Recovery of Parametric Models from Range Images: The Case for Superquadrics with Global Deformations

FRANC SOLINA AND RUZENA BAJCSY, SENIOR MEMBER, IEEE

Abstract—A method for recovery of compact volumetric models for shape representation of single-part objects in computer vision is introduced. The models are superquadrics with parametric deformations (bending, tapering, and cavity deformation). The input for the model recovery is three-dimensional range points. We define an energy or cost function whose value depends on the distance of points from the model's surface and on the overall size of the model. Model recovery is formulated as a least-squares minimization of the cost function for all range points belonging to a single part. The initial estimate required for minimization is the rough position, orientation, and size of the object. During the iterative gradient descent minimization process, all model parameters are adjusted simultaneously, recovering position, orientation, size, and shape of the model, such that most of the given range points lie close to the model's surface. Because of the ambiguity of superquadric models, the same shape can be described with different sets of parameters. A specific solution among several acceptable solutions, which are all minima in the parameter space, can be reached by constraining the search to a part of the parameter space. The many shallow local minima in the parameter space are avoided as a solution by using a stochastic technique during minimization. Results using real range data show that the recovered models are stable and that the recovery procedure is fast.

Index Terms—Computer vision, least-squares fitting, range image interpretation, shape deformations, shape representation, volumetric models.

I. Introduction

VISUAL perception enables intelligent interaction with the environment. It provides us with information that makes it possible to locate and recognize objects and their interrelationships without direct physical contact. In short, vision makes the outside world accessible to thought [11]. Although we are far from understanding the complexities of human visual perception, we are trying to give machines the sense of vision for the very same reason—to enable them to interact with a changing environment. The type of interactions and the environment where machine vision systems are currently employed are quite restricted and limited in comparison to capabilities of the human visual system. Nevertheless, computer vision and the study of human vision share some common problems on the computational level, for example, *what* information is derived from the world and *how* is it done, independent of specific algorithms and mechanisms.

The main problem in vision research is that two-dimensional images *underdetermine* the three-dimensional world. In theory, an infinite number of 3-D scenes can produce the same 2-D image. *Additional* information is required not only to invert the projection but to derive and infer meaningful descriptions of the world. This additional knowledge is about image formation and the structure of the world. In computer vision systems, this knowledge is incorporated in models. Since a large part of visual information is in geometrical form, most of these models are models for representing shape. The way information in a vision system can be organized and processed is closely linked to the selected representation of shape. We believe that inadequate shape representation is a central problem in computer vision.

The pixel by pixel information in images must be organized into larger entities or models. Models on any level achieve reduction of data by imposing a certain organization on it. This organization is the knowledge or expectation of how data is structured. The higher the level of a model, the more structure it imposes on data. An open question in computer vision research is what are the highest level generic models that can be recovered from unconstrained environments in a purely data-driven fashion—that is, without invoking any semantic or context dependent knowledge. Most so called "model driven" object recognition vision systems use context dependent information by relying on a set of rigid or precise models for all objects expected to be found in the scene [7], [14]. This approach is possible in tightly controlled environments, such as in industrial applications. It clearly cannot work in unconstrained environments due to the sheer number of necessary models, not to mention the problem of building such a model database. A more general way of describing objects and scenes must be used in such cases. Traditional approaches advocate a stepwise reduction of data [20]. First, low level shape models such as

Manuscript received April 25, 1988; revised July 15, 1989. Recommended for acceptance by J. L. Mundy. This work was supported by the following grants and contracts: USPS 104230-87-H-0001/M-0195, ONR Subcontract SB35923-0, NSF/DCR-8410771, ARMY/DAAG-29-84-K-0061, NSF-CER/DCR82-19196 A02, Air Force/F49620-85-K-0018, and DARPA/ONR.

F. Solina was with GRASP Laboratory, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104. He is now with the Faculty of Electrical Engineering and Computer Science, University of Ljubljana, Tržaška 25, 61001 Ljubljana, Yugoslavia.

R. Bajcsy is with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104.

IEEE Log Number 8931852.

0162-8828/90/0200-0131$01.00 © 1990 IEEE

edges, corners and surface patches are computed locally. Because in natural scenes the image formation parameters can change abruptly from image point to image point, these local models are error-prone. Due to the small granularity of these models, a large number of them is required even for a moderately complicated scene. Since any matching with a large set of models to recognize objects on this level of representation leads to a combinatorial explosion, another step of information compression by infusion of outside knowlege is required. This can be done either by retaining the low level models and selecting only the relevant ones by using the laws of perceptual organization [37], or by building compact volumetric models of larger granularity. In either case, if the additional information is not explicit, this is a difficult step to make. The additional knowledge should enable patching up for missing information and rejecting erroneous information in a robust and verifiable manner. We believe that the solution is to develop powerful mathematical models whose parameters can be recovered from the image under the internal constraints of the models. Those internal constraints should reflect the background knowledge of image formation and shape of real-world objects.

In this paper, we introduce a method for recovery of compact volumetric models for single part objects. To solve the shape recovery problem in isolation from segmentation, we assume that only a single object is present in the scene at a time. Although we made this simplification to break up the problem, this assumption is still valid for some restricted environments [30]. We show that the *shape* of those objects can be recovered subject to the model's internal constraints. In this work we use a particular example of compact volumetric models—superquadric primitives with parametric deformations. We introduce a least-squares minimization method to recover model and deformation parameters using range data as the input. Range data enables us to study shape recovery independent of different passive techniques of obtaining depth data, such as depth from stereo, depth from focus, or depth from motion. The fitting function which we minimize is a cost or energy function whose value depends on the distance of points from the model's surface and on the overall size of the model. We show that the solution space, which can have more than one "deep" minimum or acceptable solution and many shallow local minima, can be searched efficiently with a gradient descent method. By using a stochastic technique, the procedure can escape from shallow local minima, and a particular solution among several acceptable solutions can be reached by searching in a constrained parameter subspace.

The paper is organized as follows. Section II is on parametric models in computer vision, focusing on comparison of generalized cylinders and superquadrics. Section III explains superquadric models in detail. Section IV is about recovery of nondeformed superquadric models, and Section V is on recovery of deformed superquadrics. Least-squares minimization of the model fitting function is introduced, and results of this model recovery are shown. Section VI examines several issues concerning stability and speed of superquadric model recovery. The discussion in Section VII compares the advantages and deficiencies of the proposed shape representation, identifies applications for it, and points to possible future extensions.

II. Parametric Models

Parametric models are suitable models for computer vision because we can form *overconstrained* estimates of their parameters. This overconstraint comes from using models defined by a few parameters to describe a large number of image points. To find parameters so that the model best fits the data is called an overdetermined optimization problem. The result of this optimization is also a measure of how well models fit the data. This is an important aspect for computer vision since normally results of interpretation are tedious to verify and verification is hence too often omitted.

Although most low level models used in computer vision are parametric models, they are too local to be able to capture or make use of the gross structure of the world. Compact volumetric models of larger granularity seem to be a better level of description. Generalized cylinders, proposed by Binford, were the first such influential models intended specifically for computer vision. The first attempts to recover generalized cylinders were made with a series of intermediate representations, using rule base reasoning, ACRONYM being the most representative of such systems [10]. Generalized cylinders can also be recovered directly from short edge segments obtained from stereo [26]. Common to all these model recovery methods is that they are based on rules, such as the importance of parallel lines, which are derived from perceptual organization ideas. In such rule-based systems, monitoring of progress is difficult and a direct evaluation criteria of results is not available. Typically, those systems can recover only a restricted subset of generalized cylinders, such as linear straight homogeneous generalized cylinders [26].

The power of parametric models is better used if mathematical methods such as linear regression, variational calculus, or least-squares minimization can be used for recovery of their parameters. The shape and recovery of such models can be explained in terms of intrinsic and extrinsic forces. Intrinsic are the internal properties of the model, governing its possibe arrangements and its potential shape. Extrinsic forces are the influences which direct the shape options allowed by the internal constraints. Shape is a result of the interaction of intrinsic and extrinsic forces.

Terzopoulos, Witkin, and Kass [33] have proposed a 3-D shape model similar to generalized cylinders, but enhanced with deformation parameters to control the elasticity of the main axis and the walls of the cylinder. Using this powerful but complicated model, they are able to recover models from 2-D silhouettes. Recovery requires setting constraints on multiple parameters and this so far requires human interaction. The fitting function for these distributed models is complicated and numerical differentiation is required.

We decided to trade some flexibility in shape description for a simpler and more efficient recovery of models by adopting a shape vocabulary for modeling on the level of parts which consists of superquadrics and global deformations [23]. Description of shapes on the part level does not require complicated models. While a cylinder with a bump on it could be modeled with a single primitive in the modeling system proposed by Terzopoulos, Witkin, and Kass, it would require two primitives or parts in the vocabulary that we use—a cylinder *and* a bump. This level of description is also favored by some psychologists when trying to explain human perception [6]. Superquadric models are implicitly constrained by the class of shapes that they can model and do not require any setting of external constraints from case to case. An important advantage for ease of model recovery is that the function that has to be minimized is differentiable everywhere. This is due to the fact that a superquadric surface is defined with a single analytic function. Superquadric shapes form a subclass of shapes describable by generalized cylinders. Boolean operations can be defined on superquadric primitives and variations of shape can be described with two shape parameters or with global parametric deformations. Superquadrics with parametric deformations encompass a large variety of natural shapes yet are simple enough to be solved for their parameters. This enables uniform handling of a large set of primitives with a small set of perceptually relevant parameters. Due to their built-in symmetry, superquadric models predict the shape of occluded parts by assuming global symmetry. This conforms with the principle of parsimony—among several hypothesis select the simplest—which guides human perception [13].

III. Superquadrics

Superquadrics are a family of parametric shapes that were discovered by the Danish designer Peit Hein [12] as an extension of basic quadric surfaces and solids. Superquadrics have been used or proposed for use as primitives for shape representation in computer graphics [4] and computer vision [23]. Superquadrics play the role of prototypical parts and can be further deformed and glued together into realistic looking models as is nicely demonstrated by Pentland's *Supersketch* graphics system [23].

