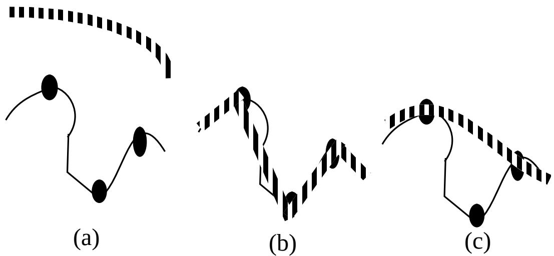

Figure 1: **a,** A contour of an object with its 3 features; **b,** Final deformation with feature attraction; **c**, Final deformation with data deformation.

(a) (b)

Figure 2 : **a,** *U(x)*; **b,** $E_{feature} = f (t, D_{feature})$

(a) (b)

Figure 3 : **a,** *dU / dx* (*x*); **b,** $F_{feature} = f (t, D_{feature})$

(a) (b)

Figure 4: **a,** *W (x)*; **b,** *dW / dx (x*)

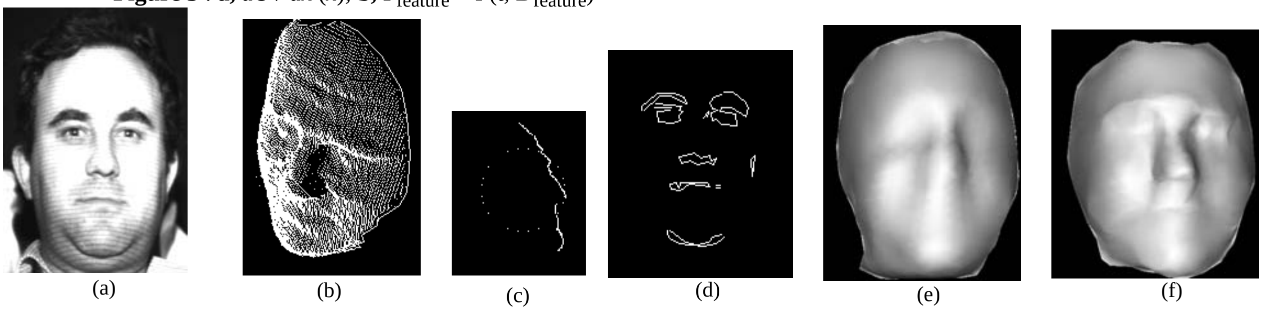

Figure 5: **a,** Intensity image; **b,** Corresponding range data; **c,** Initial sphere; **d,** Facial features; **e,** Reconstruction of the face using only contour features; **f,** Reconstruction of the face with facial features.

(g) (h)

Figure 5: **g,** Triangulated surface corresponding to (e); **h,** Triangulated surface corresponding to (f)

(a) (b) (c)

Figure 6: a, Three dimensional view of range data from three different views; **b,** Feature segments; **c,** Reconstructed pebble.

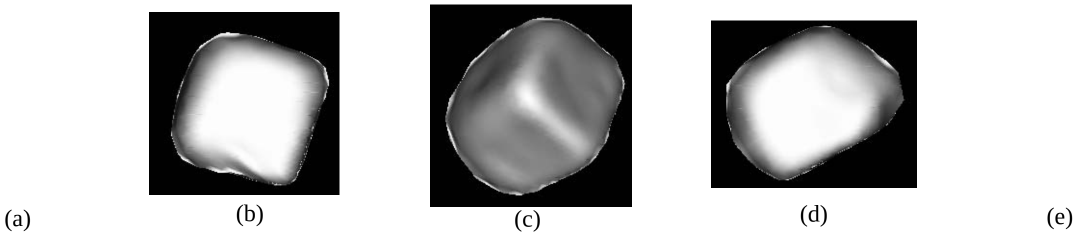

Figure 7: **a,** Scene picture; **b,** First reconstructed cube; **c,** Second reconstructed cube; **d,** Third reconstructed cube; **e,** Reconstruction of the scene

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 considering using this technique in other systems as well. One candidate is the segmentation of outdoor scenes and the representation of landmarks using a long-range laser range finder[3].

References

1 M. Berter, T. Poggio and V. Torre, "*Ill-Posed Problems in Early Vision*", M.I.T Art. Int. Memo 924, 1987

2 P. Besl and R. Jain, "*Segmentation Through Symbolic Surface Descriptions*", Proc. CVPR Miami, pp 77-85, 1986.

3M. Hebert, "*Building Object and Terrain Representations for an Autonomous Vehicle*", Proc. American Control Conference, June 1991.

4 R.Brooks, "*Symbolic Reasoning Among 3D Models and 2D Images*". Artificial Intelligence Journal, 17:285-348, 1981.

5 T. Choi, H. Delingette, M. Hebert and K. Ikeuchi, "*A Perception and Manipulation System for Collecting Rock Samples*", SOAR,1990.

6 F. Ferrie, J. Lagarde and P. Whaite "*Darboux Frames, Snakes and Superquadrics: Geometry from the Bottom-Up*", IEEE Workshop on Interpretation of 3D Scenes, Austin, 1989.

7 C. Fox, "*An Introduction to the Calculus of Variations*", Dover Publications Inc., 1963.

8 C. Francois, K. Ikeuchi and M. Hebert, "*A Three-Finger Gripper for Manipulation in Unstructured Environments*", 1990.

9 K. Ikeuchi and B. Horn, "*Numerical Shape from Shading and Occluding Boundaries*", Art. Int. 17, pp141-184, 1981.

10 M. Kass, A. Witkin and D. Terzopoulos, "*Snakes: Active Contour Models*", Int. Journ. of Computer Vision, pp321-331,1988.

