Shape representation and image segmentation using deformable surfaces

H Delingette, M Hebert and K Ikeuchi

We present a technique for constructing shape representation from images using free-form deformable surfaces. We model an object as a closed surface that is deformed subject to attractive fields generated by input data points and features. Features affect the global shape of the surface or unstructured environments. The algorithm is general in that it makes few assumptions on the type of features, the nature of the data and the type of objects. We present results in a wide range of applications: reconstruction of smooth isolated objects such as human faces, reconstruction of structured objects such as polyhedra, and segmentation of complex scenes with mutually occluding objects. We have successfully tested the algorithm using data from different sensors including grey-coding range finders and video cameras, using one or several images.

Keywords: range data, 3D models, deformable surfaces

The recovery of object shape from **3D** is one of the key issues in computer vision. One could define this task as the segmentation of a large set of data points into shapes corresponding to objects in the scene. Coupled with the extraction of objects from a scene is the issue of finding an efficient computational representation of object shapes. The shape representation should be general enough to handle a wide variety of scenes, yet simple enough to be usable for other tasks such as recognition and manipulation. In other words, the shape representation should have enough parameters to describe the specificity of the shape, but must have as few parameters as possible to be usable, and **to** be robustly extracted from visual data. This conflict can be seen as a scale space problem, where one would like to find a description fine enough to capture the key details of the shape, but coarse enough to get rid of spurious details.

In this paper we propose an approach that attempts to solve this conflict by using the feature/data duality in a way similar to the fine/coarse approach. Several psychophysical experiments have proved that the human eye is able to capture the main shape of an object by seeing only a few characteristic elements or features. These features can be either geometric (distance discontinuities, surface orientation discontinuities, corners, minimum of curvature, etc.), or higher level such as reflectance properties. While these features capture most of the shape information, it is difficult without ***a priori*** knowledge to build a full reconstruction of the object.

Several solutions have been proposed. Besl and Jain' built curvature-based object representations by classifying surfaces according to the sign of their principal curvatures. Pentland and Sclasoff[2] presented a physically-based algorithm to recover a model in a unique manner from a set of features and vibration modes. In this approach, the key is to find the correct features, thus restraining the algorithm to either smooth or structured shapes. Another approach is to use both sets, according to geometric properties (symmetry, connexity, etc.), and in a second stage models are fit to the segmented parts. This idea of hierarchical representation was initiated by Marr and Nishihara[3] and pushed further by the seminal **ACRONYM**[4] vision system by using generalized cylinders. Pentland's 'representation by parts' using deformed superquadrics'. [6], proved to have some successful results, but encounters some limitations. While the feature grouping requires some accurate feature extraction and high level reasoning, the fitting of superquadrics[5,7] to the range data has some unstable behaviour, due to its non-linear nature, and is suitable for only smooth and simple shapes.

Those techniques attempt to represent all shapes with a set of elementary shapes (superquadrics, generalized cylinders, parametric patches, etc.) that can be described by a few parameters. This is clearly

The **Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA** 15213, **USA**

0262-8856/92/003132-13

. beneficial from the point of view of object recognition which amounts to manipulating analytical equations of the elementary shapes. In practice, however, it restricts considerably the class of objects and scenes to which the techniques can be applied. More general representations could be obtained by adding degrees of freedom to the elementary shapes (e.g. adding tapering and bending to superquadrics). However, the non-linear fitting algorithms involved in the recovery of such shapes become rapidly computationally expensive and numerically difficult. Furthermore, most of those techniques assume a pre-segmentation of the scene into regions corresponding to individual objects in order to successfully carry out the surface fitting algorithm. In realistic situations, this pre-segmentation is however a difficult task due to noise and occlusions.

To address those problems, Terzopoulos and Witkin[8,9] proposed the concept of deformable contours and deformable surfaces that are subject to forces generated by image elements. Given a tube initialized around the line of symmetry of an object, the surface is deformed by a potential field computed from an intensity image, until it matches the main shape of the object. While this work demonstrates that free-form surfaces can be used to efficiently represent a wide variety of objects, it still constraints the object to be both symmetric and smooth, and it does not address the local minima problem. Because the deformation of the surface occurs only locally, the surface has to be initialized close to the final position. Terzopoulos and Metaxas[10] combined the elementary shape and the deformable surface approaches to gain a local/global representation.

Our approach is to use a free-form representation in order to retain the generality of the shape, and to use a local/global deformation scheme in order to get a robust shape extraction. Given an observed scene, a surface is initialized in the vicinity of detected features, The surface is then deformed subject to forces generated by features and data points. The forces generated by data points control ***local*** shape, while forces generated by features control ***global*** shape. A smoothness energy is added to the deformation equations to take into account the fact that data and features may be sparse and noisy. Clearly, such free-form surfaces can represent a large class of objects since few constraints are put on the resulting shape. Furthermore, the algorithm must be robust enough so that it does not require a precise segmentation of the scene as an input.

DEFORMABLE SURFACES

In this section we describe the deformation scheme that is used to model different objects in a scene. We use a free-form representation which means that the surface is defined in terms of parametric equations. Since the surface is deformed during the algorithm, the surface is parametrized with three parameters (u, v, t), where (u, v) correspond to the spatial parameters, and t corresponds to the time parameter. By using explicit time dependance, we can model the surface as a mechanical system, and therefore use the classical Lagrangian equation to formulate the deformation process. Let (O, i, j, k) be a frame of $\mathcal{R}^3$, U be a connected open set of $\mathcal{R}^2$, then the surface $S(t)$ at time t, $t \in [0, T_0]$ is defined by:

$$(P(u, v, t) \in S(t)) \Leftrightarrow \left(OP = \mathbf{r}(u, v, t) = \begin{bmatrix} X(u, v, t) \\ Y(u, v, t) \\ Z(u, v, t) \end{bmatrix} \right)$$
$$((u, v) \in U, t \in [0, T_0])$$

The use of a parametric form of $S(t)$ has several advantages over an explicit form $z = f(x, y)$: first, it allows a viewpoint independent representation of the object. In other words, the representation of the object is independent of the frame associated with the sensor that delivered the range data. Bolle and Vemuri[11] pointed out the importance of viewpoint invariance for surface reconstruction methods in order to build models suitable for recognition tasks. In particular, this invariance is essential in order to perform fusion of data from different sensors: we will give several examples of object modelization using several range finders. Second, the parametric form allows one to handle objects that do not have a planar topology. We will use surfaces that are topologically equivalent to a sphere, but other topology could be used without modification of the algorithm.

Given an initial surface $S(t = 0)$, different actions force the surface to deform until it matches the shape of the object. The deformation process is ruled by the equations of motion derived from the laws of classical mechanics and that involve three types of forces: external, internal and inertial forces.

External forces: are generated by input data points and input features. External forces apply deformations that bring the surface as close as possible to the data corresponding to the object. Their influence is both local and global: local in order to get a faithful model of the object and global in order to avoid the local minima problem, and therefore to increase the robustness of the algorithm. The local influence is provided by the range data available for the scene while the global influence is exclusively provided by feature points. Those feature points are points of the scene that contains some high level information which reveals the existence of the object. This information can be of geometric nature (discontinuities, jumps, . . .), of reflectance nature (intensity discontinuity) or of any other depending on the sensors available. Those points are first computed, clustered into segments and then are used by the deformation program.

Because of the global deformation entailed by features, we do not need to predefine a complete mapping between data and surface points as was required in earlier work on surface reconstruction. This mapping assumes an initial segmentation of the object from the scene and above all requires the surface to be initially very close to the solution.

Internal forces: are generated by the surface itself as it is deformed. The inclusion of internal forces ensures that the surface will not tear apart, fold onto itself, or exhibit high curvature points or sharp discontinuities in curvature. The other role of internal forces is to

provide constraints in regions in which little or no data is available. This is similar to the regularization approach to surface reconstruction from sparse data[11,12]. The standard way of defining internal forces is to define the corresponding energy as the integral over the surface of the magnitude of the first and second derivatives[12–15] which characterize the surface smoothness. The relative importance of external and internal forces is a trade-off between accuracy and smoothness. High internal forces generate a very smooth surface that may be far from the input data. Low internal forces allow the surface to fit the data closely but they also allow the surface to fit any noise in the data.

Inertial forces: are generated by the motion of the surface as it evolves over time assuming that the surface has a non-zero mass. Inertial forces are necessary to model the deformable surface as a dynamic mechanical system.

In this section we first describe the components of the three types of energy and the equations of motion, then describe in detail how each energy is computed in the case of a continuous deformable surface. We then describe the computation of the energies and the implementation of the equations of motion in the case of a discrete deformable surface.

General equations of motion

The internal and external energies involved in the deformation of the surface are:

- ***Smoothness energy*** $E_{smoothness}$: the smoothness energy is related to the geometric property of the surface. The smoothness energy is internal in that it depends only on the shape of the surface in the vicinity of each point.
- ***Feature energy*** $E_{feature}$: the feature energy quantifies the interaction between the features and the surface; its magnitude is a function of the distance between surface point and feature and of time t.
- ***Data energy*** E_{data}: the data energy quantifies the effect of data points on the surface.