A superquadric surface is defined by the following 3-D vector

$$\boldsymbol{x}(\eta, \omega) = \begin{bmatrix} a_1 \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\omega) \\ a_2 \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\omega) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix} \qquad \begin{aligned} -\pi/2 &\leq \eta \leq \pi/2 \\ -\pi &\leq \omega < \pi. \end{aligned} \tag{1}$$

The vector $\boldsymbol{x}$ originates in the coordinate center and sweeps out a closed surface in space when the two independent parameters, angles η and ω, change in the given intervals. ω is the angle between the x-axis and the projection of vector $\boldsymbol{x}$ in the x-y plane, while η is the angle between vector $\boldsymbol{x}$ and its projection in the x-y plane. Parameters η and ω correspond to latitude and longitude angles of vector $\boldsymbol{x}$ expressed in spherical coordinates. Parameters a_1, a_2, a_3 define the superquadric size in x, y, and z coordinates, respectively. ϵ_1 is the squareness parameter in the latitude plane and ϵ_2 is the squareness parameter in the longitudine plane. In writing (1) we followed the established notation [4], [23]. But since function values of cos and sin are negative for same values of both η and ω, the first component of vector $\boldsymbol{x}$ should be written as sign $(\cos(\eta)\cos(\omega))\,|\cos(\eta)|^{\epsilon_1}\,|\cos(\omega)|^{\epsilon_2}$. This prevents components of vector $\boldsymbol{x}$ to have complex values, which are in general a result of a negative number raised to a real power. The same holds for the other two components of vector $\boldsymbol{x}$.

Superquadrics can model a large set of standard building blocks, like spheres, cylinders, parallelepipeds and shapes in between (Fig. 1). When both ϵ_1 and ϵ_2 are 1, the surface vector defines an ellipsoid or, if a_1, a_2, a_3 are all equal, a sphere. When ϵ_1 is $\ll 1$ and $\epsilon_2 = 1$, the superquadric surface is shaped like a cylinder. Parallelepipeds are produced when both ϵ_1 and ϵ_2 are $\ll 1$. Flat beveled shapes are produced when either ϵ_1 or $\epsilon_2 = 2$, and pinched shapes are produced when either ϵ_1 or $\epsilon_2 > 2$. We do not use pinched shapes in our model vocabulary because we prefer to model concavities with deformations. Star-like shapes $(\epsilon_1, \epsilon_2 \gg 2)$ perceptually consist of parts and should be modeled as a union of parts.

Modeling capabilities of superquadrics can be enhanced by deforming them in different ways, including tapering, bending and making cavities (Fig. 1). Definitions for tapering, bending and cavity deformations and a method for their recovery are given in Section V.

A. Superquadric Inside-Outside Function

Equation (1) is a parametric equation of a superquadric surface. By eliminating parameters η and ω, using equality $\cos^2(\alpha) + \cos^2(\alpha) = 1$, we get the following implicit equation

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

Based on this implicit equation of the superquadric surface we define the following function

$$F(x, y, z) = \left(\left(\left(\frac{x}{a_1}\right)^{2/\epsilon_2} + \left(\frac{y}{a_2}\right)^{2/\epsilon_2}\right)^{\epsilon_2/\epsilon_1} + \left(\frac{z}{a_3}\right)^{2/\epsilon_1}\right)^{\epsilon_1}. \tag{3}$$

We refer to this function as the inside-outside function because it determines where a given point $[x, y, z]^T$ lies relative to the superquadric surface. If $F(x, y, z) = 1$, point (x, y, z) is on the surface of the superquadric. If

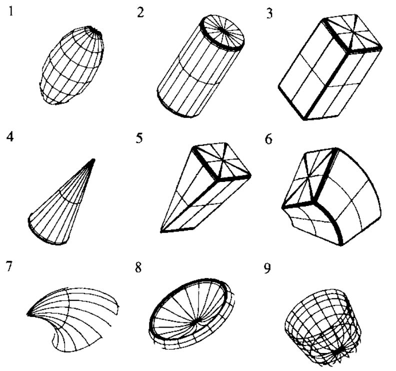

Fig. 1. Superquadrics (1, 2, 3) and deformed superquadrics (models 4, and 5 are tapered, model 6 is bent, model 7 is tapered and bent, models 8 and 9 are a result of a cavity deformation).

$F(x, y, z) > 1$, the corresponding point lies outside and if $F(x, y, z) < 1$, the corresponding point lies inside the superquadric.

Note the outermost exponent ϵ_1 in (3). This additional exponent ϵ_1 does not change the superquadric surface itself but is *necessary* if the function is used for shape recovery with a least squares minimization method. It ensures that, independent of the current value of ϵ_1, points at the same distance from the superquadric surface have the same value of $F(\boldsymbol{x})$. Otherwise, when $\epsilon_1 \ll 1$, even very small deviations of a point from the superquadric surface in the z coordinate are greatly amplified. For example, minimizing the inside–outside function without this correction in the case of a cylindrically shaped object, where $\epsilon_1 = 0.1$ and $\epsilon_2 = 1$, does not give consistent solutions. Instead of finding the expected cylindrical shape, a gradient minimization algorithm might converge towards a very long, flat beveled shape ($\epsilon_1 = 2$, $\epsilon_2 = 1$, $a_3 \gg a_1, a_2$), even when the initial estimates are close to the desired solution. The same problem is described elsewhere although no explanation for the cause of the problem is given [8].

B. Superquadric Inside–Outside Function for General Position and Orientation

The inside–outside function (3) defines the superquadric surface in an object centered coordinate system (x_S, y_S, z_S). Input 3-D points from passive stereo or range imaging, on the other hand, are expressed in a world coordinate system. We move points to the center of the world coordinate system with a homogeneous coordinate transformation $\boldsymbol{T}^{-1}$ (Fig. 2)

$$\begin{bmatrix} x_S \\ y_S \\ z_S \\ 1 \end{bmatrix} = \boldsymbol{T}^{-1} \begin{bmatrix} x_W \\ y_W \\ z_W \\ 1 \end{bmatrix}. \qquad (4)$$

Transformation $\boldsymbol{T}^{-1}$ is the inverse of transformation matrix $\boldsymbol{T}$, which first rotates a point and then translates it from the origin of the world coordinate system for $[p_x, p_y, p_z, 1]^T$. By inverting the homogeneous transformation matrix $\boldsymbol{T}$ and using Euler angles (ϕ, θ, ψ) to express the elements of the rotational part of transformation matrix $\boldsymbol{T}$ [22] we get the inside–outside function for superquadrics in general position

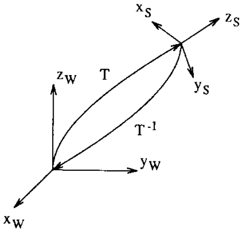

Fig. 2. Coordinate transforms $\boldsymbol{T}$ and $\boldsymbol{T}^{-1}$ link the object centered coordinate system of the superquadric with the world coordinate system of the ranging device.

$$F(x_W, y_W, z_W) = F(x_W, y_W, z_W; a_1, a_2, a_3, \epsilon_1, \epsilon_2, \phi, \theta, \psi, p_x, p_y, p_z). \qquad (5)$$

The expanded inside–outside function has 11 parameters; a_1, a_2, a_3 define the superquadric size; ϵ_1 and ϵ_2 are for shape; ϕ, θ, ψ for orientation; and p_x, p_y, p_z for position in space. We will refer to the set of all model parameter values as $\Lambda = \{a_1, a_2, \cdots, a_{11}\}$.

C. Ambiguity of Superquadric Shape Description

Different sets of superquadric parameters can produce identical shapes. Besides the common symmetries with respect to rotation, superquadrics can also describe the same shape with different sets of shape parameters ϵ_1 and ϵ_2. For example, the same parallelepiped with slightly rounded edges ($a_1 = a_2 = 1$, $\epsilon_1 = 0.1$ and $\epsilon_2 = 0.1$) can be represented after a rotation for $\pi/4$ around axis z of the object centered coordinate system, with $\epsilon_1 = 0.1$, $\epsilon_2 = 1.9$, and $a_1 = a_2 = \sqrt{2}$.

If the recovered models are used only to represent space occupancy (to do path or grasp planning, for example) such ambiguities are not a problem. Nonunique shape description is a problem when the recovered shape description is used for matching with a database of models. There are two ways that one can solve the problem. One can accept this lack of uniqueness and find the correspondent equivalence class after shape recovery. The other possibility is to constrain the parameter space during shape recovery to force the system to find particular solutions. An example of such constrained shape recovery is given in Section VI-C.

IV. Recovery of Nondeformed Superquadric Models

In this section, recovery of nondeformed superquadrics in general position is described. The recovery method is based on least squares minimization of a fitting function which is based on the inside–outside superquadric function (3).

Other approaches to recovery of superquadrics for shape description described in the literature are due to Pentland [23], [24] and Boult and Gross [8]. Pentland [23] first suggested to solve analytically for all independent parameters using the superquadric parametric equation (1) and the corresponding parametric equation for superquadric surface normals. Linear regression on all available information would provide the best fit. As input Pentland suggested information from 2-D contours and shading. An analytical solution turned out to be too complicated and this approach was never implemented for objects in general position.

Pentland's second approach [24] is a combination of part model recovery and segmentation based on search through the entire superquadric parameter space. Input are 3-D points from range images. For each range image point or pixel, the superquadric parameter space is searched to provide the best fit for the immediately surrounding area. The goodness of fit is evaluated with a linear combination of the data error using a minimal-length encoding and the number of parameters in the description. Such a brute force is computationally expensive and practical only on powerful parallel computers. We will show in the rest of this section that a faster search in the superquadric parameter space is possible by minimizing a fitting function when combined with a stochastic technique to avoid shallow local minima.

Boult and Gross [8] also use a gradient descent minimization technique. Since they do not correct the inside-outside function for fitting (see Section III-A), they report problems with convergence of cylindrically shaped objects.

