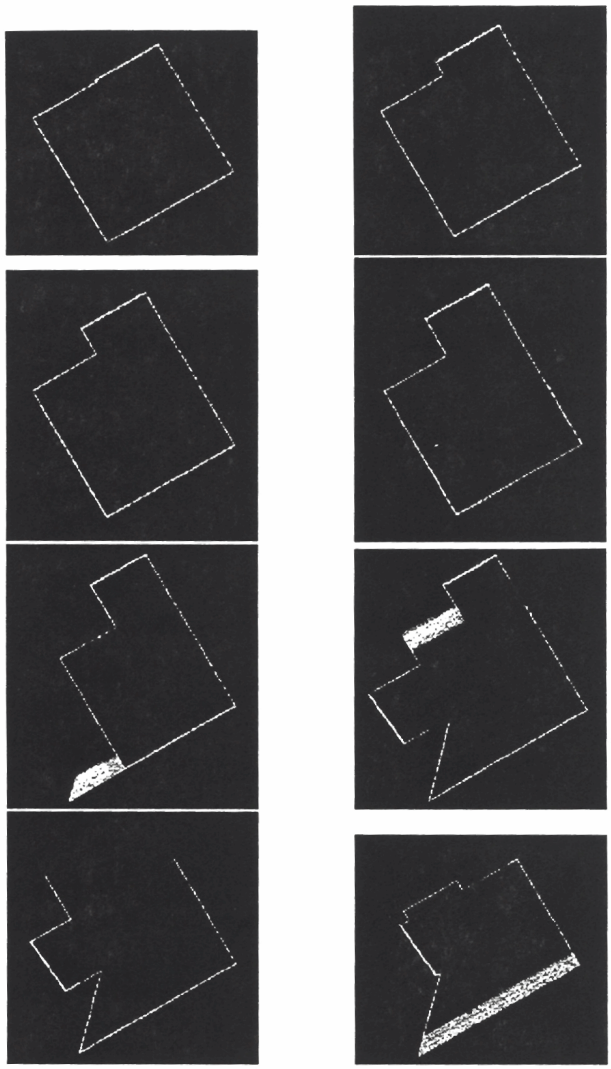


Figure 5: A collection of isocontours along the X axis ($1.05 < X < 1.5$). For each isocontour, data were collected from all the views using the registrations obtained via the ICP algorithm. The first view was used as a reference frame.

The model construction examples are given using the data sets depicted in Figure 6. The Cylindrical Pin data is synthetic data where Gaussian distributed noise with zero mean and $\sigma = 4.0$ as well as random bit error noise with probability $P = 0.001$ are added. The image resolution is 256-by-256 pixels and 16 bit quantization is used for the depth values. The Face Mask image and the Hand image from NRCC [24] range image library are chosen to demonstrate the capability to model free-form surfaces and surfaces of arbitrary topology.

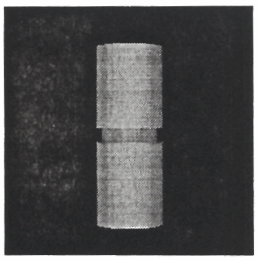
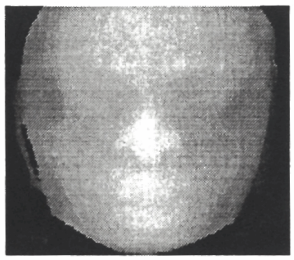
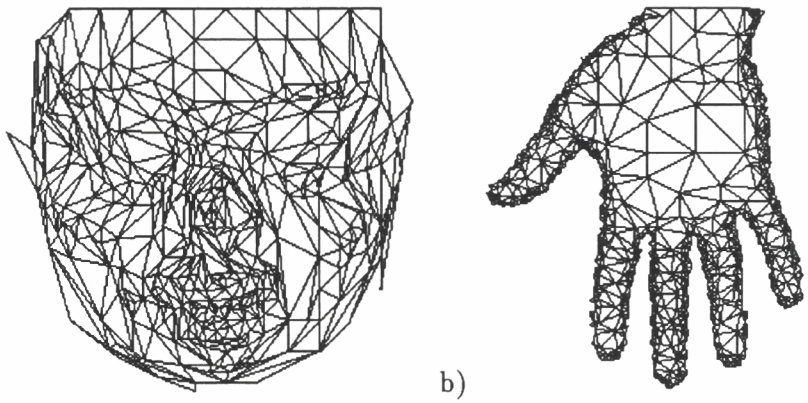
a)  b)  c) 

Figure 6: Test images: a) The Cylindrical Pin is a synthetic range image produced from a CAD model, b) the Face Mask and c) the Hand are real range images from NRCC range image library.