To calculate the equilibrium position of the surface using mechanical systems theory, we need to introduce two additional inertial energy terms:

- ***Kinetic energy T*** a mass μ is associated with each data point thus generating a kinetic energy term. Unlike other dynamic splines[8,9], our scheme uses explicitly the kinematic energy to link the deformation of the surface with the minimization of energy.
- ***Raleigh Dissipation energy D:*** the dissipation term is added to simulate the exchange of energy between the dynamic surface and a virtual medium in which it evolves. This damping term is added to avoid cases in which the surface oscillates around an equilibrium position.

Following the equations of mechanics and the principle of least action[16], the surface reaches a stable equilibrium when the Lagrangian of the system reaches a minimum. The Lagrangian of the system is:

$$L = T - E_{Smoothness} - E_{Feature} - E_{Data}$$

Using the calculus of variations, the condition for which $L(u, v, t)$ is minimum is (similar to Terzopoulas *et al.*[9]):

$$\mu \cdot \frac{\partial^2}{\partial t^2} \boldsymbol{r}(u, v, t) = -k \cdot \frac{\partial}{\partial t} \boldsymbol{r}(u, v, t) + \boldsymbol{F}_{Smoothness} + \boldsymbol{F}_{Feature} + \boldsymbol{F}_{Data}$$

where $\boldsymbol{r}(u, v, t)$ is the position vector of a point $P(u, v, t)$ on the surface, μ is the mass density of the surface and k is the damping factor.

Smoothness energy

We use the bivariate generalized spline functionals of the first and second order as a smoothness measure[17]. The first order is a measure of the distance discontinuities, while the second order is a measure of the surface orientation discontinuities. Denoting partial derivatives by subscripts (e.g. $\boldsymbol{r}_u = (\partial/\partial u)\boldsymbol{r}(u, v, t)$), the energy is defined by:

$$E_{smoothness1} = \alpha_1 \cdot \int_0^{T_0} \left(\int_u \int_v (\|\boldsymbol{r}_u\|^2 + \|\boldsymbol{r}_u\|^2 \cdot du \cdot dv) \right) dt$$

$$E_{smoothness2} = \alpha_2 \cdot \int_0^{T_0} \left(\int_u \int_v (\|\boldsymbol{r}_{uu}\|^2 + 2\|\boldsymbol{r}_{uv}\|^2 + \|\boldsymbol{r}_{vv}\|^2 \cdot du \cdot dv) \right) dt$$

The corresponding force is therefore:

$$\boldsymbol{F}_{smoothness} = -\alpha_1 \cdot \begin{bmatrix} \Delta_{uv} X \\ \Delta_{uv} Y \\ \Delta_{uv} Z \end{bmatrix} + \alpha_2 \cdot \begin{bmatrix} \Delta_{uv}\Delta_{uv} X \\ \Delta_{uv}\Delta_{uv} Y \\ \Delta_{uv}\Delta_{uv} Z \end{bmatrix}$$

where Δ_{uv} denotes the Laplacian operator with respect to (u, v). The coefficients α_1 and α_2 control the relative smoothness of the surface.

Feature energy

Because every feature contributes to the global deformation of the surface, our approach is to link every feature to every point of the surface (see Figure 1). Therefore, if $E^i_{feature}$ is the energy between a point on the surface and the feature number i, the total feature energy of the surface is:

$$E_{Feature} = \sum_{i=0}^{n} \left[\int_0^{T_0} \left(\int_u \int_v (E^i_{feature} \cdot du \cdot dv) \right) \right]$$

Our algorithm satisfies two requirements:

- The initial surface can be 'far' from the object.
- The points are going to concentrate toward the features. Because regions that enclose features are very important to describe the object (edges in a polyhedron, eyes and nose for a human face, ...),

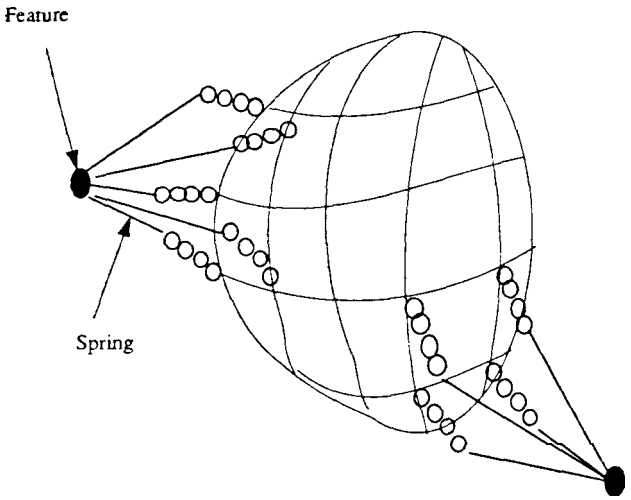


Figure 1. Links between features and surface points

our scheme is therefore able to render a model that has the same hierarchical description.

To get both global and local deformations is hard because the two types of deformation, feature and data, should be balanced. If only feature deformation were applied one would get a surface that connects the features with mostly planar surfaces in between because of the smoothness constraint. On the contrary, in the case of data deformation, one would get a surface that would faithfully follows the shape of the object only where the surface is close enough initially (see Figure 2).

Our solution is to change the relative influence of both types of deformations over time: initially, the surface is mostly influenced by feature forces, therefore moving toward its global shape; then the feature forces decrease and the surface is deformed locally so that it smoothly interpolates the shape of the object. To achieve this shift over time, the expression of $E^i_{feature}$ is defined as follows: let the distance between the feature and the point P on the surface be $D(Feature, P)$ and let a reference distance be $Dref(t)$ at time t:

- if $D(Feature,P) > D_{ref}(t)$ then the feature does not attract the point P: $E^i_{feature} = cste$.
- if $D(Feature,P) < D_{ref}(t)$ then the feature attracts the point P like a spring: $E^i_{feature} = cste \cdot [D(Feature,P)]^2$.

By setting $D_{ref}(t) = D^i_0 \cdot (T_0 - t)$, this reference distance decreases linearly from D^i_0 to 0 which means that point of the surface has to get closer and closer to be attracted by the feature, as t increases. Therefore the influence of the features becomes smaller during the deformation.

In order to avoid any discontinuity, we use a formulation that acts smoothly when $D(Feature,P)$ is close to $D_{ref}(t)$. In the current implementation, we model features as 3D line segments. We denote the midpoint of the segment by F_i, and its length by l_i. With these notations, the energy field generated by feature i, $E^i_{feature}$, (see Figure 5(b)) is:

$$E^i_{feature} = U\left(\frac{\|PF_i\|}{D^i_0 \cdot (T_0 - t)}\right)$$

where:

- $U(x)$ is a function that is quadratic if $x < 1$, constant if $x > 1$ and a cubic polynomial if x is close to 1 (see Figure 3a). Therefore $U(x)$ is C^1 continuous, leading to an expression of the force that is C^0 continuous.
- $\|PF_i\|$ represents the distance between the point $P(u, v)$ of the surface and the feature segment of middle point F_i. Other feature representations can be used in the same framework by replacing this term by the appropriate value $D(Feature, P)$. For example, for a point feature it would be the distance between surface point and feature point.
- D^i_0 is the distance between the feature and the centre of the surface at its initial position. If the surface is initialized as a sphere, it is the distance to the centre of the sphere.

Figure 4 shows how the potential field varies over time in the case of four coplanar features. The centre of the surface does not coincide with the centre of gravity of the features in this example.

From the definition of feature energy, the force generated by the feature i at surface point P at time t is the opposite of the gradient of $E^i_{feature}$, i.e.:

$$F^i_{feature} = \frac{-l_i}{D_0 \cdot (T_0 - t) \cdot \|PF_i\|} \cdot \frac{dU}{dx}\left(\frac{\|PF_i\|}{D^i_0 \cdot (T_0 - t)}\right) \cdot PF_i$$

where dU/dx is the derivative of $U(x)$ see Figure 3c.

Figures 3a and c show the variation of the energy and force with respect to time, and how this force varies with respect to time and distance $D(Feature, P)$; there is no singularity when t is close to T_0. By continuity, we can set $E_{feature}(t = t_0) = cste$ and $F_{feature}(t = T_0) = 0$.

Forces from each feature may be weighted to reflect the relative importance of different types of features.