The next two sections are on least-squares minimization (Section IV-A) and on computing rough position, orientation and size of the model (Section IV-B), which is necessary to initialize minimization. Throughout this work we used range images obtained with a laser imager built at University of Pennsylvania [34]. Before model recovery, the supporting surface was removed by fitting a plane to it and subtracting the points on or close to that plane.

A. Least-Squares Minimization of Superquadric Inside-Outside Function

Suppose we have N 3-D surface points $(x_{W_i}, y_{W_i}, z_{W_i})$, $i = 1, \cdots, N$ which we want to model with a superquadric. We want to vary the 11 parameters $a_j, j = 1, \cdots, 11$ in (5) to get such values for a_j's that most of the 3-D points will lay on, or close to the model's surface. There will probably not exist a set of parameters Λ that perfectly fits the data. Finding the model Λ for which the distance from points to the model is minimal is a least-squares minimization problem. Since, for a point $[x_W, y_W, z_W]^T$ on the surface of a superquadric $F(x_W, y_W, z_W; a_1, \cdots, a_{11}) = 1$, we have to find

$$\min \sum_{i=1}^{N} \left[1 - F(x_{W_i}, y_{W_i}, z_{W_i}; a_1, \cdots, a_{11})\right]^2. \quad (6)$$

Due to self occlusion, not all sides of an object are visible at the same time. We assume a general view for recovery of object shapes. Seeing, for example, just one side of a cube does not provide enough information on the extent of the whole object [18], [17]. But even assuming a general viewpoint, objects such as parallelepipeds or cylinders (objects with surfaces where at least one principal curvature = 0) do not provide enough constraints for shape recovery with the inside-outside function alone. Parallelepipeds of different size, for example, satisfy (6) given range points on three or two adjacent faces. Among all those solutions we want to find the smallest superquadric that fits the given range points in the least squares sense. We have to find a function with a minimum corresponding to the smallest superquadric that fits a set of 3-D points *and* such that the function value for surface points is known before minimization. We define a new fitting function

$$R = \sqrt{a_1 a_2 a_3}\,(F - 1), \quad (7)$$

which fulfills the first requirement with factor $(a_1 a_2 a_3)^{1/2}$ and the second requirement with factor $(F - 1)$, because function $R = 0$ for all points on the superquadric surface. In Section VI-C we explain how the factor $(a_1 a_2 a_3)^{1/2}$, which corresponds to the superquadric size, was chosen.

Now, we have to minimize the following expression

$$\min \sum_{i=1}^{N} \left[R(x_{W_1}, y_{W_1}, z_{W_1}; a_1, \cdots, a_{11})\right]^2. \quad (8)$$

Since R is a nonlinear function of 11 parameters $a_j, j = 1, \cdots, 11$, minimization must proceed iteratively. Given a trial set of values of model parameters Λ_k, we evaluate (7) for all N points and employ a procedure to improve the trial solution. The procedure is then repeated with a set of new trial values Λ_{k+1} until the sum of least-squares (8) stops decreasing, or the changes are statistically meaningless.

We had to introduce inequality constraints on the set of function parameters Λ. Constraints that we use are simple bounds on parameter values in the form of intervals. A constraint becomes active when a parameter reaches the lower or upper bound of the allowable interval. Observing that the constraints became active mostly for those sets of parameters Λ that were not accepted because the step in the derivative direction was too large, we concluded that constraints are necessary not so much to assure convergence to a local minimum but to achieve numerical stability. For example, when any of parameters a_1, a_2, a_3, ϵ_1, $\epsilon_2 = 0$, the inside-outside function (3) becomes singular.

We use the Levenberg-Marquardt method for nonlinear least squares minimization [25], [29] since first derivatives $\delta R / \delta a_i$ for $i = 1, \cdots, 11$ can be computed analytically. Parameter constraints are implemented by a simple projection method [29]. We take the search vector or the trial set of parameters Λ, generated by the unconstrained minimization technique, and project it so that it lies in the intersection of the set of constraint intervals. We use the following constraints: $a_1, a_2, a_3 > 0$ and $0.1 < \{\epsilon_1, \epsilon_2\} < 2$. When $\epsilon_1, \epsilon_2 < 0.1$, the inside–outside function (3) might become numerically unstable, although the superquadric shape stays perceptually the same. When $\epsilon_1, \epsilon_2 > 2$, superquadrics have concavities and, as explained in Section III, we do not use those models in our shape vocabulary. We explain in the next section how the first trial set of parameters Λ_E is estimated experimentally.

B. Computation of Initial Estimates for Model Parameters

Initial estimate of the set of model parameters Λ_E determines to which local minimum the minimization procedure will converge. When testing the iterative model recovery method described in the previous section, we found that only very rough estimates of object's true position, orientation, and size suffice to assure convergence to a local minimum that corresponds to the actual shape. This is important since these parameters can be estimated only from the range points on the visible side of the object and hence the estimates cannot be very accurate to begin with. Initial values for both shape parameters, ϵ_1 and ϵ_2 can always be 1, which means that the initial model Λ_E is always an ellipsoid. This insensitivity towards correct estimation of the two shape parameters was achieved when the outermost exponent ϵ_1 was added to the inside–outside function (3). The position of the initial ellipsoid Λ_E is set to the center of gravity of all N range points:

$$p_{xE} = \bar{x},$$

$$p_{yE} = \bar{y},$$

$$p_{zE} = \bar{z}.$$

To compute the orientation of the object centered coordinate system, we compute first the matrix of central moments

$$M = \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} (y_i - \bar{y})^2 + (z_i - \bar{z})^2 & -(y_i - \bar{y})(x_i - \bar{x}) & -(z_i - \bar{z})(x_i - \bar{x}) \\ -(x_i - \bar{x})(y_i - \bar{y}) & (x_i - \bar{x})^2 + (z_i - \bar{z})^2 & -(z_i - \bar{z})(y_i - \bar{y}) \\ -(x_i - \bar{x})(z_i - \bar{z}) & -(y_i - \bar{y})(z_i - \bar{z}) & (x_i - \bar{x})^2 + (y_i - \bar{y})^2 \end{bmatrix}. \tag{9}$$

Central moments are moments with respect to the center of gravity $(\bar{x}, \bar{y}, \bar{z})$. We want to find a rotation matrix $\boldsymbol{R}$ which makes the matrix of moments $\boldsymbol{M}$ diagonal [28], [16]. The new diagonal matrix of moments $\boldsymbol{D}$ is then

$$\boldsymbol{D} = \boldsymbol{R}^{-1}\boldsymbol{M}\boldsymbol{R}, \tag{10}$$

where $\boldsymbol{R}$ is the rotational part of transform $\boldsymbol{T}$ in Section III-B. On the other hand, matrix $\boldsymbol{M}$ can be diagonalized with a diagonalization matrix $\boldsymbol{Q}$, whose columns are eigenvectors of matrix $\boldsymbol{M}$ [31]

$$\boldsymbol{D} = \boldsymbol{Q}^{-1}\boldsymbol{M}\boldsymbol{Q}. \tag{11}$$

Comparing (10) and (11) gives

$$\boldsymbol{R} = \boldsymbol{Q}. \tag{12}$$

Rotation matrix $\boldsymbol{R}$ that orients the object centered coordinate system along the axis of minimum and maximum inertia can be assembled out of eigenvectors of matrix $\boldsymbol{M}$. Eigenvector $\boldsymbol{e}_1$ with the smallest eigenvalue λ_1 corresponds to the minimum-inertia line and the eigenvector $\boldsymbol{e}_3$ with the largest eigenvalue to the maximum inertia line. The minimum-inertia line is also known as the principal axis [28]. We use the Jacobi method for computing the matrix of eigenvectors which consists of a sequence of orthogonal similarity transformations designed to annihilate one of the off-diagonal matrix elements. The Jacobi method is fast for matrices of order less than 10 and absolutely foolproof for all real symmetric matrices [25] which is the case with matrix $\boldsymbol{M}$.

We decided to orient the object centered coordinate system so that the new axis z lies along the longest side for elongated objects (axis of least inertia) and along the shortest for flat objects (axis of largest inertia), based on the assumption that bending and tapering deformations normally affect objects along their longest side. For round flat objects, on the other hand, we want the z coordinate axis to coincide with the axis of the rotational symmetry (Fig. 8). Given the three eigenvectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$, we have to assign them coordinate axes labels. As explained, we want to control only the orientation of the z axis. For ordered eigenvalues $\lambda_1 < \lambda_2 < \lambda_3$, of the three corresponding eigenvectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$, the z-axis is assigned according to the following rule

$$\textbf{\textit{if}}\ |\lambda_1 - \lambda_2| < |\lambda_2 - \lambda_3|\ \textbf{then}\ z = \boldsymbol{e}_3$$

$$\textbf{else}\ z = \boldsymbol{e}_1.$$

This puts the axis z along the long side of elongated objects and perpendicular to flat, rotationally symmetric objects. From the elements of the rotation matrix $\boldsymbol{R}$ [which makes up the rotational part of transform $\boldsymbol{T}$—(4)] we compute the equivalent Euler angles ϕ_E, θ_E, ψ_E [22].

For evaluating the inside-outside function (5) we could use the elements of the rotational matrix $\boldsymbol{R}$ directly, but the partial derivatives required for minimization of the fitting function are all expressed in terms of Euler angles. The size of the initial ellipsoid Λ_E is simply the distance between the outermost range points along each coordinate axis of the new object centered coordinate system.

Rotations expressed with homogeneous transforms become degenerate for some combinations of angles [22].

During experimentation with shape recovery we did not have problems with degenerate rotations. If necessary, the problem can be avoided by using quaternions for expressing rotations.

The initial estimates computed in the described fashion are sometimes very close to the actual parameter values. In the example shown in Fig. 3, however, the estimated orientation is quite different from the actual orientation. This is due to occlusion and unequal distribution of range points on the object's surface, all made worse because the object is not elongated. The fitting procedure recovers the right model in spite of very poor estimates for orientation. This would suggest that we do not have to estimate orientation at all. The initial ellipsoid could have a default orientation, the same as the world coordinate system for example. This would be acceptable for blob-like objects. But for elongated objects, that might require additional deformations for shape description, because we would lose control over the orientation of the model's z-axis.