The surface triangulation is produced in order to model surfaces of arbitrary topology, polygonal shapes, and irregular shapes that do not consists of smooth surfaces typically found in manufactured objects. Furthermore, such representation is useful in view registration and integration. Triangulation does not convey much structural information about the part geometry. The starting point of the triangulation was chosen on the center of gravity. The triangulations were refined by adding or deleting points until a tolerance value was achieved. In the case of the Face mask, 724 triangles were needed to meet the 0.5 mm tolerance value for the description. The Hand data required 1260 triangles with tolerance value 0.4 mm. Triangulations of the Face Mask and Hand are depicted in Figure 7.

The superellipsoid model recovery is used to reveal global shape properties. The obtained shape parameters are used as a hypothesis to invoke the appropriate model building procedure. Table 2 shows the recovered shape parameters and the quality of the fit measures for the test images. The shape parameters reveal the rotational symmetry of the Cylindrical Pin. The quality of the fit is also high, hence surface of



a) b)

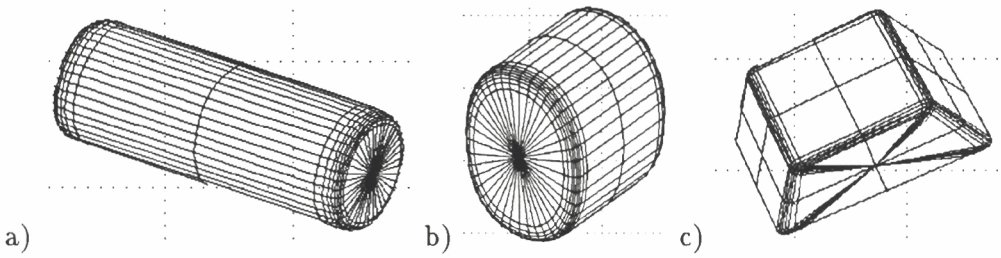
Figure 7: Triangulation examples: a) The Face Mask, and b) the Hand.

revolution modeling primitive is selected. The quality of the fit is low for the Face Mask and the Hand data and they are modeled as a collection of bounding surfaces. In addition, a tapering deformation is obtained for the Hand data. The recovered

Table 2: Superquadric model recovery results

Parameters/ Test image	Shape	Goodness of fit GOF
Cylindrical Pin	$\varepsilon_1 = 0.10, \varepsilon_2 = 1.08$	0.06
Face Mask	$\varepsilon_1 = 0.15, \varepsilon_2 = 0.87$	0.15
Hand	$\varepsilon_1 = 0.10, \varepsilon_2 = 0.10$	0.22

superellipsoid models for the test pieces are depicted in Figure 8.



a) b) c)

Figure 8: The obtained superellipsoid models of the test pieces: a) the Cylindrical Pin, b) the Face mask, and c) the Hand

Sculptured surfaces are approximated using NURBS. Surface characterization is employed to estimate the number of control points needed in the control mesh to be

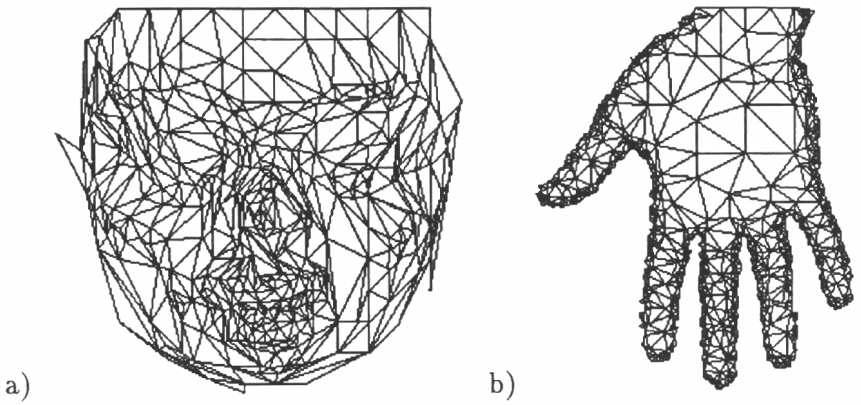


Figure 7: Triangulation examples: a) The Face Mask, and b) the Hand.

revolution modeling primitive is selected. The quality of the fit is low for the Face Mask and the Hand data and they are modeled as a collection of bounding surfaces. In addition, a tapering deformation is obtained for the Hand data. The recovered

Table 2: Superquadric model recovery results

Parameters/ Test image	Shape	Goodness of fit GOF
Cylindrical Pin	$\varepsilon_1 = 0.10, \varepsilon_2 = 1.08$	0.06
Face Mask	$\varepsilon_1 = 0.15, \varepsilon_2 = 0.87$	0.15
Hand	$\varepsilon_1 = 0.10, \varepsilon_2 = 0.10$	0.22

superellipsoid models for the test pieces are depicted in Figure 8.