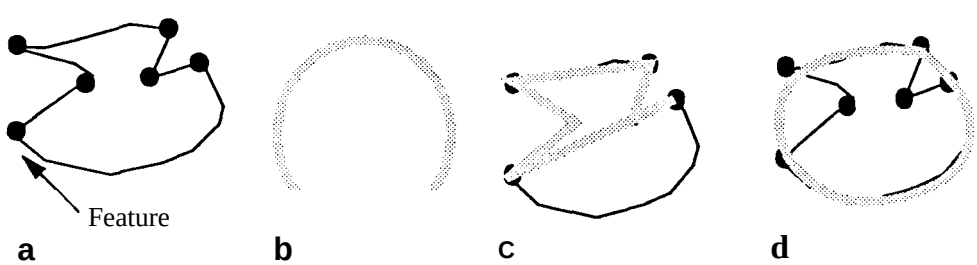


Figure 2. (a) Object with its features (normal discontinuties); (b) surface initialization; (c) final shape if feature deformation is predominant; (d) final shape if data deformation is predominant

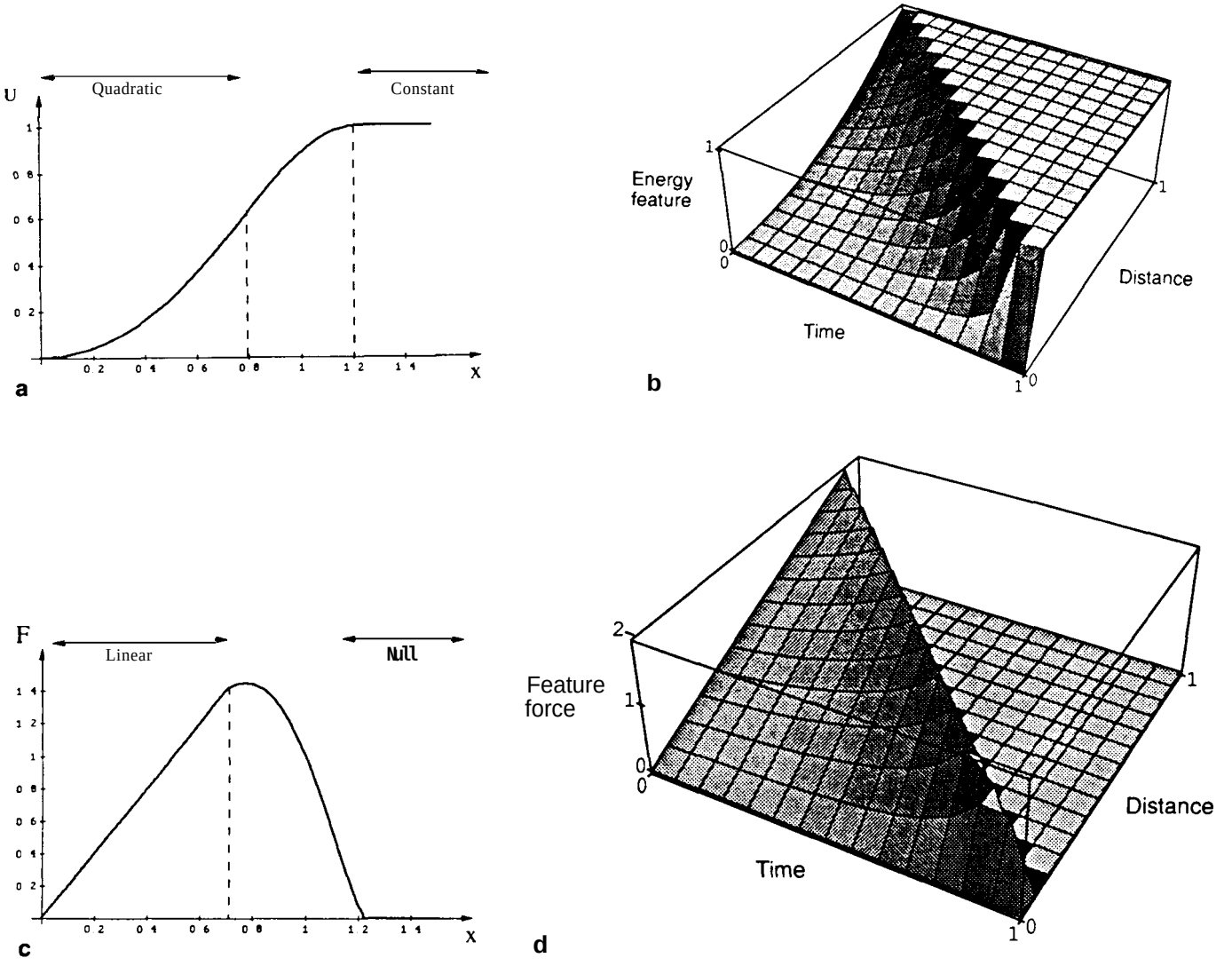


Figure 3. (a) Curve U(x): (h) *evolution of* $E_{feature}$ *as a function of time arid distance between surface point and feature;* (c) *curve* F = dU/dx(x); *(d) evolution of* $F_{feature}$

Data energy

Theoretically, a surface point is subject to forces from all the data points. However, for computational reasons we take into account only the closest data point. For every point P of the surface, the closest data point is denoted by C_{data}. Since data information is used only for local deformation, the corresponding force should decrease sharply with distance. Therefore, a gravity-type field where the energy decreases with the inverse of the distance is appropriate for this type of deformation. But to avoid the singularity when the distance is null, C_{data} acts like a spring when the point $P(u, v)$ is close to C_{data}. If E_{data} is the data energy of a point $P(u, v)$ of the surface:

$$E_{data} = W\left(\frac{\|PC_{data}\|}{K}\right) \quad .$$

$W(x)$ is a function that is quadratic if $(x < 1)$ and in $1/r$ if $(x > 1)$, (see Figure 7). K is a normalizing constant that has the dimension of a distance. Intuitively, K represents the range of the influence of the data on the surface. In practice, K is chosen small compared to the maximum expected object size. The corresponding force is K:

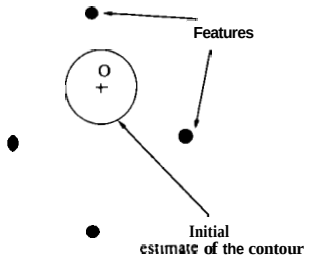
$$F_{data} = -\nabla E_{data} = \frac{\frac{dW}{dx}\left(\frac{\|PC_{data}\|}{K}\right)}{\|PC_{data}\| K} \cdot PC_{data}$$

where dW/dx is the derivative function of $W(x)$ (see Figure 6). It can be noticed that for $x > 2$ the force is very small which means that if $\|PC_{data}\| > 2.K$ the action of the data point is negligible.

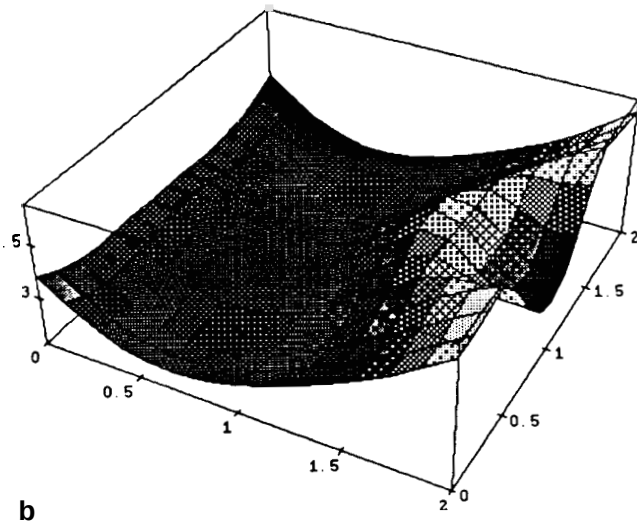
Implementation

We have assumed so far that our model is a continuous surface topologically equivalent to a sphere parameterized in $(u, v,$ t). In practice, however, we can manipulate only discrete surfaces. This raises the problem of the parameterization of such surfaces and, in particular, the impossibility to map a sphere into a square in a uniform way. To avoid creating poles, we adopt the tesselated icosahedron as a structure. Each

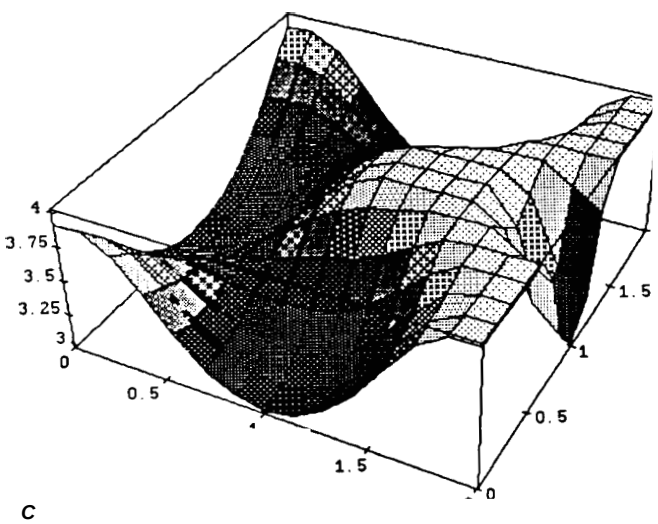
Figure 4. (a) Features and initial position of the surface; (b) initial potential field; (c) intermediate potential field; (d) final potential field