V. Recovery of Parametric Deformations

Deformed superquadrics can be recovered using the same technique as for the recovery of nondeformed superquadrics. The only difference is that some additional parameters describing deformations must be recovered also. In general, any shape deformation can be recovered using the technique described in Section IV, as long as the inverse transformation is available such that x, y, z components of the nondeformed superquadric can be expressed in terms of X, Y, Z components of the deformed superquadric and the necessary deformation parameters. Of interest are deformations that occur often in nature or are used for manufacturing of man-made objects. Deformations such as simplified tapering, bending and twisting require just a few parameters [5].

A shape deformation is a function D which explicitly modifies the global coordinates of points in space

$$\boldsymbol{X} = D(\boldsymbol{x}) \tag{13}$$

where $\boldsymbol{x}$ are the points of the undeformed solid and $\boldsymbol{X}$ are the corresponding points after deformation. Both $\boldsymbol{x}$ and $\boldsymbol{X}$ are expressed in the object centered coordinate system. Any translation or rotation is performed after the deformation. This can be described schematically [38] by

$$\text{Trans}\left(\text{Rot}\left(\text{Deform}\left(\boldsymbol{x}\right)\right)\right). \tag{14}$$

Tangent and normal vectors at every point on the deformed surface, which are important for rendering or for checking the consistency of the model with the input range points, can be computed from the tangent and normal vector of the undeformed model simply by a matrix multiplication. Tangent vectors transform under deformations by multiplication with the deformation function's Jacobian and normal vectors transform by multiplication with the inverse transpose of the same Jacobian matrix [5].

We started by using Barr's definitions for tapering and bending [5]. At the very beginning of this work, we considered the use of symmetry axis and Gaussian spheres as indicators of deformations. Now we believe that it would be difficult to estimate deformations accurately and reliably enough to be really useful. We follow instead the principle that deformation parameters should be recovered directly from the input data through shape model recovery. In the initial set of parameters Λ_E, all deformation parameters correspond to nondeformed primitives by default. This called for several changes in definitions of deformations which were originally provided by Barr [5]. Barr defined deformations for computer graphics applications where the designer has the freedom to place and orient the primitives to his liking. During shape recovery such interaction is not possible. Consider, for example, that bending is defined in a single plane as in [5]. Once the initial model is set into the world coordinate system, the whole model cannot rotate freely during the recovery procedure to bring the plane where bending is defined into the position where bending is required. Deformations should be defined so that the object centered coordinate system constrains as little as possible the execution of deformations. In other words, the solution space around acceptable solutions should be convex. We introduced an additional parameter for the bending deformation, which allows for bending in any plane that goes through the z-axis of the object centered coordinate system. Tapering deformation was enhanced to enable different tapering in x and y directions so that modeling of wedge shapes is possible (Fig. 4).

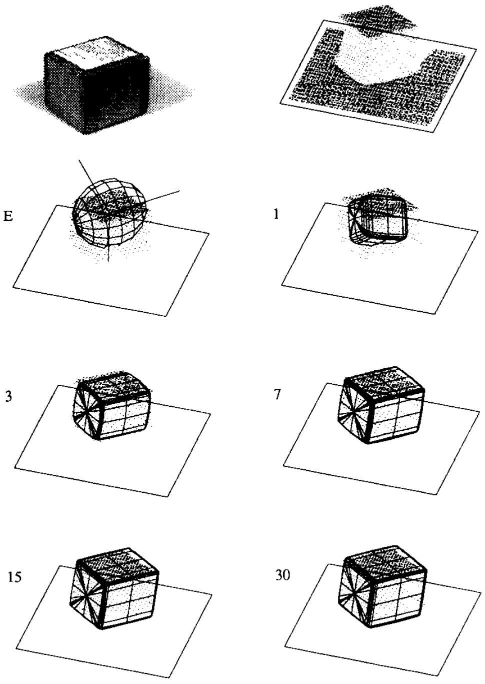

Fig. 3. Shape recovery of a parallelepiped-like object. On top are the intensity image and the corresponding range image. Below is the recovery sequence showing the initial model estimate (E) and models after the 1st, 3rd, 7th, 15th and 30th iteration.

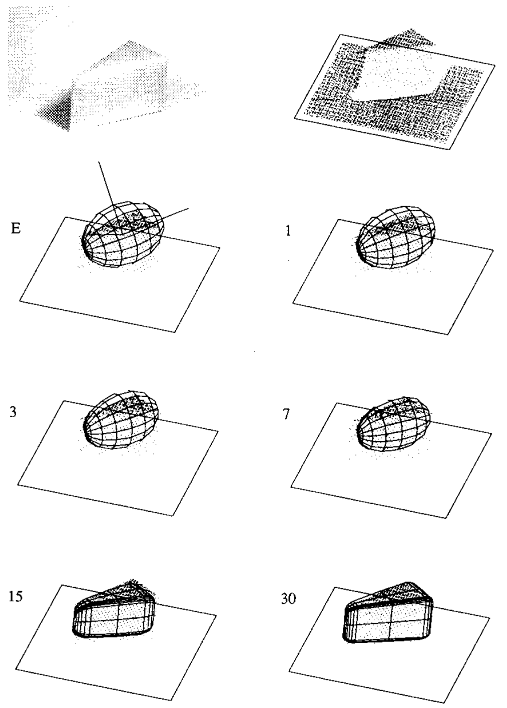

Fig. 4. Shape recovery of a tapered object—a wedge. On top are the intensity image and the corresponding range image. Below is the recovery sequence showing the initial estimate (E) and models after the 1st, 3rd, 7th, 15th and 30th iteration when 13 model parameters were adjusted simultaneously.

We introduced also a new deformation for modeling cavities. The prevailing way of modeling concave parts is to use set difference or intersection. This is probably the best way for modeling holes small in comparison to the rest of the object. In man-made objects, such holes are actually made by removing material. Larger cavities, like various kinds of vessels (plates, bowls, cups), are normally not made by taking material away, but by deforming material into the desired shape. A process resembling a "potter's wheel" technique of making cavities is behind our definition of a cavity deformation. Using such deformation one can model a whole continuum of shapes ranging from flat plates to bowls (Fig. 8).

In the rest of this section, tapering, bending, and cavity deformations are defined and their inverse deformations derived. For each deformation an example of model recovery is shown.

A. Tapering

Tapering deformation along axis z is

$$\begin{aligned} X &= f_x(z)\,x \\ Y &= f_y(z)\,y \\ Z &= z, \end{aligned} \tag{15}$$

where X, Y, Z are the components of the surface vector $\boldsymbol{X}$ of the deformed superquadric, f_x and f_y are the tapering functions in the x- and y-axes of the object centered coordinate system, and x, y, z are the components of the original surface vector $\boldsymbol{x}$.

To be able to recover the deformation parameters, the original surface vector components x, y, z must be expressed in terms of the deformation parameters and coordinates of input points X, Y, Z. The inverse transformation is given by

$$\begin{aligned} x &= X/f_x(z) \\ y &= Y/f_y(z) \\ z &= Z. \end{aligned} \tag{16}$$

For linear tapering, the two tapering functions are

$$f_x(Z_S) = \frac{K_x}{a_3} Z_S + 1 \tag{17}$$

$$f_y(Z_S) = \frac{K_y}{a_3} Z_S + 1, \tag{18}$$

where $-1 \leq K \leq 1$. When expressions for x, y, z are inserted into (5), we get the inside–outside function for a tapered superquadric in general position

$$F(X, Y, Z) = F(X, Y, Z; a_1, \cdots, a_{11}, K_x, K_y). \tag{19}$$

In the set of initial parameters Λ_E, parameters $K_{xE} = K_{yE} = 0$, which corresponds to a nondeformed model. During model recovery both tapering parameters are adjusted simultaneously with the other 11 parameters (Fig. 4).

B. Bending

We use a simple bending deformation which corresponds to a circular section in the bending plane. The bending plane is defined by coordinate axis z and vector $\boldsymbol{r}$ in the x-y plane whose direction is defined by angle α (Fig. 5). The bending deformation is performed first by projecting the x and y components of all points onto the bending plane, performing the bending deformation in that plane, and then projecting the points back to the original plane. The projection of a point (x, y) on the bending plane is

$$r = \cos(\alpha - \beta)\sqrt{x^2 + y^2}, \tag{20}$$

and

$$\beta = \arctan \frac{y}{x}. \tag{21}$$

Bending transforms r into

$$R = k^{-1} - \cos(\gamma)(k^{-1} - r), \tag{22}$$

where γ is the bending angle, computed from the curvature parameter k

$$\gamma = zk^{-1}. \tag{23}$$

By projecting R back onto the original plane, which is parallel to the bending plane, we get the transformed sur-

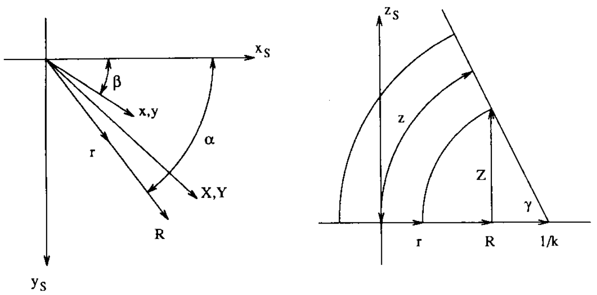

Fig. 5. Definitions for bending parameters: left is the x-y plane, right is the bending plane z-r.

face vector

$$X = x + \cos(\alpha)(R - r),$$
$$Y = y + \sin(\alpha)(R - r),$$
$$Z = \sin(\gamma)(k^{-1} - r). \qquad (24)$$

The inverse transformation is given by

$$x = X - \cos(\alpha)(R - r),$$
$$y = Y - \sin(\alpha)(R - r),$$
$$z = k^{-1}\gamma, \qquad (25)$$

where

$$\gamma = \arctan \frac{Z}{k^{-1} - R}, \qquad (26)$$

$$r = k^{-1} - \sqrt{Z^2 + (k^{-1} - R)^2}, \qquad (27)$$

$$R = \cos\left(\alpha - \arctan \frac{Y}{X}\right) \sqrt{X^2 + Y^2}. \qquad (28)$$

Inserting (25) into (5), we get the inside–outside function for bent superquadrics in general position, which has 13 parameters

$$F(X, Y, Z) = F(X, Y, Z; a_1, \cdots, a_{11}, k, \alpha). \qquad (29)$$

The default values for bending parameters in the initial set of model parameters Λ_E correspond to an unbent superquadric. Since, for $k = 0$, the above equations become singular, a very small number is used for k_E instead. $\alpha_E = 0$, although any real value could be used.