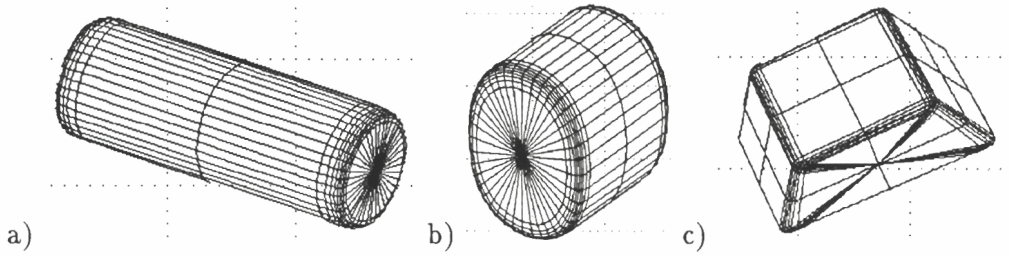


Figure 8: The obtained superellipsoid models of the test pieces: a) the Cylindrical Pin, b) the Face mask, and c) the Hand

Sculptured surfaces are approximated using NURBS. Surface characterization is employed to estimate the number of control points needed in the control mesh to be

able to describe all the degrees of freedom of the underlying surface using B-splines. For each second order surface patch we need 3 control points in each parameter direction. The maximum number of control points in each parameter direction was selected to be the size of the surface mesh in that direction. A patch is considered significant, hence included in the mesh size estimation, if the number of pixels exceeds a given threshold value. Surface characterization results for test images are depicted in Figure 9.

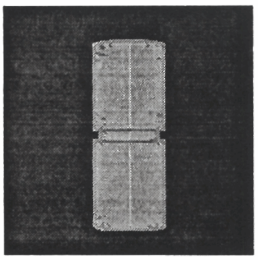
a)  b)  c) 

Figure 9: Surface characterization results: a) the Cylindrical Pin, b) the Face Mask, and c) the Hand.

We chose to use chord length parameterization in spline approximation. In general, the accuracy is good if the surface is smooth. Larger errors are caused by rapid changes in the surface shape. The surface description is refined to meet a user defined tolerance value by inserting knots, and consequently control points. A knot is added to a point where local error maxima exceeding the tolerance value occurs. Figure 10 depicts a profile from the Face Mask and the corresponding error distances to the approximating spline obtained using chord length parameterization. The result of refinement by knot insertion using tolerance value *1.0 mm* are also shown.

Surface discontinuities cause errors because of the continuity property of B-splines are violated. Weaker continuity properties can be introduced by inserting multiple knots at same point, for instance, four in the case of subdividing a cubic B-spline where a discontinuity occurs. An example of subdivision is depicted in Figure 11.

Sometimes rectangular arrangement of data is completely inappropriate. For such situations we chose to use trimmed surfaces. The fit is run on the bounding box of the part and the boundaries are used for computing trimming curves. A fairly dense control point mesh is needed to isolate the errors introduced on the boundaries. Two different parameterization methods were experimented in the context of trimmed

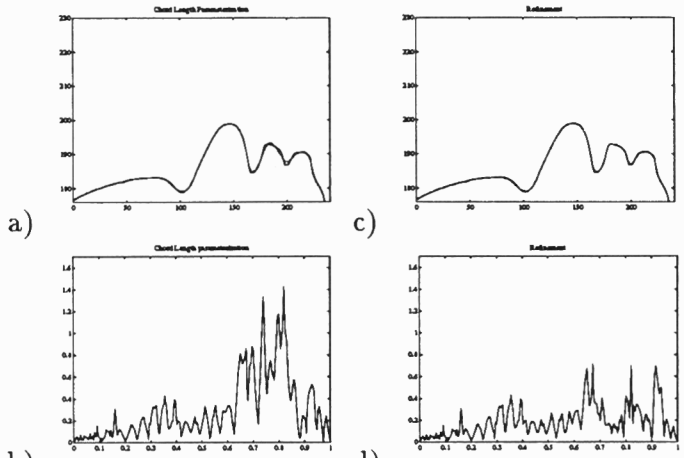


Figure 10: a) The original profile from Face Mask and the obtained B-spline (dashed line), and b) the corresponding error distances. c) shows the B-spline refinement result using 1.0mm tolerance value, and d) the error distances, respectively.

surfaces: Chord length parameterization consumes most of the parameter space by the boundaries, whereas uniform parameterization provides less accurate result by the boundaries but it gives better approximation for the valid parts of the surface. Therefore, use uniform parameterization and insert more knots where large errors occur in order to make the approximation more accurate. An example of surface trimming operation is performed on the Hand image. The boundaries are detected using Deriche edge detector [9]. The complete B-spline surface and the resulting trimmed surface are depicted in Figure 12.