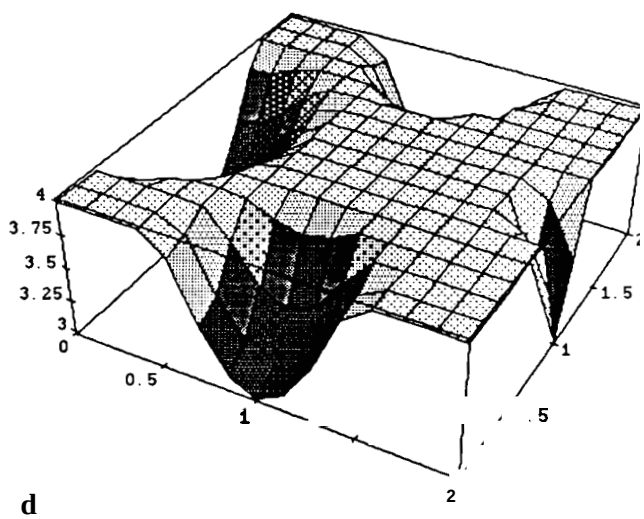
a



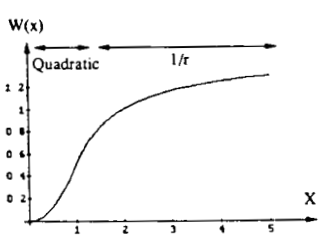
b



c



d



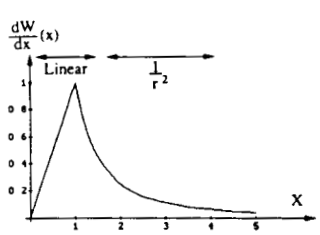


Figure 5 (left). Curve W(x). *Figure 6 (right). Curve* dW/dx(x)

face of the icosahedron is subdivided to yield arbitary resolution of the parameter space. The number of faces of the tessellation is $20N^2$, where N is the density of the subdivision. Typically we use N = 5 yielding a decomposition of the parameter space into 500 faces (see Figure 7). We use the centre of each triangle as a node, every node having therefore three neighbours. If we write r_i^t as the position of the node i at the time t, then the discrete version of the motion equation is:

$$r_{t+1}^i = r_t^i + (1-k)\cdot(r_t^i - r_{t-1}^i)\,\mathrm{t}$$

$$F_{smoothness} + F_{data} + \sum_{i=0}^{n} F_{Feature}^i$$

The surface is initialized as a sphere at $t = 0$ and is deformed by applying repeatedly the equation of motion at each node. Forces F_{data} and $F_{feature}$ are computed at each node independently in a parallel manner. $F_{smoothness}$ is computed by approximating the first and second derivatives of the surface by finite differences. The most expensive part of the algorithm is the computation of the closest data point C_{data} used in the computation of F_{data}. It is theoretically in $O(lm)$,

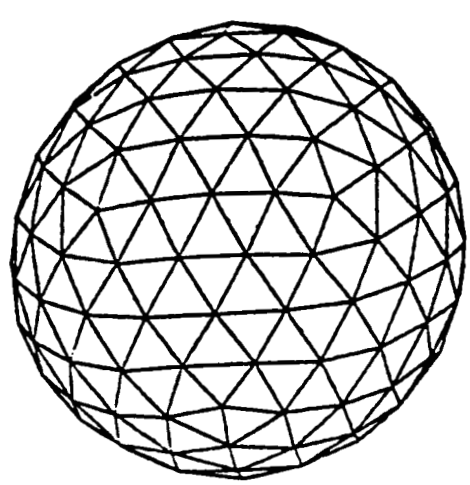


Figure 7. Tessellated icosahedron

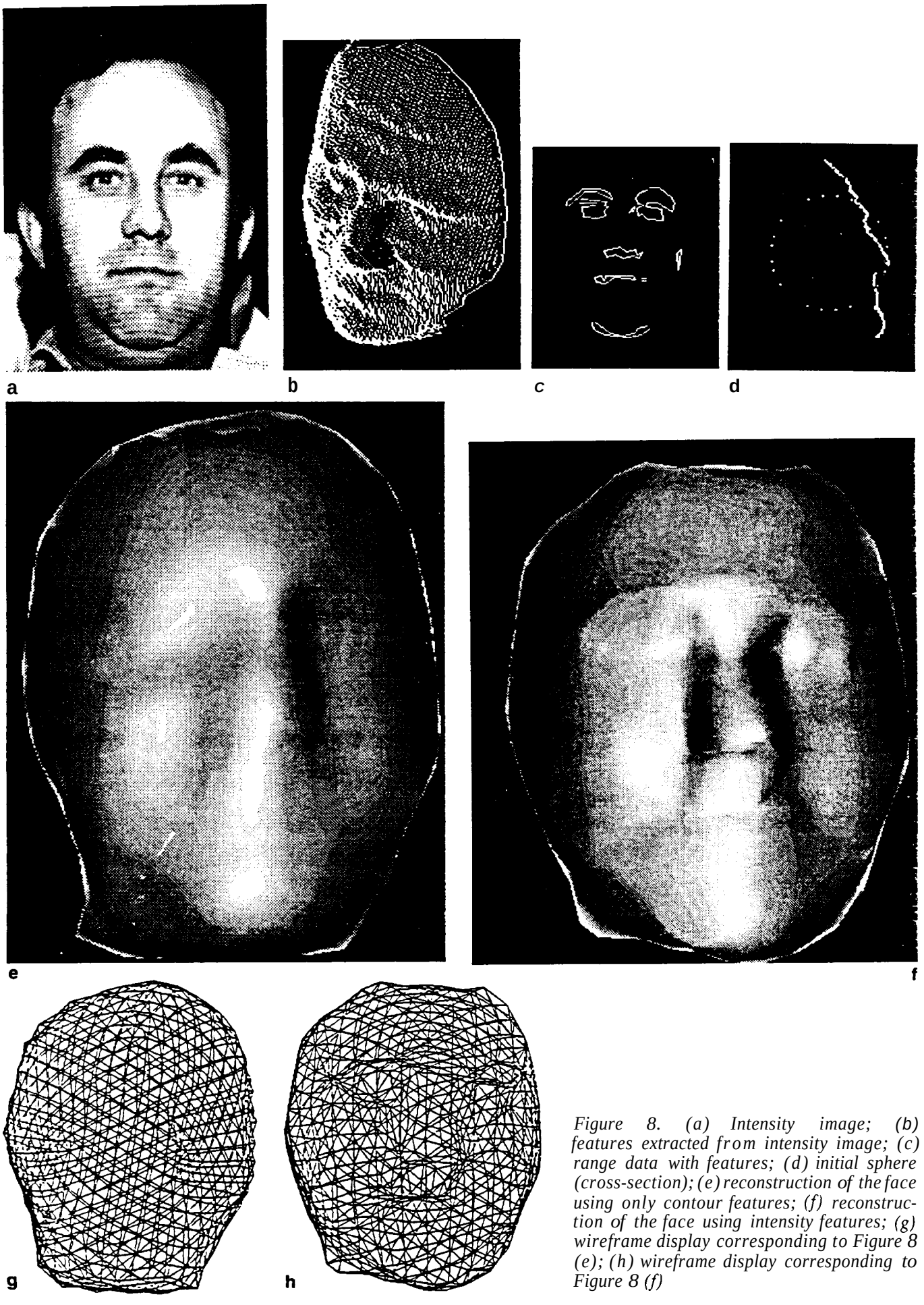


Figure 8. (a) Intensity image; (b) features extracted from intensity image; (c) range data with features; (d) initial sphere (cross-section); (e) reconstruction of the face using only contour features; (f) reconstruction of the face using intensity features; (g) wireframe display corresponding to Figure 8 (e); (h) wireframe display corresponding to Figure 8 (f)

where l is the number of data points and m the number on nodes. In practice, some precomputations and assumptions allow us to improve this computational time greatly. The algorithm is otherwise linear in the number of features and the number of iterations. Due to its highly parallelizable nature, substantial speed-ups can be achieved.

Several parameters must be set to apply the motion equation. The parameter settings in the current implementation are:

- ***Radius and centre of initial sphere:*** the determination of centre and radius of the initial sphere depend on the nature of the data. The initialization procedure is described in the next section for each type of data.
- ***Number of iterations,*** T_0: because T_0 is explicitly used for the computation of the feature force, it has an influence on the recovery of the object. The larger T_0, and the smaller the detormation due to the feature being between two iterations, the smoother the final shape is.
- ***Smoothness coefficient*** α_1 ***and*** α_2: the smoothness coefficients should be between 0 and 1 with $\alpha_1 > \alpha_2$. Actual 'good' values have to be determined empirically. We use $\alpha_1 = 0.75$ and $\alpha_2 = 0.4$ in our experiments.
- ***Damping factor k:*** the damping factor should be close to (but lower than) 1 to ensure smooth deformation of the surface over time and to avoid oscillations. We use $k = 0.9$.
- ***Normalizing factor*** K: K depends on the environment. It is computed as one-fifth of the radius of the largest object expected in a scene for a given application. The algorithm is robust with respect to K so that a rough estimate of object size is sufficient.