C. Combinations of Deformations

Deformations can be combined in the form of hierarchical structures [5]. A tapered model can be bent by taking the tapered surface position vector and inserting it into the bending equations. The new surface normal vector transformation matrix is computed by multiplying the previous two normal transformation matrices. Since matrix multiplication is not commutative, it is not surprising that deformations are not commutative

$$\text{Bend}\left(\text{Taper}\left(\boldsymbol{x}\right)\right) \neq \text{Taper}\left(\text{Bend}\left(\boldsymbol{x}\right)\right). \qquad (30)$$

During model recovery, we have to settle for a specific ordering of deformations. Tapering usually precedes bending, or in other words, tapering is perceived as to affect the model along the longest axis of symmetry for elongated objects, whether the axis is bent or not. Leyton [19] has shown that deformations acting on prototypical shapes have a specific order, such that the transformation is more structure-preserving. The corresponding model has the following structure

$$\text{Trans}\left(\text{Rot}\left(\text{Bend}\left(\text{Taper}\left(\boldsymbol{x}\right)\right)\right)\right). \qquad (31)$$

Model recovery of a tapered and bent object is shown in Fig. 6.

C. Cavity Deformation

Cavities are made by starting with a thin circular plate and bending its sides uniformly in all directions symmetrically around axis z. Models for objects with such cavities cannot be recovered with the same general fitting function as bending and tapering deformations can. The reason is that a cylindrical model, for example, fits the body of a cup with thin walls as well as a model with an actual cavity does. The flaw of a the first model can be detected though. If the range image is taken from a single viewpoint, then the dot product of the viewing direction vector with the model surface vector through any input range should have the same sign. In the above example, the normals on the cylinder through the range points on the inside of the cavity have the wrong sign. In this way, a cavity can be inferred. To recover cavities, we could use the surface normal information during model recovery as we suggested before [2]. We decided to do it in two steps instead (Fig. 8). First we use the fitting function (7) to recover a convex model enclosing the object. If the surface normals through the input range points are not consistent with the first model, another recovery is done using a fitting function tuned to cavities (Fig. 8). Cavities typically have very thin walls in comparison to other dimensions. Since thickness corresponds to parameter a_3, we want to emphasize models with small a_3. We use the following fitting function for recovery of cavities

$$R_C(X, Y, Z) = a_3^2(F - 1). \qquad (32)$$

The second recovery can start either from the initial model Λ_E or closer to the solution by using the recovered convex model Λ.

The cavity deformation is defined as follows (Fig. 7). First, Cartesian coordinates x and y of a point on the surface of the nondeformed superquadric are transformed into polar coordinates

$$r = \sqrt{x^2 + y^2}, \qquad (33)$$

$$\alpha = \arctan \frac{y}{x}. \qquad (34)$$

r is then bent as defined by the radius of curvature k^{-1}. The range of the bend is controlled by parameters r_0 and

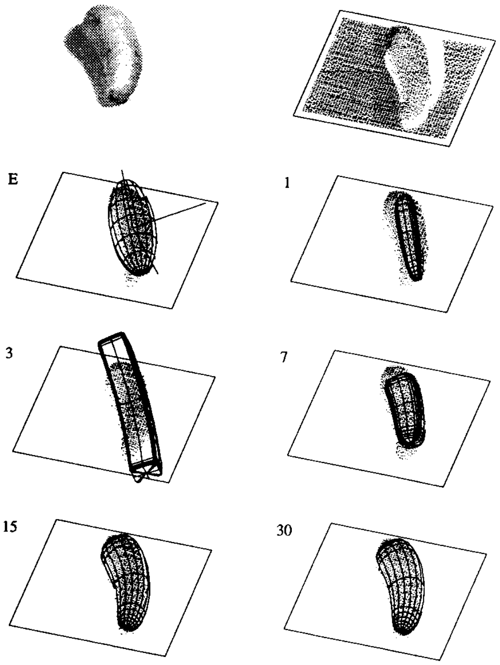

Fig. 6. Shape recovery of a tapered and bent object—a squash. On top are the intensity image and the corresponding range image. Below is the model recovery sequence when a total of 15 model parameters, including tapering and bending parameters, were adjusted simultaneously. The above sequence took about 20 seconds on a VAX 785 computer.

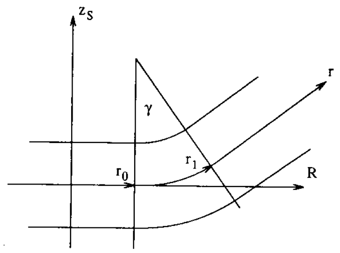

Fig. 7. Definitions of cavity deformation parameters. z_S is the axis of rotational symmetry. r_0 and r_1 define the bent region.

r_1

$$\hat{r} = \begin{cases} r_0 & \text{if } r < r_0 \\ r & \text{if } r_0 < r < r_1 \\ r_1 & \text{if } r_1 < r. \end{cases} \quad (35)$$

The bending angle is given by

$$\gamma = k(\hat{r} - r_0). \quad (36)$$

After bending the polar coordinate r becomes

$$R = \begin{cases} r & \text{if } r < r_0 \\ r_0 + \sin(\gamma)(k^{-1} - z) & \\ \quad \text{if } r_0 < r < r_1 & \\ r_0 + \sin(\gamma)(k^{-1} - z) + \cos(\gamma)(r - r_1) & \\ \quad \text{if } r_1 < r. & \end{cases} \quad (37)$$

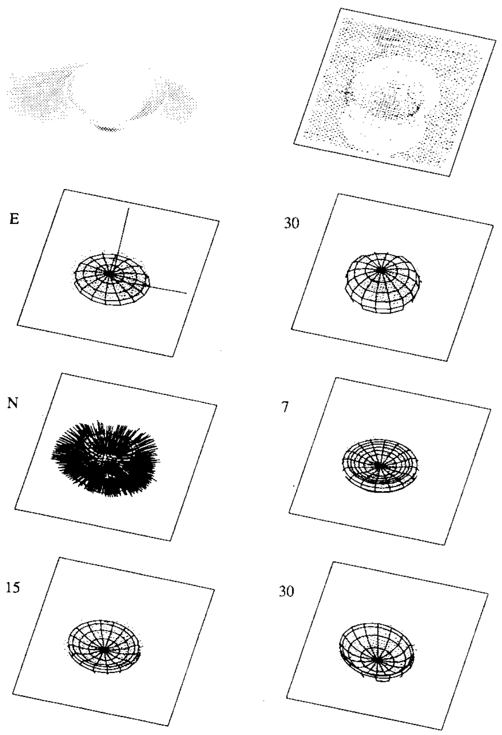

Fig. 8. Shape recovery of an object with a cavity—a bowl. On top are the intensity image and the corresponding range image. Below is the initial estimate (E) and the best fitting convex model after 30 iterations—an ellipsoid in this case. In model N are displayed those surface normals on the recovered model that have the wrong direction, indicating that the points through which those normals go could not be seen from the given viewpoint. Next is the recovery sequence (7-15-30) using a fitting function tuned to cavities (32).

When converting back to Cartesian coordinates, the surface position vector X of the deformed superquadric is

$$X = R\cos(\alpha)$$
$$Y = R\sin(\alpha)$$

$$Z = \begin{cases} z & \text{if } r < r_0 \\ k^{-1} - \cos(\gamma)(k^{-1} - z) & \\ \quad \text{if } r_0 < r < r_1 & \\ k^{-1} - \cos(\gamma)(k^{-1} - z) + \sin(\gamma)(r - r_1) & \\ \quad \text{if } r_1 < r. & \end{cases} \quad (38)$$

The inverse transformation is given by

$$R = \sqrt{X^2 + Y^2}. \quad (39)$$

If $R < r_0$, then

$$x = X,$$
$$y = Y,$$
$$z = Z. \quad (40)$$

If $r_0 < R$, then compute first

$$\alpha = \arctan \frac{Y}{X}, \tag{41}$$

$$\gamma = \arctan \frac{R - r_0}{k^{-1} - Z}, \tag{42}$$

$$r = r_0 + \frac{\gamma}{k}. \tag{43}$$

If $r < r_1$, then

$$x = r \sin(\alpha)$$

$$y = r \cos(\alpha)$$

$$z = \frac{Z - k^{-1}}{\cos(\gamma)} + k^{-1}, \tag{44}$$

else

$$\gamma = k(r_1 - r_0), \tag{45}$$

$$r = r_1 + \cos(\gamma)(R - r_0) + \sin(\gamma)(Z - k^{-1}), \tag{46}$$

and

$$x = r \cos(\alpha)$$

$$y = r \sin(\alpha)$$

$$z = \frac{1}{\cos(\gamma)} \left(Z - k^{-1} - \sin(\gamma)(r - r_1) + \frac{\cos(\gamma)}{k} \right). \tag{47}$$

In the initial set of parameters Λ_E, which is a nondeformed model, parameters $r_{0E} = r_{1E} = 0$. k_E is a very small number since, for $k = 0$, the above deformation equations become singular.

VI. Stability and Speed of Model Recovery

The fitting function (7) can be regarded as an energy function on the space of model parameters. In this space, several minima that are equally deep can exist, corresponding to the shape ambiguities described in Section III-C. Some of the deep minima are weak minima. A weak minimum, as opposed to a strong minimum, is when one or several model parameters can take up to a range of values without affecting the function value. For example, a model for a cylindrical object can rotate around the z-axis without changing the value of the fitting function. Besides these significant minima, a profusion of shallow local minima can be expected. This section explores different stability issues in light of this existence of more than one acceptable solution.