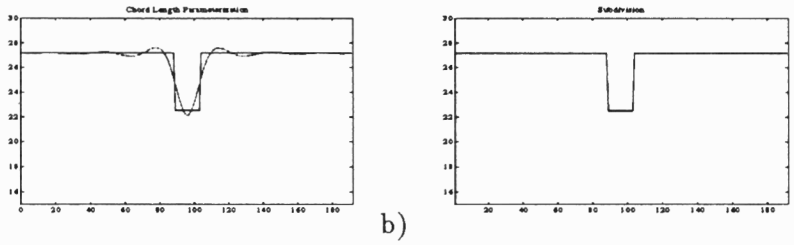


Figure 11: B-spline subdivision allows representing discontinuities: The simulated data from the Cylindrical Pin and the obtained B-spline (dashed line) a) before and b) after the subdivision.

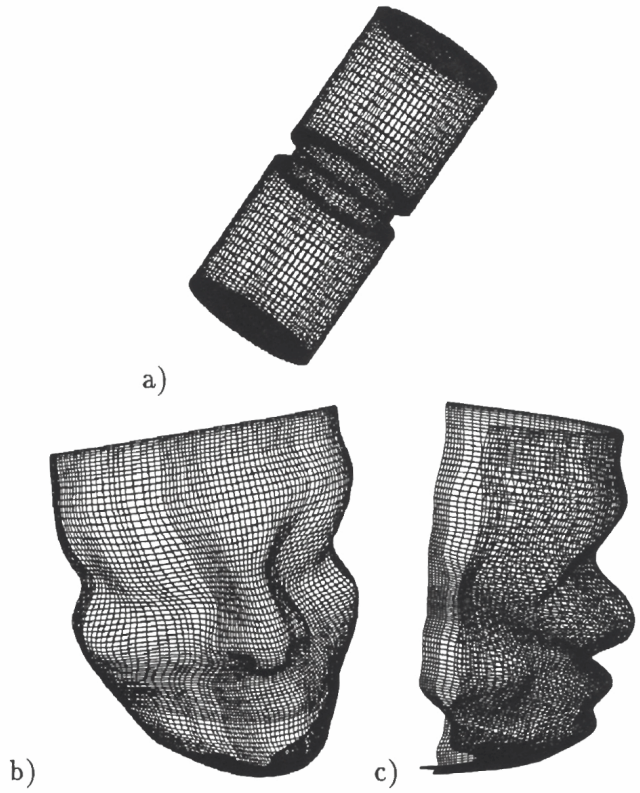


Figure 13: The CAD model for a) the Cylindrical Pin, and b) and c) for the Face Mask, respectively.

a)

```
p12 := projPt(0.000000,80.61196,27.174069,1.0);
p13 := projPt(0.000000,100.16872,27.123658,1.0);
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,sc_open,MyKnotlist),
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);
);
```

b)

```
IGES OUTPUT FROM Rapid Prototyping System                                S0000001
1H,,1H;,,12H symmavy.igs,20HRapid Prototyping    ,5H1.0.0,32,38,6,38,15,,G      1
1.,2,2HMM,32,5.,13H920821.120414,0.00001,1000.0000,2HVK,,9,0;                G      2
     162       1       0       1       0       0       0          0000000ID0000001
     162       0       0       1       0                                  ID0000002
     126       2       0       1       0       0       0                  ID0000003
     126       0       0      32       0                                  ID0000004
162,3,1.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,                                     1P0000001
126,34,3,0,0,0,0,                                                          3P0000002
000.00000,000.00000,000.00000,000.00000,000.03125,000.06250,               3P0000003
000.09375,000.12500,000.15625,000.18750,000.21875,000.25000,               3P0000004
```

Figure 14: Model data for the Cylindrical Pin: a) a part of the Alpha_1 model procedure, and b) a part of the IGES description.