EXPERIMENTAL RESULTS

In this section we present experimental results obtained by applying the deformable surface algorithm to real image data. We present two sets of results. The first set is obtained using range data from a light-stripe range finder. The second set of experiments involves the recovery of three-dimensional objects from intensity images. The goal of those experiments is to validate the claim that the algorithm is independent of the nature of the data.

Range data

For our experiments with range data, we use a commercial light-stripe range-finder that consists of a camera and a projector that projects patterns through a **LCD** board". The sensor processes the images of the patterns using standard light-striping goemetry. It delivers a set of four images of 240 rows by *256* columns, one intensity image and three images of the three coordinates of every pixel with respect to a reference frame. Several sensors can be used at once to yield multiple views of a scene. **A** calibration procedure is used to express all data points coordinates with respect to a single world-centred frame. We conducted the experiments with either single views **or** multiple views. Using multiple views demonstrates that the algorithm is completely independent of an image-centred reference frame. In particular, it does not use the grid structure of the image or the uniform sampling of pixels in the image, and therefore can be used with non-imaging sensors that measure non-uniform sparse data. This is a major difference with other reconstruction algorithms that use the image grid as a discretization of the surface.

We analyse the experiments in a simple to complex fashion. Our first experiment addresses the case of a single image of an isolated smooth object, a human face, which includes small local shape features (nose, lips, . . .). The second experiment uses a natural environment made of sand and pebbles, and demonstrates the ability to deal with data from multiple sensor, and to handle arbitrary reference frames. The third experiment demonstrates how the algorithm can be used to segment a scene into individual object models.

Human face

A human face is a good example of a complex object with parts of very distinct nature: the forehead and jaws areas are of little interest for face recognition, whereas eyes, nose, mouth and chin are the main characteristics of a face. We would like a surface reconstruction algorithm to smooth the input data and generate a compact representation of the face while keeping an accurate description of the areas of interest. Figure 8a shows the intensity image, and Figure 8b shows the range data of a face; the data corresponding to the hair of the person is removed because the sensor generates extremely noisy measurements there. Only a partial view of the face is available because of self-occlusions. For example, the left part of the nose is not visible. In a first experiment, we consider as features only the boundary of the face. Using this minimum information, the surface is initialized as a sphere roughly tangent to the face (see Figure 8d). The resulting shape is displayed in two manners: Figure 8g is the triangular mesh of points, while Figure 8e is a shaded display obtained from ray-tracing. While the overall shape of the face was found, important facial features such as nose, mouth and eyebrow were smoothed.

To avoid this smoothing effect it is necessary to tag the eye, nose and mouth as being important features. The thresholding of the magnitude of an edge detector on the intensity image provides a way to extract those features (see Figure 8c). The result of the final recovery is shown in Figure 8f and h: this time the nose, eyebrow and mouth are clearly visible and the left part of the nose was interpolated. This experiment clearly demonstrates how fine details as well as general shape can be recovered due to the combination of feature and data forces.

Stacked pebbles

The generality of our approach makes it particularly well-suited for unstructured environments in which there are few constraints on object shapes. The scene we use in this example consist in a set of stone pebbles that are lying on top of sand. In order to have more reliable data and features, we took three range images from three different viewpoints (see Figure 9a). The features are of three natures: distance discontinuities, surface orientation discontinuities and shadow bound-

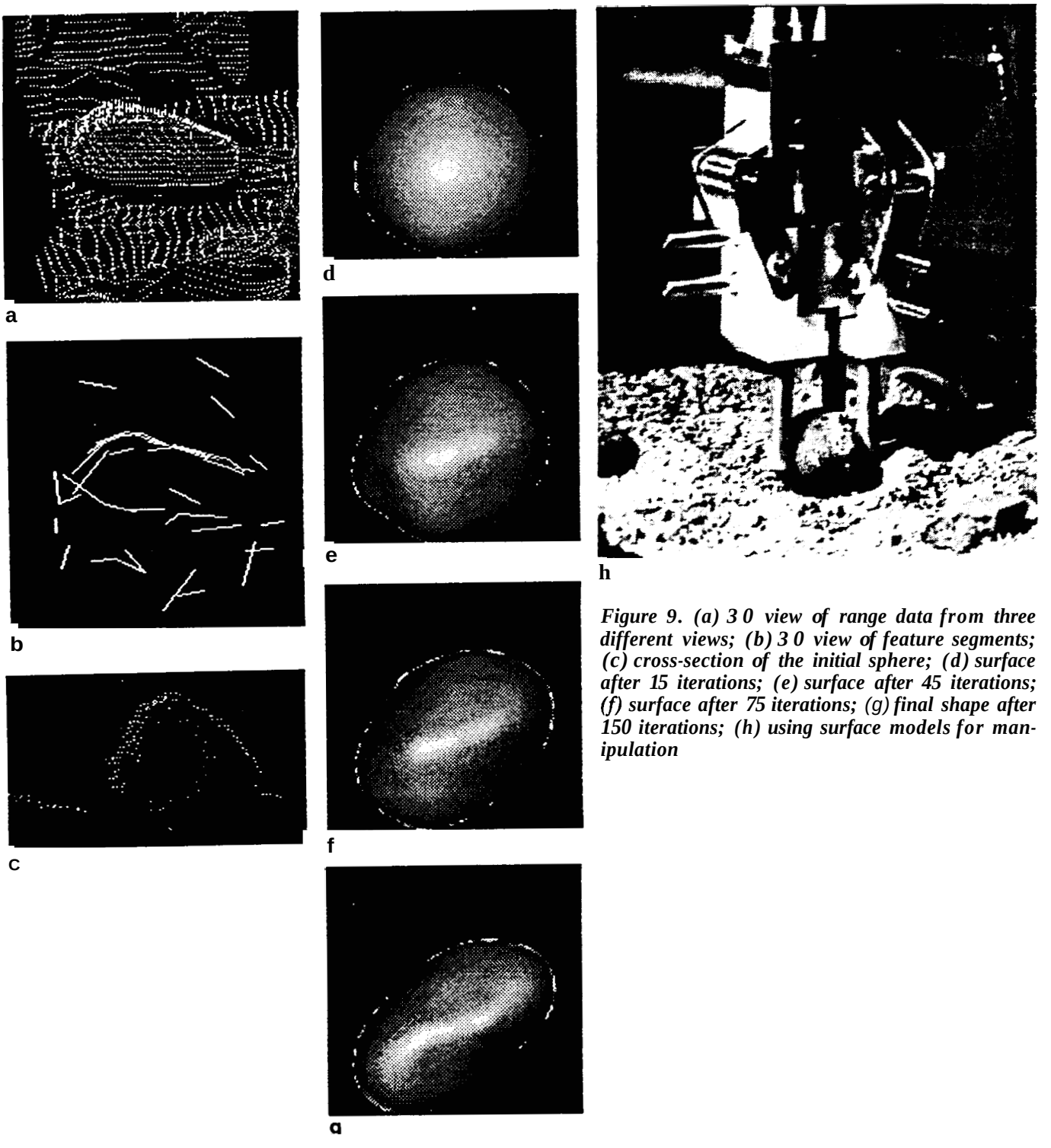


Figure 9. (a) 3 0 view of range data from three different views; (b) 3 0 view of feature segments; (c) cross-section of the initial sphere; (d) surface after 15 iterations; (e) surface after 45 iterations; (f) surface after 75 iterations; (g) final shape after 150 iterations; (h) using surface models for manipulation

aries. Shadows are parts of the scene that are visible from the camera but are not illuminated by the projector. Shadows are used here as clues for the presence of an object since a shadow is created by an object occluding the light coming from the projector. The feature segments are shown in Figure 9b. The figure demonstrates that using classical feature-based segmentation techniques would be a very challenging task.

The segmentation proceeds by selecting a shadow region, starting with the largest in the image. Using the geometry of the sensors as computed from an off-line calibration procedure, a point that is assumed to be inside the object is selected based on the position of the shadow in the image. The centre of the initial sphere is initialized at this point, its radius is initialized to a constant value that is the average size of the expected objects. Figure 9c displays the initial sphere. Only data points and features that are less than a fixed distance away from the starting point are taken into account in the computation of the forces. Figures 9d–g show the shape as it evolves from the initial sphere to the final model. By merging multiple views of the scene, we were able to faithfully extract a model of the pebble without any ***a priori*** segmentation.