A. Local Convexity

First, to show local convexity (Fig. 9) in the neighborhood of a solution, we examined how the fitting function changed when one of the model parameters was changing for the cylinder model in Fig. 17. This simple way of looking at the neighborhood of a solution does not fully reflect the complex relationships in the 11+ dimensional parameter space of deformed superquadric models.

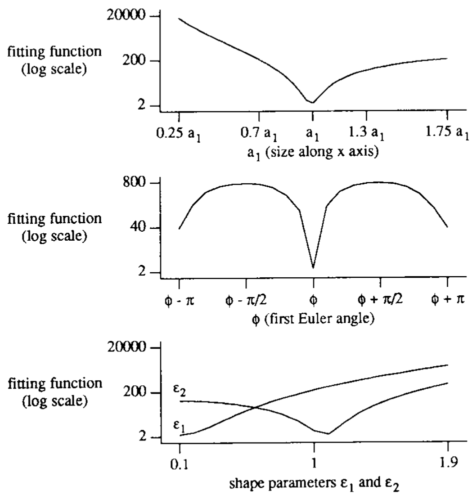

Fig. 9. Influence on the fitting function for the cylinder in Fig. 17 obtained by changing one model parameter at a time. Although all parameters are interdependent, these 2-D plots give some insight into the behavior of the fitting function around a "deep" minimum.

B. Stochastic Technique to Combat Shallow Local Minima

Minimization methods guarantee convergence in general only to a local minimum. It depends on the starting position in the parameter space (Λ_E) to which minimum will the minimization procedure converge. We have to assure that the minimization procedure does not get stuck in a shallow local minimum, but finds the deepest minimum in the close vicinity. Shallow local minima are avoided as solutions during model recovery by adding Poisson distributed noise to the value of the fitting function of the accepted model before comparing it with the value of the fitting function of the model under consideration. Note the spikes in fitting function in Figs. 11 and 18—they are due to the added noise. This stochastic technique introduces "jitter" into the fitting procedure and resembles simulated annealing.

C. Constraining Parameters to Achieve Unique Solutions

When describing ambiguities of superquadric shape parameters in Section III-C, we suggested constraining the parameter space as a way to achieve unique solutions. We took the range image of the box in Fig. 3 and recovered another model, this time with shape parameters constrained to $0.1 < \{\epsilon_1, \epsilon_2\} < 1$. The solution was again a parallelepiped but with $\epsilon_2 = 0.1$ (Fig. 10) instead of $\epsilon_2 = 1.9$ as in Fig. 3.

D. Influence of the Size Parameter on the Fitting Function

The size factor in the fitting function R (7) is necessary to transform weak minima in the inside–outside function

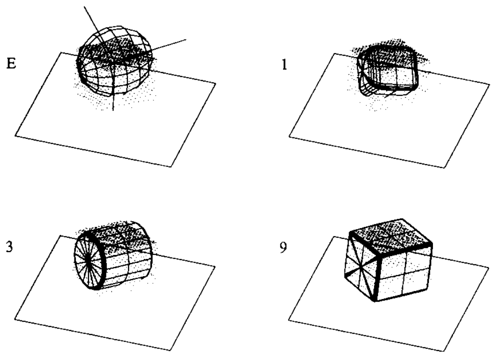

Fig. 10. Model recovery of the same object as in Fig. 3. By constraining the range of shape parameters to $0.1 \leq \epsilon_1, \epsilon_2 \leq 1.0$, another parallelepiped-like model was recovered with $\epsilon_1 = \epsilon_2 = 0.1$.

to strong minima, which correspond to models with the smallest volume that still fit the input range points in the least squares sense. Note that the size factor $\sqrt{a_1 a_2 a_3}$ does not introduce a new global minimum into fitting function R. When a_1, a_2, or $a_3 \rightarrow 0$, fitting function $R \rightarrow \infty$. The size factor in the fitting function also changes the form of the parameter space—compare Figs. 11 and 12. The farther away a point is from a deep minimum ($R \approx 0$) in the parameter space, the larger is the value of the inside-outside function in that point. In the fitting function, this value is multiplied with the size factor, which in general also increases when a point gets farther away from a deep minimum (when any of the two shape parameters increase, the size parameters must also increase, to keep the model's volume about the same size). The result is that, in comparison to the plain inside-outside function parameter space (Fig. 12), the fitting function parameter space has steeper walls around the deep minima (Fig. 11). This eliminates some of the shallow local minima and flattens the rest of the local minima. Larger gradients around the deep minima also cause faster convergence of the minimization algorithm (see Section VI-F).

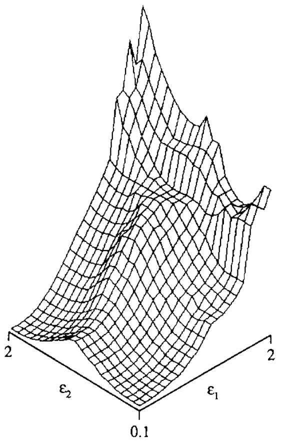

Fig. 11. The above surface is part of the parameter space using the fitting function (7) for recovery of the box-shaped object in Fig. 3. The two "deep" minima correspond to two acceptable solutions: ($\epsilon_1 = 0.1$, $\epsilon_2 = 2$), shown in Fig. 3 and ($\epsilon_1 = 0.1$, $\epsilon_2 = 0.1$), shown in Fig. 10. The surface was computed by fixing the two shape parameters at 400 choices of ϵ_1 and ϵ_2 and computing a minimum in 20 iterations. The sharp peaks on the surface are due to the addition of noise during minimization.

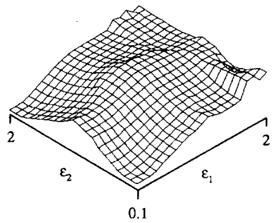

Fig. 12. This is the same surface and for the same object as in Fig. 11, but showing only the inside-outside function (5). Such as in Fig. 11, the surface has two global minima which correspond to two acceptable solutions, but the slopes around the minima are flatter.

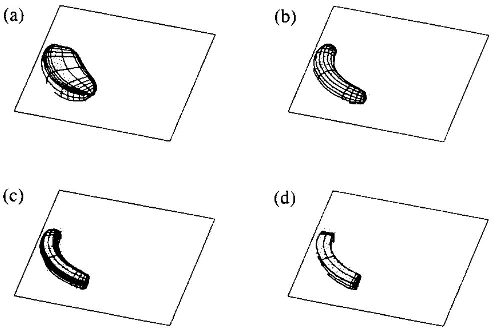

Fig. 13. Models of the same object, recovered in 20 iterations with the following size factors in the fitting functions: (a) $(a_1 a_2 a_3)^{1/3}$, (b) $(a_1 a_2 a_3)^{1/2}$, (c) $(a_1 a_2 a_3)^{2/3}$, and (d) $(a_1 a_2 a_3)$.

We recovered a model of the same object with four different size factors in the fitting function (Fig. 13). The noticeable differences in recovered shape are due to the size factor in the fitting function. Because the largest volume obtainable for a given $a_1 a_2 a_3$ corresponds to parallelepiped shapes, such shapes are favored when the size factor overpowers the inside-outside function. Note that computing the actual volume of the superquadric would not eliminate this bias toward square shapes. The expression for superquadric volume has the form: $K(a_1 a_2 a_3)$, where K is a function (series expansion) of ϵ_1 and ϵ_2. For example, common geometric primitives that can be modeled with superquadrics have the following values of K: for diamond shapes $K = 4/3$, for ellipsoids $K = 4/3\pi$, for cylinders $K = 2\pi$, and for parallelepipeds $K = 8$. The larger the weight of the size factor in the fitting function, the more favored a model with a rectangular cross section (Fig. 13). When testing different size factors on different shapes, we selected the size factor $\sqrt{a_1 a_2 a_3}$ because it gives perceptually better results and the residuals recomputed with the inside-outside function alone are the smallest (Fig. 14). When comparing the rate of convergence using different size factors, we observe that the larger the factor, the steeper are the walls around the deep minima, and the faster is the convergence rate (Fig. 15). An adaptive technique would make the size factor a func-

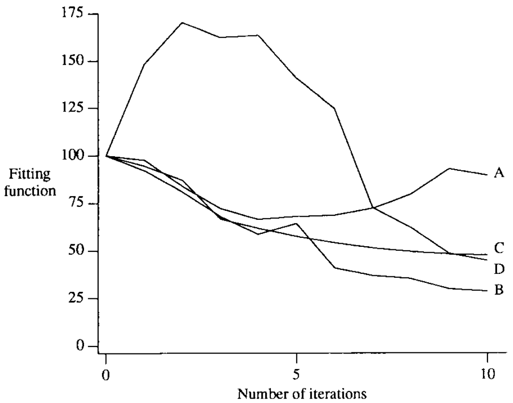

Fig. 14. Residuals, recomputed for the inside-outside function (without the size factor), during the recovery of the four models in Fig. 13: A—$(a_1a_2a_3)^{1/3}$, B—$(a_1a_2a_3)^{1/2}$, C—$(a_1a_2a_3)^{2/3}$, and D—$(a_1a_2a_3)$.

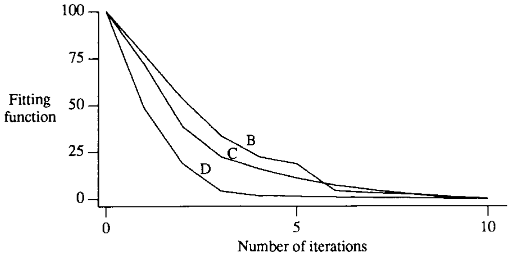

Fig. 15. Convergence rate for three different fitting functions used for recovery of models in Fig. 13: B—$(a_1a_2a_3)^{1/2}$, C—$(a_1a_2a_3)^{2/3}$, and D—$(a_1a_2a_3)$. The fitting functions are scaled differently and their values cannot be compared directly. The larger the size factor in the fitting function, the faster the convergence.

tion of the residual. At the start of minimization when the residual is large, the weight of the size factor should be large to speed up convergence and come close to a deep minimum. Once there, the size factor is reduced to get a better fit.