5 Summary

In this paper, we presented a computer-aided engineering tool for building geometric models of parts from 3-D sensor data. Data have to be acquired from several viewpoints and integrated into a complete 3-D data set in common coordinate frame. We showed preliminary results on registration using a simple polyhedron type object. Multiple representations are used in model construction in order to model efficiently different shapes and consequently, employ appropriate CAD modeling primitives. Moreover, there is no single representation that would be always appropriate. In particular, we choose to employ superellipsoids, NURBS and Delaunay triangulation in order to cover standard geometric shapes as well as free form and irregular, complicated shapes. The result is a procedural CAD model which is able to convey structural information about the part. A procedure which is generating the part geometry is relatively easy to modify by the user which is necessary because the design typically evolves. The model building is addressed as a part of concurrent engineering environment. Hence, the model have to be translated to standard product data exchange format to enable data sharing with other processes.

The ongoing and future research is directed toward refining and extending the data acquisition part in order to combine data from different vantage points even if the object has sculptured free form shape. In particular, surface triangulations are employed to register the views incrementally. Methods for modeling surfaces of arbitrary topology are also developed further. Moreover, interfacing to engineering analysis tools, e.g., dynamic simulation, is under work. The motivation is to be able to view a part, accurately model and represent it, and predict the effect of different design parameters on the performance of the mechanical system in order to optimize or redesign the parts without actually building prototypes.

References

[1] Alpha_1 User's Manual (1992) University of Utah. USA.

[2] Barr, A., "Superquadrics and Angle Preserving Transformations", IEEE Computer Graphics and Applications, Vol. 1, pp. 11-23, 1981.

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

[4] Besl, P., McKay, N., "A Method for Registration of 3-D Shapes", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 239-256, 1992.

[5] 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.

[6] Chen, Y., Medioni, G., "Integrating Multiple Range Images Using Triangulation" ARPA Image Understanding Workshop, pp. 951-958, 1993.

[7] Chiyokura, H., Takamura, T., Konno, K., Harada, T., "G^1 Surface Interpolation over Irregular Meshes with Rational Curves", In: Farin G. (ed), NURBS for Curve and Surface Design, SIAM, Philadelphia, pp. 15-34, 1991.

[8] 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.

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

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

[11] Field, D., "A Generic Delaunay Triangulation Algorithm for Finite Element Meshes", Adv. Eng. Software, 1991, Vol. 13, No. 5/6 combined, pp. 263-272, 1991.

[12] Initial Graphics Exchange Standard, Version 5.1., IGES/PDES Organization, 1991.

[13] Koivunen, V. "Robust Signal Restoration and Local Estimation of Image Structure", Technical Report MS-CIS-92-92, University of Pennsylvania. 1992.

[14] Koivunen, V. "Robust Approach to Filtering of Scalar and Vector Valued Signals", IEEE Workshop on Image and Multidimensional Signal Processing, Cannes, France, 1993.

[15] Koivunen, V., Bajcsy, R., "Geometric Methods for Building CAD Models from Range Data", In: Geometric Methods in Computer Vision II, SPIE Vol. 2031, San Diego, 1993.

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

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

[18] Mortenson, M., "Geometric Modeling", John Wiley & Sons. 1985.

[19] Nielson, G., "Scattered Data Modeling", IEEE CG & A, Vol. 13, No. 1, pp. 60-70, 1993.

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

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

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

[23] Requicha, A., Voelcker, H., "Solid Modeling: A Historical Summary and Contemporary Assessment", IEEE Computer Graphics and Applications, Vol. 2, No. 2, pp. 9-24, 1982.

[24] Rioux, M., Cornoyer, L. "The NRCC Three Dimensional Image Data Files". CNRC 29077, National Research Council of Canada, 1988.

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

[26] Shah, J., "Assessment of Features Technology", Computer-Aided Design, Vol. 23, No. 5, pp. 331-343, 1991.

[27] Schumaker, L., "Triangulations in CAGD", IEEE CG & A, Vol. 13, No. 1, pp. 47-52, 1993.

[28] Solina, F., Bajcsy, R., "Recovery of Parametric Models from Range Images: The case of Superquadrics with Global Deformations", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-12, No. 2, pp. 131-147, 1990.

[29] Soucy, M., Laurendeau, D., "Building Surface Model of an Object Using Multiple Range Views", Intelligent Robots and Computer Vision X, SPIE Vol. 1608, pp. 85-94, 1991.

[30] Trucco, E., Fitzgibbon, A., Fisher, R., "Model Acquisition from Multiple Range Views: a Survey", ACME Techical Report D1, Dept. of Artificial Intelligence, Univ. of Edinburgh, U.K., 1993.

[31] Vezien J-M, "Data Acquisition and Registration of Range Images for Reverse Engineering Applications", Technical Report, University of Pennsylvania, Department of Computer Science, July, 1993.