This application to the segmentation of natural

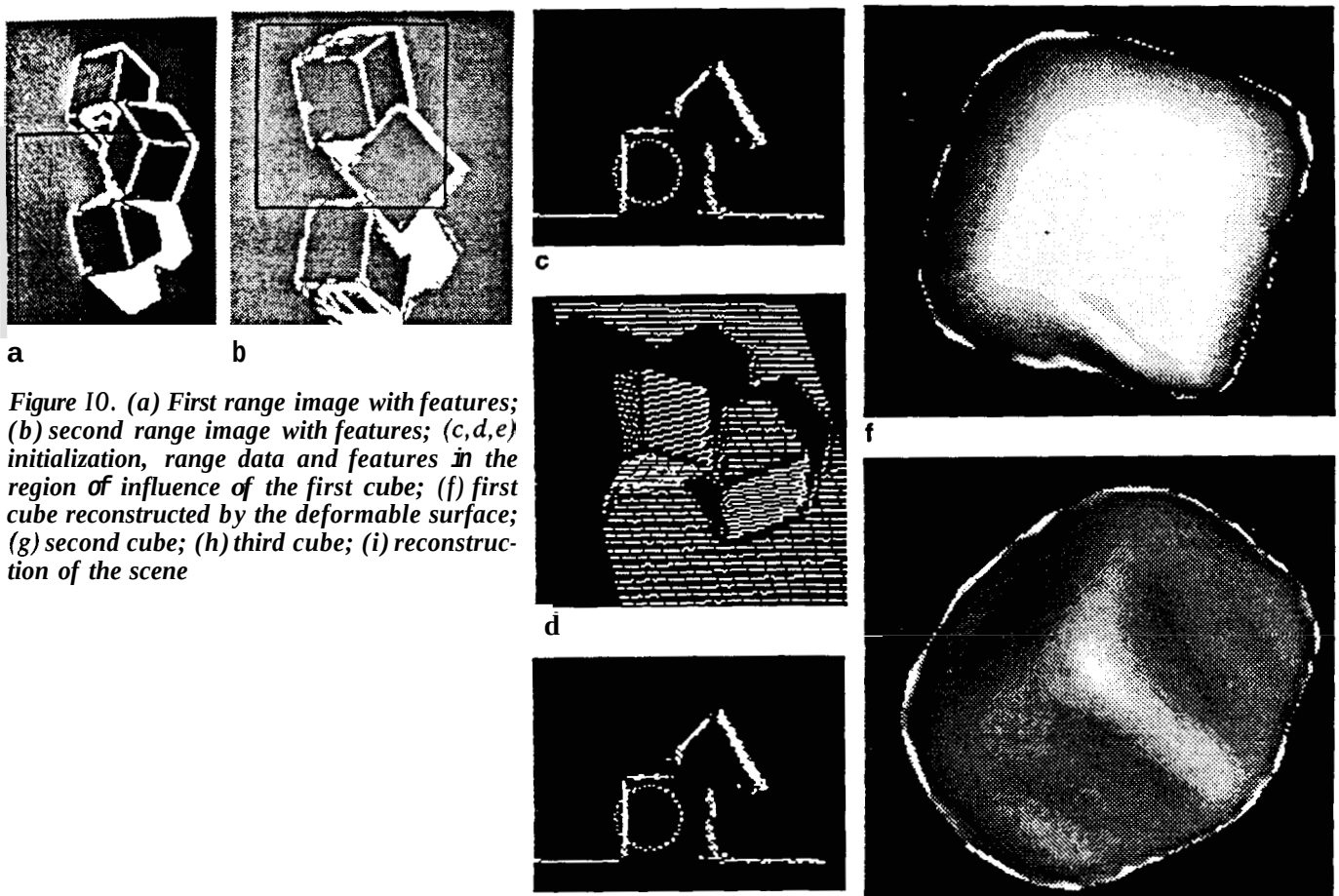


Figure 10. (a) First range image with features; (b) second range image with features; (c,d,e) initialization, range data and features in the region of influence of the first cube; (f) first cube reconstructed by the deformable surface; (g) second cube; (h) third cube; (i) reconstruction of the scene

scenes is part of an operational system that automatically picks up pebbles in a cluttered natural environment[19]. The discrete surfaces extracted from range images are fed to a program that computes the stable grasp positions for a three-finger gripper[20]. A manipulator executes the grasping. Figure 9h shows the gripper picking up a pebble in a typical environment.

Stacked cubes

To investigate further the ability of the algorithm to perform segmentation and surface reconstruction, we designed a simple scene with three overlapping cubes. Using simple objects of known shapes such as cubes allows us to compare the resulting segmentation and reconstruction with the actual scene more easily. Figures 10a and b show two opposite views of a scene where one cube is in balance between two adjacent cubes; the squares drawn corresponds to the region used for the recovery of the first cube. The range data and features that are enclosed in that region are displayed in Figures 10d and e, as well as the initial position of sphere (see Figure 10c). The model extraction is applied successively to the three cubes (Figures 10f–h), and a reconstruction of the scene is displayed in Figure 10i. The algorithm is clearly able to automatically discard features and data that are not consistent with the surface. Among the three reconstructed shapes, the second seems to be the best (see Figure 10g), mostly because it is not occluded by any other cube. On the contrary, the third cube (see Figure 10h) has one face that is poorly recovered because there are

no features and data points corresponding to this face of the cube. Therefore all visible parts were correctly extracted.

Intensity images

The next results concern the very under-constraint problem of extracting a three dimensional shape information from a contour of an intensity image. Terzopoulos[9] proved that the use of a free form representation is particularly well-suited for this task, since only weak assumptions are required to extract models. Given a rough axis of symmetry for each contour, he was able to infer the three dimensional shape of the object by propagating the symmetry constraint through the surface. Our scheme is more general because no axis of symmetry is needed so that objects that are non-symmetric and non-convex can be recovered. If we denote by *(xy)* the plane of the intensity image, and by z the axis perpendicular to the image, then we first initialize the model as a superquadrics, and we apply deformations from data and features such that these deformations occur only in the *(xy)* plane. In other words, every point of the surface keeps its coordinate z constant during the deformation. Therefore the choice of the initial surface sets the constraints on the model such as symmetry, elongation, and roundness.

We show an example of such reconstruction by using an intensity image of a tape dispenser (see Figure 11a). From this intensity information, we extract the contour by thresholding the image magnitude of an edge detector. The contour is then used both as a set of feature segments that will globally deform the surface, and as set of data points that will locally deform the surface (see Figure 11b). We display the initial position of the surface (see Figure 11c) which can be far from the object. As initial shape, we choose three different types of superquadrics. Superquadrics are generalized ellipsoids whose shapes are controlled by two parameters ε_1 and ε_2. The equation of the superquadrics is:

$$\left[\left[\frac{x}{a_1}\right]^{2/\varepsilon_2} + \left[\frac{y}{a_2}\right]^{2/\varepsilon_2}\right]^{\varepsilon_2/\varepsilon_1} + \left[\frac{z}{a_3}\right]^{2/\varepsilon_1} = 1$$

By setting $\varepsilon_2 = 1$, $a_1 = a_2$ and by choosing $\varepsilon_1 =$ 0.5, 1.0 and 2.0 we get the shapes shown in Figure 13. $= 0\,S$ is well suited for the reconstruction of man-made objects, with sharp edges, while $\&] = 1.0$, a sphere, is better for symmetrical objects and $\varepsilon_1 = 2.0$ can be used for elongated objects. In fact, $\varepsilon_1 = 1.0$ is a good approximation of the symmetry-seeking shapes. To reconstruct the tape dispenser with a realistic height in comparison with its longitudinal dimensions, we chose $a_3 = 0.2\,a_1$. Figures 11d and e show an intermediate step and the final convergence of the surface. Figures 12a-c present the three reconstructed objects. The side views allows us to see how the reconstructed surface is influenced by the initial profile $z = f(x, y)$.