E. Consistency of Interpretation

To show that the recovery method that we devised has consistent solutions we did the following experiment. We took a range image of the same object in several different positions and orientations and compared the recovered models (Fig. 16). Some of the recovered model parameters can be compared directly, others, such as a_1 and a_2 parameters cannot be compared directly due to the ambiguous orientation of the bending plane. The sum or product of a_1 and a_2 parameters reflects the size of the cross section better.

F. Speed of Model Recovery and Multiresolution

The most time consuming part in the described model recovery is the repeated evaluation of the fitting function and of its partial derivatives for all input range points. Since the sum of least squares is a monotonically increasing function, an iteration can be shortened by monitoring the partial sum after each addition. As soon as the sum is larger than the sum of least squares of the accepted model, it makes no sense to continue. The model cannot be accepted.

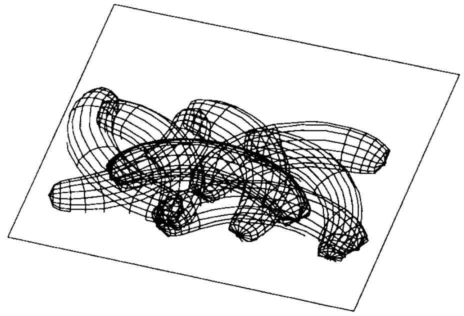

Fig. 16. Recovered models of the same object (a banana) in eight different positions and orientations. All recovered models are perceptually similar.

Faster convergence is possible by improving the method for selecting the trial set of parameters in the minimization method [29]. This reduces the number of rejected iterations because of a too big jump in parameter values.

A substantial speed up can be achieved by subsampling the original range map. The models recovered from coarser range maps can still be a very good representation of the imaged object (Fig. 17). During iterative model recovery, the fitting function typically drops very fast until it reaches a plateau. Further iterations gain no substantial improvements of fit (Fig. 18). It is well known that copies of an image at multiple resolutions, generally known as a pyramid data structure, can dramatically improve the speed and effectiveness of many early vision algorithms [1]. One can view the coarser grids as correction grids, accelerating convergence of the minimization scheme on the finest grid by efficiently liquidating smooth error components [9]. Fast and efficient recovery of superquadric models can be done on a hierarchy of range images with different resolution. We implemented a multiresolution model recovery scheme which starts on a very coarse range map. Once no improvement in fit is made, the minimization continues on a denser range map until the finest or the original range map is reached (Fig. 18). Multiresolution is faster because it takes less time for computation in each iteration and not because of a smaller number of iterations. During multiresolution recovery the number of iterations can get even larger because the models for sparser range maps converge to a somewhat different set of parameters then the model for the finest grid.

Implementing the recovery procedure on a fine grained parallel architecture would be straightforward since the evaluation of the fitting function and its partial derivatives is independent for each range point. By assigning a processor to each range point, very fast model recovery would be possible.

Recovery of models shown in this work, where the

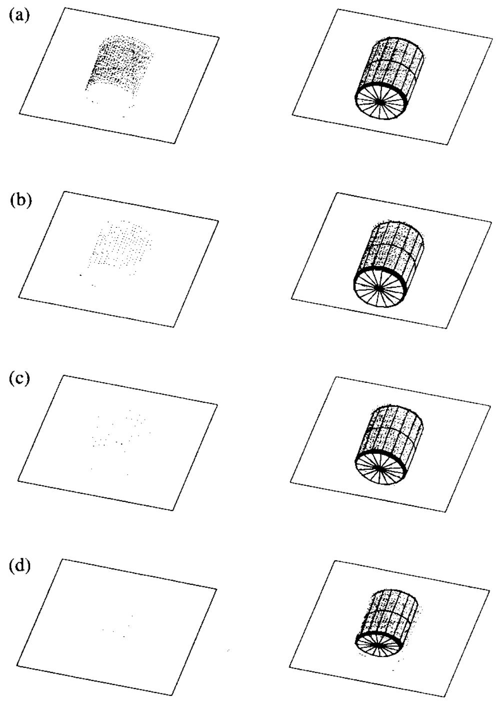

Fig. 17. Influence of coarser range maps on the recovered models. On the left, from top down, are finer to coarser rage maps, obtained by picking every second (b), 4th (c) and 8th range point (d) in the x and y axis of the original range map (a). On the right are the corresponding recovered models which are shown against the original range map.

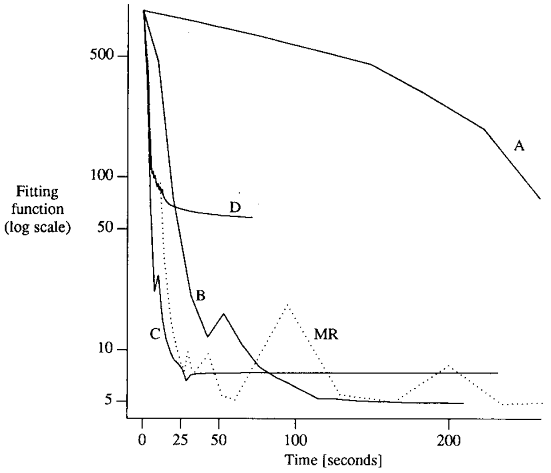

Fig. 18. Fitting function as a function of CPU time during recovery of the four models (a), (b), (c), (d) in Fig. 17. The jaggedness of the function is due to the addition of Poisson noise which enables escaping from shallow local minima. When the fitting function reaches a plateau, the corresponding model cannot improve any more. *MR* shows the fitting function for a multiresolution fitting technique when model recovery starts on the coarsest map and switches to a finer map when the fitting function does not improve any more.

number of range points for each model is on the order of several hundred, took about 20 seconds of CPU time on a VAX 785 computer.

VII. Discussion

We adopted a shape vocabulary [23] which consists of superquadric primitives and deformations. Together with global deformations, superquadrics can model on the level of parts a large variety of geometrical and natural shapes. A new method for recovery of such part models was introduced, which is based on least squares minimization of a newly defined fitting function for all input range points belonging to a single-part object. A gradient descent method combined with a stochastic technique is used for minimization. To start minimization, only estimates for rough position, orientation and size of the part have to be computed. Model's shape parameters, as well as the bending, tapering and cavity deformation parameters, always start with default values. Their actual values are recovered through model recovery. Results of investigating model stability issues, consistency of interpretation and speed of model recovery are reported.

We would like to stress that superquadrics are "not" necessary for the approach to shape recovery that we took. They are just a handy mathematical representation of prototypical parts that can be made to fit the actual shape of parts by scaling and global deformation. Other compact volumetric primitives, like a type of generalized cylinders, can be recovered through free-form fit [33]. The advantage of superquadric primitives is that they can capture a large variety of shapes with a small number of parameters and have nice mathematical properties which make model recovery efficient. We made use of the analytic inside–outside function to precompute analytically the partial derivatives for all model parameters, which speeds up the minimization. In general, any parametric model would make sense as long as its parameters are perceptually relevant.

The adopted shape vocabulary is intended for rough description of parts, suitable for object avoidance, grasping or shape classification/recognition of basic categories [27]. Inadequate representation can occur if the occluded side of an object is not symmetrical to the visible side, or because the deformations that we use are defined only along the longest side of the object or along the axis of rotational symmetry. Although deformations are sufficient most of the time, they do not cover all possible cases. Deformations used in this paper are quite simplified, but with more parameters, more accurate modeling is possible. Details in the context of this part-level modeling, on the other hand, could be better described with surface representations, or some formative processes like fractals [23]. The level of required description depends ultimately on the task the vision system has to perform. The recovered model in Fig. 19, for example, is not detailed enough to recognize the object, but good enough to grasp or avoid it. When refinement of shape is required, the recovered part model gives a good starting point for recovery of local surface patch models. For refinement of shape representation, local deformations could be used by applying local push and pull to the part models [3]. Models

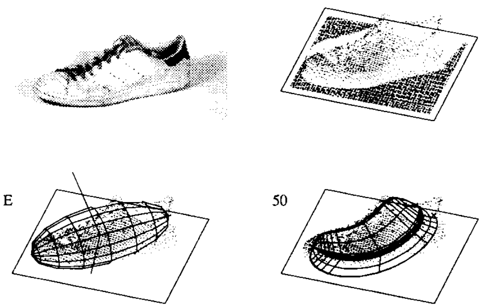

Fig. 19. Model, recovered for an irregular object—a tennis shoe.

could be enhanced also with parameters representing some physical properties related to the real world, like elasticity or stiffness [32].

What seems to be important is the chosen granularity of representation. There is psychological evidence showing the particular salience of parts—especially on the basic category level [35]. For objects and biological categories, basic category cuts seem to follow natural breaks in the structure of the world and this structure is determined by part configuration. Parts and part configuration seem to form a natural bridge connecting perception (appearance) of objects and behavior (activity) toward them, and in turn communication about them. Perceived part configuration underlies both perceived structure and perceived function, and forms the basis of the intuitive causal reasoning and naive induction. The basis of naive induction is that separate parts have separate functions, similar parts have similar functions, and more salient parts have more important functions. This close relation opens up possibilities to infer function from shape [36]. There is also ample evidence that human perception makes use of the prototype and deformation paradigm. Using deformations is an efficient way of representing a large variety of shapes and has a causal structure because it is related to the process of how the shape evolved over time. Leyton [19] has shown that prototypification of shapes can be done with a decomposition into primitive transformations (deformations) and that a particular ordering is psychologically salient. The hierarchical structure of the model we used seems to have also a perceptual significance in the sense that one has to know first where an object is and how it is oriented, to recover its size and shape. This order resembles the structure of solving geometry analogy tasks which involve finding first location, orientation, size, general outline, and additional small parts and details at the end [21].