For each contour an infinite number of models can be extracted by choosing a different initial surface. An important feature of this algorithm is that during the deformation, it conserves the symmetry with respect the image plane *(xy)* or with respect an axis in the image plane. By choosing an initial shape symmetric with respect to *(xy)* the resulting model is guaranteed to satisfy the same property. A more general scheme

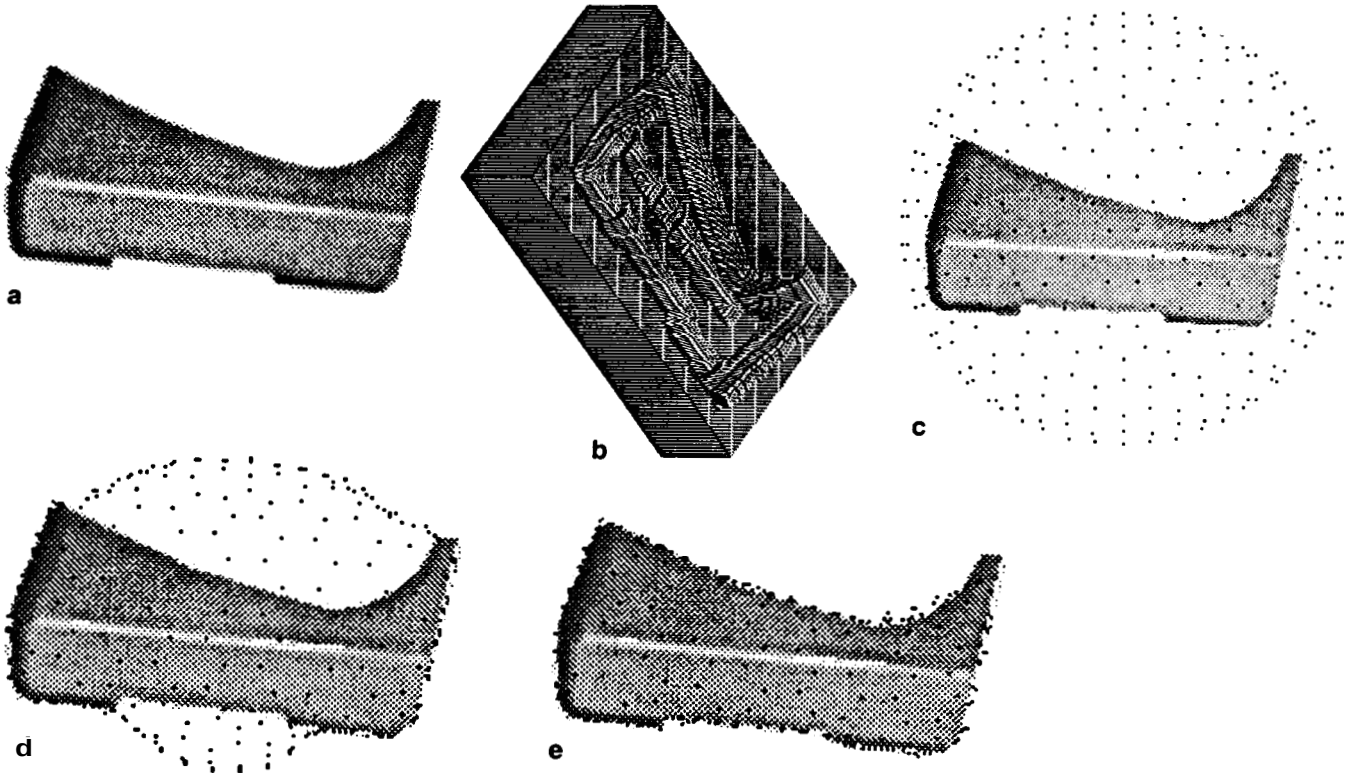


Figure 11. (a) Intensity image of the tape dispenser; (b) potential field created by the edges; (c) initialization; (d) intermediate position; (e) final position

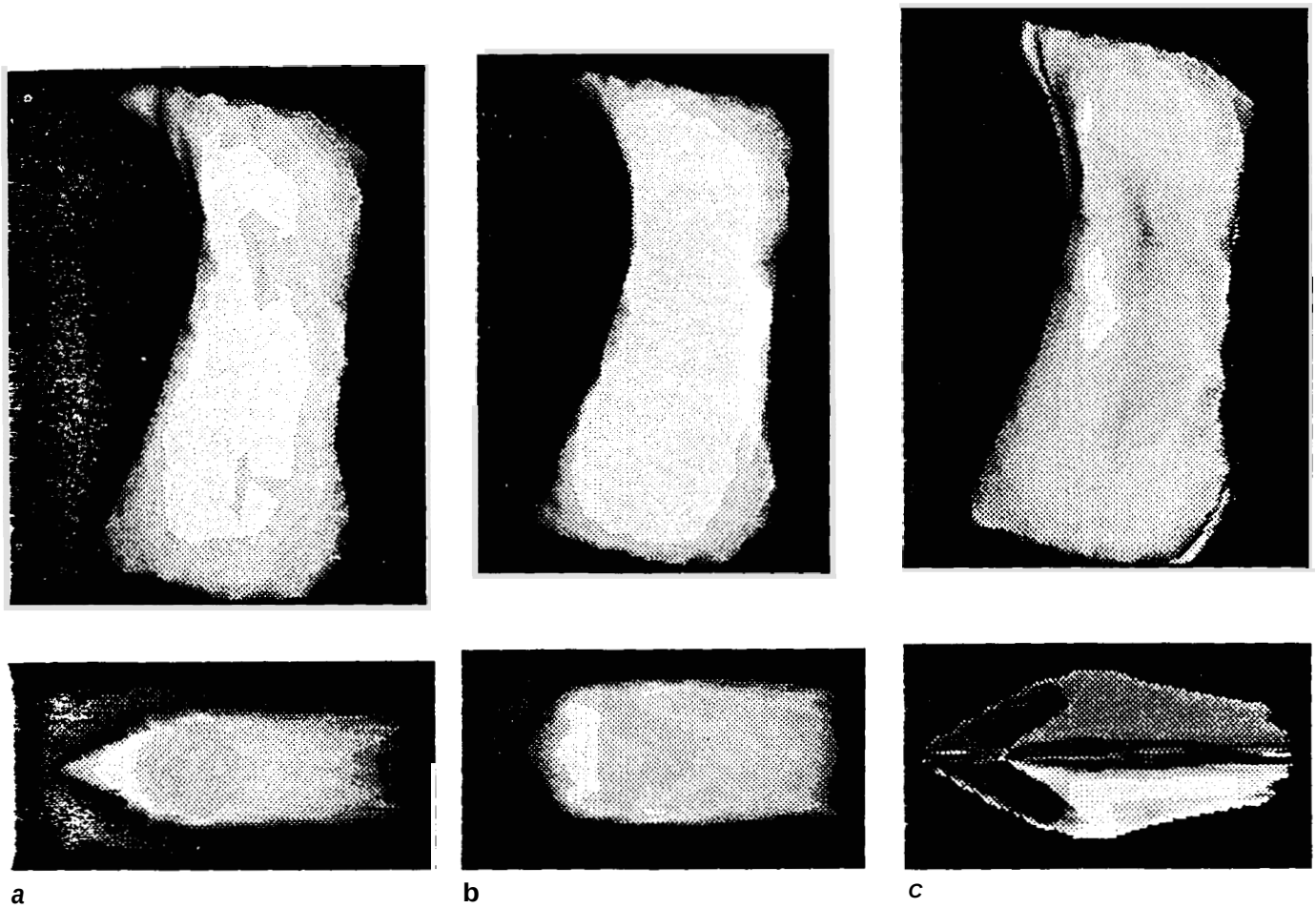


Figure 12. (a) Front and side view and of the object using a profile with $\varepsilon_1 = 1$; (b) front and side view and of the object using a profile with $\varepsilon_1 = 0.5$; (c) front and side view and of the object using a profile with $\varepsilon_1 = 2$

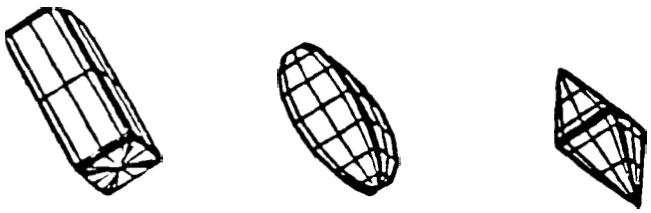


Figure 13. Different types of superquadrics that are used for the initialization of the surface

would be to add other geometric constraints such as the projected contour of the object from different view-Joints or surface normals obtained from shape from shading.

CONCLUSION

We have designed and implemented an algorithm that constructs 3D shape representations from observed data. Initial experiments show that we are able to represent a large class of objects from input data with few assumptions on the nature of the data and without requiring perfect segmentation as an input. Based on the experiments, the algorithm has the following characteristics:

- ***Physically-based algorithm:*** the dynamics of the deformation is modelled by the Lagrangian equations of mechanical systems. As demonstrated in previous works, using physically based procedures provides better control on the stability and convergence of the algorithm.
- ***Enhanced shape description:*** our algorithm enables us to describe hierarchically the input data between two classes, data points and features, according to their influence on the overall shape of the object. Our algorithm creates models that respect the same hierarchy since regions corresponding to feature points are better described than region of data points.
- ***Stability:*** because the algorithm uses both features and data, it is less sensitive to spurious features, noisy data or missing data. Moreover, an approximate partition of data and features into regions of interest is sufficient to extract the shape of an object. Therefore, our algorithm performs a segmentation by discarding the features and data that is incompatible with the current shape of the surface. This is in sharp contrast with other techniques that require the observed scene to be already segmented into regions corresponding exactly to individual objects.
- ***Generality:*** the algorithm makes few assumptions on data and observed objects. The only requirement is that some features can be extracted from input data, and that the minimum and maximum sizes of the object expected in a typical scene are known. We have successfully applied the algor-

ithm to the reconstruction of smooth isolated objects such as human faces, to the reconstruction of structured objects such as polyhedra, and to the segmentation of complex scenes with mutually occluding objects. We have tested the algorithm using data from different sensors including grey-coding and laser range finders and video cameras, using one or several images.

The algorithm has some limitations as demonstrated in the experiments. First, it tends to smooth the sharpest features such as the corners of a polyhedron. The smoothing of surface normal discontinuities is a well known problem in surface reconstruction techniques". This can be avoided by decreasing the smoothness parameters α_1 in the vicinity of such a discontinuity. Second, it generates a surface even where no data is available by interpolating from regions where data is available. A mechanism should be added to attach a degree of confidence on each part of the surface, this number being low when the part was interpolated and high when the model is very close to the data. This number can further be used as uncertainty measure for high level modules such as recognition.

We intend to extend the current applications by using additional data such as surface from shape from shading, or sparse 3D points from stereo, and by using additional feature types such as point features, planar patches, and corners. Future work concentrates on the use of the techniques presented in this paper in comprehensive robotics systems. We have already used deformable surfaces in a system for manipulating objects in natural environments using a three-finger gripper. We are currently using those techniques in a landmark mapping and recognition system for autonomous navigation.

ACKNOWLEDGEMENTS

This research was supported in part by NASA under Grant NAGW 1175, and in part by DARPA through ARPA order No. 4976. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of NASA, DARPA, or the US government.

REFERENCES

1 **Besl, P and Jain, R** 'Segmentation through symbolic surface descriptions', ***Proc CVPR,*** Miami, FL (1981) pp 77–85

2 **Pentland, A and Sclaroff, S** ***From Features to Solids*** MIT Media Lab Tech. Rep. No 135 (1990)

3 **Marr, D and Nishihara, K** 'Representation and recognition of the spatial organization of three-dimensional shapes', ***Proc. Roy. Soc. London,*** Vol 200 (1978) pp 269–294

4 **Brooks, R** 'Symbolic reasoning among 3D models and 2D images', ***Artif. Intell.***, Vol 17 (1981) pp 285–348

5 **Pentland, A** 'Recognition by parts', ***Proc. ICCV*** London, UK (1987) pp 612–620

6 **Pentland, A** ***Perceptual Organization and the Representation of Natural Form,*** SRI International Tech. Rep. no 357 (1986)

7 **Yokoya, N, Kaneta, M and Yamamoto, K** ***Recovery of Superquadrics Primitives from Range images by Simulated Annealing,*** ETL Tech. Rep. TR-90-18 (1990)

8 **Kass, M, Witkin, A and Terzopoulos, D** 'Snakes: active contour models', ***Int. J. Comput. Vision,*** Vol 1 No 4 (1988) pp 321–331

9 **Terzopoulos, D, Witkin, A and Kass, M** 'Symmetry-seeking models for 3D object reconstruction', ***Int. J. Comput. Vision,*** Vol 1 No 3 (October 1987) pp 211–221

10 **Terzopoulos, D and Metaxas, D** 'Dynamic 3D models with local and global deformations: deformable superquadrics', ***Proc. ICCV,*** Tokyo, Japan (1990) pp 606–615

11 **Bolle, R and Vemuri, B** 'On three-dimensional surface reconstruction methods', ***IEEE Trans. PAMI***, Vol 13 No 1 (January 1991)

12 **Terzopoulos, D** ***Computing Visible-Surface Representations,*** MIT Art. Int. Memo 800 (1985)

13 **Berter, M, Poggio, T and Torre, V** ***111-Posed Problems in Early Vision,*** MIT AI, Memo 924 (1987)

14 **Ikeuchi, K and Horn, B** 'Numerical shape from shading and occluding boundaries', ***Artif. Intell.*** Vol 17 (1981) pp 141–184

15 **Zucker, S** 'The organization of curve detection: coarse tangent fields and fine spline coverings', ***Proc. ICCV,*** Tampa, FL (1988) pp 568–577

16 **Fox, C** ***An Introduction to the Calculus of Variations,*** Dover Publications Inc. (1963)

17 **Meinguet, J** 'Multivariate interpolation at arbitrary points made simple', ***J. Applied Math. and Physics,*** Vol 30, pp 292–304

18 **Sato, K and Inokuchi, S** 'Range-imaging system utilizing nematic liquid crystal mask', ***Proc. ICCV,*** London, UK (1987) pp 657–661

19 **Choi, T, Delingette, H, Hebert, M and Ikeuchi, K** 'A perception and manipulation system for collecting rock samples', ***Proc. SOAR,*** Alberquerque, USA (1990)

20 **Francois, C, Ikeuchi, K and Hebert, M** 'A three-finger gripper for manipulation in unstructured environments', ***Proc. IEEE Conf. on Robotics and Automation,*** Sacramento, USA (1991)

21 **Bolles, B and Bobick, AF** 'Representation space: an approach to the integration of visual information', ***Proc. Image Understanding Workshop,*** Palo Alto, USA (1989)

Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 467-472, Lahaina, Maui, Hawaii, June 3-6, 1991.

Shape Representation and Image Segmentation Using Deformable Surfaces[1]

H. Delingette, M. Hebert, K. Ikeuchi
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA 15213

Abstract

We present a technique for constructing shape representatinn from images using free-form deformable surfaces. We madei an object as a closed surface that is deformed subject to attractive fieldsgenerated by input data points and features. Features affect the global shape of the surface while data points control its local shape. Our approach is used to segment objects even in cluttered or unstructured environment. The algorithm is general in that it makes few assumptions on the type of features, the nature of the data and the type of objects. Wepresent results in a wide range of applications: reconstruction of smooth isolated objects such as h u mfaces, reconstruction of structured objects such as polyhedra, and segmentation of complex scenes with mutually occluding objects. We have successfully tested the algorithm wing &a from different sensors including grey-coding range finders and video cameras, using one or several images.

1 Introduction

The recovery of object shape from 3D data is one of the key issues in vision. One could define this task as the segmentation of a large set of data points into shapes corresponding to objects in the scene. The shape representation should be general enough to handle a wide variety of scenes yet simple enough to be usable for other tasks such as recognition and manipulation. In other words. the shape representation should have enough parameters to describe the specificity of the shape but must have as few parameters as possible to be usable and to be robustly extracted from visual data. This contradiction is similar to the scale space problem, where one would like to find a description fine enough to capture the key details of the shape, but coarse enough to get rid of spurious data.

In this paper, we propose an approach that attempts to solve this conflict by using the feature / data duality in a way similar to the fine / coarse approach. Several psychophysical experiments have proved that the human eye is able to capture the main shape of an object by seeing only a few characteristic elements or features. These features can be either geometric (distance discontinuities, surface orientation discontinuities, comers, minimum of curvature, etc.) or higher level such as reflectance properties. While these features capture most of the shape information, it is difficult without a priori knowledge to construct a full reconstruction of the object. however.

Several solutions have been proposed. Besl and Jain[2] built curvature-based object representations by classifying surfaces according to the sign of its principal curvatures. Pentland[15] presented a physically-based algorithm. to recover in a unique manner a model from a set of features and a set of vibration modes. Another approach is to use both features and range data in separate stages. In a first stage, features are grouped into hierarchical sets, according to geometric properties (symmetry, connexity, etc.) and in the second stage models are fit to the segmented parts. This idea of hierarchical representation was initiated by Marr and Nishihara[11] and pushed further by the seminal ACRONYM[4] vision system by using generalized cylinders. Pentland[13][14]'s "Representationby parts" using deformed superquadrics, proved to have some successful results but encounters some limitations[6]. While the feature grouping requires some accurate feature extraction and high level reasoning, the fitting of superquadrics[13][21] to range data has some unstable behavior. due to its non-linear nature, and is suitable for only smooth and simple shapes.

Those techniques attempt to represent all shapes by using a set of elementary shapes (superquadrics, generalized cylinders, parametric patches, etc.) that can be described by a few parameters. This is clearly beneficial from the point of view of object recognition which amounts to manipulating analytical equations of the elementary shapes. In practice, however, it restricts considerably the class of objects and scenes to which the techniques can be applied. More general representations could be obtained by adding degrees of freedom to the elementary shapes (e.g., adding tapering and bending to superquadrics). However, the non-linear fitting algorithms involved in the recovery of such shapes become rapidly computationally expensive and numerically unstable. Furthermore, most of those techniques assume that the observed scene is first segmented in regions corresponding to individual objects. The shape extraction algorithms are then applied to each object. However, accurately segmenting a scene is a hard problem in itself.

To address those problems, Terzopoulos and Witkin[10][17][18][20] proposed the concept of deformable contours or surfaces that are subject to forces generated by image elements such as edgels. This work demonstrated that free-form shapes are powerful tools for shape representation. However, to perform

1. This research was supported by NASA under Grant NAGW-1175 and by DARPA through ARPA Number 4976 monitored by the air Force Avionics Laboratory under contract F33615-87-C-1499. The views and conclusions contained in this document are those of the authors and shmld nor be interpreted as representing the policies of NASA, DARPA. or the US government.