The model parameter space (superquadrics with deformations) is continuous, but some sets of parameters correspond to easily identifiable geometric primitives such as parallelepipeds, ellipsoids and cylinders. For those categories, the within-category parameter differences look smaller than between-category parameter differences even when they are of the same size. This differential responding is referred to as categorical perception [15]. By mapping symbols on the continuous model parameter space it is possible to define distinctive volumetric primitives such as Geons, a finite set of shape primitives believed to correspond to letters in a shape vocabulary [6]. A representation, using prototypes associated with category names, is more compact and more likely used for cognitive activities such as recognizing, learning and remembering. Recovering the larger structure of the world before recovering the details is very compelling since human vision also seems to proceed from general to particular and not the other way around. People do not apprehend shape by patching it together through the tracing of its parts but seize an overall pattern first [13].

Most methods for recovery of midgrain or larger models in computer vision are based on circumstantial evidence, typically using rules encoding some perceptually relevant relation. We view the importance of the introduced model recovery in the mathematical formulation of the problem. This enables constant monitoring of the goodness of fit, enabling to stop when the fit is sufficient or detecting that the model is inappropriate when the residual does not drop as expected.

In this work we used range data exclusively. In this way, shape representation and recovery could be studied in isolation from problems specific to particular modalities (shape from x). We believe that other shape cues can be used as input for this model recovery method, especially if several visual cues are used in combination. Model recovery could serve as a medium for merging information from different cues. Since shape information from different visual cues can be basically interpreted as position (passive stereo, focusing), and orientation (shading, texture, occluding contours) the expanded fitting function would have two types of terms; the first ones, like the one we used here, for position, and a second type of terms, for surface normals [2].

Part models seem to be the right level for integrating information from several views and over the time. For integration, it would be beneficial to include in the model some information indicating, which part of the model was based on the input data, and which part was occluded but inferred from the model symmetry.

During experimentation we faced the problem of nonuniform range data density and a large number of singular views in range images. For example, if only one face of a cube is seen, a very thin parallelepiped is recovered which fits to that face. Images taken with a passive range imager, which uses triangulation, have more singular views than normally associated with intensity images. The larger the distance (angle) between the source of illumination and the camera of the range imager, the better the accuracy, but the more singular views, since every recorded range point must be illuminated *and* seen by the camera at the same time. Singular views can be resolved by taking into account the structure of the surrounding scene. Objects normally rest on some support, they can touch but usually do not penetrate each other. We believe that this type of constraint could be incorporated in the part recovery procedure by a method devised by Witkin, Fleischer, and Barr [38]. They developed an elegant method for describing geometric constraints between parts or objects in terms of energy. To solve for all constraints, the sum of all corresponding energy terms, including the shape fitting function, would have to be minimized. The inclusion of such constraints into this shape recovery would be a first step towards segmentation based on part-level shape recovery [30].

Nonuniform range point density causes that parts with higher density have more influence on the shape of the recovered model then parts with lower density. However, the model recovery method is quite robust in this regard—note that no range data is available from occluded parts to begin with. During minimization, we used the same weight for all range points and all model parameters. By using a larger weight (penalty) for points outside the model, a convex-hull-like model can be recovered, and by putting different weights on parameters, a perceptually relevant ordering of parameter importance can be enforced.

Acknowledgment

We wish to thank S. Pentland for encouraging us to use superquadric models and M. Mintz for helping with the minimization procedure.

References

[1] D. H. Ballard and C. M. Brown, *Computer Vision.* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[2] R. Bajcsy and F. Solina, "Three dimensional shape representation revisited," in *Proc. First Int. Computer Vision Conf.*, London, England, 1987, pp. 231-241.

[3] R. Bajcsy and S. Kovačič, "Multiresolution elastic matching." *Comput. Vision, Graphics, Image Processing J.*, vol. 46, no. 1, 1989.

[4] A. H. Barr, "Superquadrics and angle-preserving transformations," *IEEE Comput. Graphics Applicat.*, vol. 1, pp. 11-23, 1981.

[5] A. H. Barr, "Global and local deformations of solid primitives," *Comput. Graphics*, vol. 18, no. 3, pp. 21-30, 1984.

[6] I. Biederman, "Human image understanding: Recent research and theory," *Comput. Vision, Graphics, Image Processing*, vol. 32, pp. 29-73, 1985.

[7] R. C. Bolles and P. Horaud, "3DPO: A three-dimensional part orientation system," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 3-26, 1986.

[8] T. E. Boult and A. D. Gross, "Recovery of superquadrics from depth information," in *Proc. Spatial Reasoning and Multi-Sensor Fusion Workshop*, St. Charles, IL, 1987, pp. 128-137.

[9] A. Brandt, "Multi-level adaptive solutions to boundary-value problems," *Math. Computation*, vol. 31, no. 138, pp. 333-390, 1977.

[10] R. A. Brooks, "Model-based 3-D interpretation of 2-D images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, no. 2, pp. 140-150, 1983.

[11] J. A. Fodor, *The Modularity of Mind*, Cambridge, MA: M.I.T. Press, 1983.

[12] M. Gardiner, "The superellipse: A curve that lies between the ellipse and the rectangle," *Sci. Amer.*, vol. 213, pp. 222-234, 1965.

[13] E. H. Gombrich, *Art and Illusion.* Oxford, England: Phaidon, 1962.

[14] W. E. L. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robotics Res.*, vol. 3, no. 3, pp. 3-35, 1984.

[15] S. Harnad, "Category induction and representation," in *Categorical Perception*, S. Harnad, Ed. Cambridge, England: Cambridge University Press, 1986.

[16] B. K. P. Horn, *Robot Vision.* Cambridge, MA: M.I.T. Press, 1986.

[17] J. R. Kender and D. Freudenstein, "What is a 'degenerate view'?" in *Proc. DARPA Image Understanding Workshop*, Los Angeles, CA, 1987, pp. 589-598.

[18] J. Koenderink and A. van Doorn, "The internal representation of solid shape with respect to vision," *Biol. Cybern.*, vol. 32, pp. 211-216, 1979.

[19] M. Leyton, "Nested structures of control: An intuitive view," *Comput. Vision, Graphics, Image Processing*, vol. 37, pp. 20-53, 1987.

[20] D. Marr, *Vision.* San Francisco, CA: Freeman, 1982.

[21] L. R. Novick and B. Tversky, "Cognitive constructs on ordering operations: The case of geometric analogies," *J. Exp. Psychol. General*, vol. 116, no. 1, pp. 50-67, 1987.

[22] R. Paul, *Robot Manipulators.* Cambridge, MA: M.I.T. Press, 1981.

[23] A. P. Pentland, "Perceptual organization and the representation of natural form," *Artif. Intell.*, vol. 28, no. 3, pp. 293-331, 1986.

[24] —, "Recognition by parts," in *Proc. First Int. Computer Vision Conf.*, London, England, 1987.

[25] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vettering, *Numerical Recipes.* Cambridge, England: Cambridge University Press, 1986.

[26] K. Rao and R. Nevatia, "Computing volume descriptions from sparse 3-D data," *Int. J. Comput. Vision*, vol. 2, no. 1, pp. 33-50, 1988.

[27] E. Rosch, "Principles of categorization," in *Cognition and Categorization*, E. Rosch and B. Lloyd, Eds. Hillsdale, NJ: Erlbaum, 1987.

[28] A. Rosenfeld and A. Kak, *Digital Picture Processing.* Orlando, FL: Academic, 1982.

[29] L. E. Scales, *Introduction to Non-Linear Optimization.* New York: Springer, 1985.

[30] F. Solina and R. Bajcsy, "Range image interpretation of mail pieces with superquadrics," in *Proc. AAAI-87*, Seattle, WA, 1987, pp. 733-737.

[31] G. Strang, *Linear Algebra and its Applications.* Orlando, FL: Academic, 1980.

[32] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," in *Proc. SIGGRAPH-87*, Anaheim, CA, 1987.

[33] D. Terzopoulos, A. Witkin, and M. Kass, "Symmetry-seeking models for 3D object reconstruction," *Int. J. Comput. Vision*, vol. 1, no. 3, 1988.

[34] G. Tsikos, "Segmentation of 3-D scenes using multi-modal interaction between machine vision and programmable mechanical scene manipulation," Ph.D. dissertation, Univ. Pennsylvania, Philadelphia, 1987.

[35] B. Tversky and K. Hemenway, "Objects, parts, and categories," *J. Exp. Psychol.: General*, vol. 113, no. 2, pp. 169-193, 1984.

[36] P. H. Winston, T. O. Binford, B. Katz, and M. Lowry, "Learning physical description from functional descriptions, examples, and precedents," in *Proc. AAAI-83*, Washington, DC, 1983, pp. 433-439.

[37] A. P. Witkin and J. M. Tenenbaum, "On the role of structure in vision," in *Human and Machine Vision*, J. Beck, B. Hope, and A. Rosenfeld, Eds. New York: Academic, 1983.

[38] A. P. Witkin, K. Fleischer, and A. Barr, "Energy constraints on parameterized models," in *Proc. SIGGRAPH-87*, Anaheim, CA, 1987.

Franc Solina received the Dipl.Ing. and M.S. degrees in electrical engineering from the University of Ljubljana, Yugoslavia, in 1979 and 1982, respectively, and the Ph.D. degree in computer science from the University of Pennsylvania, Philadelphia, in 1987, where he was affiliated with the GRASP Laboratory (General Robotics and Active Sensory Perception).

He is now an Assistant Professor of Computer Science at the University of Ljubljana, Faculty of Electrical Engineering and Computer Science. His main research interests are models for shape representation and segmentation in vision.

Dr. Solina is a member of the American Association of Artificial Intelligence.

Ruzena Bajcsy (M'81-SM'88) received the M.S. degree in mathematics and her first Ph.D. degree in electrical engineering from the Slovak Technical University in 1957 and 1967, respectively. She received her second Ph.D. degree in computer science from Stanford University, Stanford, CA, in 1972.

She joined the Department of Computer and Information Science at the University of Pennsylvania in 1972 where she now serves as Professor and Chairman of the department as well as Director of the GRASP Laboratory. Her research interests include computer vision, multisensor integration, and robotics.

Dr. Bajcsy has authored numerous book chapters and journal publications and has served as editor and associate editor of several journals, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.