11 D. Marr and K. Nishihara, "*Representation and Recognition of the Spatial Organization of Three Dimensional Shapes*", Proc. Roy. Soc. London, 200: pp269-294,1978.

12 J. Meinguet, "*Multivariate interpolation at arbitrary points made simple*", J. Applied Math. and Physics , 30, 292-304

13 A. Pentland, "*Recognition by Parts*", Proc. ICCV London, pp 612-620, 1987.

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

15 A. Pentland and S. Sclaroff, "*From Features to Solids*", M.I.T Media Lab Tech. Rep. No 135, 1990.

16 K. Sato and S. Inokuchi, "*Range-Imaging System Utilizing Nematic Liquid Crystal Mask*", Proc. ICCV, pp 657-661, 1987

17 D. Terzopoulos, A. Witkin and M. Kass, "*Symmetry-Seeking Models for 3D Object Reconstruction*", Int. Journ. Comp. Vision, Vol 1, No 3 , pp211-221, Oct 1987.

18 D. Terzopoulos, "*Computing Visible-Surface Representations*", M.I.T Art. Int. Memo 800, 1985.

19 D. Terzopoulos, D. Metaxas, "*Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics*", Proc. ICCV Osaka, 606-615, 1990.

20 A. Witkin, M. Kass, D. Terzopoulos, "*Physically Based Modeling for Vision and Graphics*", Int. Journ. of Computer Vision, pp321-331, 1988.

21N.Yokoya, M. Kaneta and K. Yamamoto," *Recovery of Superquadrics Primitives from Range images by Simulated Annealing*", ETL Tech. Rep. TR-90-18, 1990.

22 S. Zucker, "*The Organization of Curve Detection: Coarse Tangent Fields and Fine Spline Coverings*", Proc. ICCV Tampa, pp 568-577,1988.

Stacked Pebbles

The generality of our approach makes it particularly well-suited for unstructured environment in which there are few constraints on object shapes. The scene we use in this example consists in a set of pebbles that are lying on top of sand, using three range images from three different viewpoints (Figure 6(a)). The features (Figure 6(b)) are of three natures: distance discontinuities, surface orientation discontinuities and shadow boundaries. Shadows are parts of the scene that are visible from the camera but are not illuminated by the projector. Final shaded display is shown in Figure 6(c).

This application to the segmentation of natural scenes is part of an operational system that automatically picks up pebbles in cluttered natural environment[5]. The discrete surface extracted from range images are fed to a program that computes the stable grasp positions for a three-finger gripper[8].

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. Figure 7(a) is a view from one of the two range finders observing the scene: one cube is in balance between two adjacent cubes. The algorithm was applied successively to each of the three cubes. For each segmentation, a region around a cube is selected (square in Figure 7(a)), features are extracted and the center of the initial sphere is set et the center of this region. Figure 7(b)(c)(d) represent the three recovered cubes and Figure 7(e) the final reconstruction of the scene.

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 (Figure 7(c)), mostly because it is not occluded by any other cube. On the opposite, the third cube (Figure 7(d)) has one face that is poorly recovered because there is no feature and no data corresponding to this face of the cube. Therefore all visible parts were correctly extracted.

4 Conclusion

We have designed and implemented an algorithm that constructs 3-D free-form deformable surfaces and is able to represent a large class of object 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.

- **Enhanced shape description:** Our algorithm enables to describe hierarchically the input data between two classes, data points and features, according to their influence on the overall shape of the object. This hierarchical description is embedded into the resulting model of the object.

- **Stability**: Because the algorithm uses both features and data, it is less sensitive to spurious features, noisy data missing data. Moreover, this stability enables to perform segmentation by discarding the features and data that is incompatible with the current shape of the surface.

- **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. This allows us to deal with non-imaging sensors and merged multiple views.

- **Radius and center of initial sphere**: The determination of center and radius of the initial sphere depends on the nature of the data. The radius is related to the expected size of the object and the center position to the set of extracted features and a priori knowledge of the scene.

- **Number of iterations T_0:** Because T_o is explicitly used for the computation of the feature force, it has an influence on the recovery of the object. The largest T_0 is and the smallest the deformation due to the feature is between two iterations and therefore 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 α_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.

3 Experimental Results

For our experiments we use a commercial light-stripe range-finder that consists of a camera and a projector that projects patterns commanded by a LCD board[16]. Several sensors can be used at once to yield multiple views of a scene and thus demonstrating that the algorithm is completely independent of an image-centered reference frame.

We first discuss an experiment with an isolated object: a human face. Then we use an unstructured environment, a pebble lying on the sand, finally a cluttered environment with 3 overlapping cubes.

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. Figure5(a)(b) show the intensity image and the corresponding range data of a human face. In a first experiment, we consider as features only the segments that surround the face (Figure 5(b)). We initialize the surface as a sphere so that it is roughly tangent to the face (Figure 5(c)). The resulting shape is displayed in two manners: Figure 5(g) is the triangular mesh of points while Figure5(e) 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 (Figure 5(d)). The result of the final recovery is shown in Figure 5(f)(h): this time the nose, eyebrow and mouth are clearly visible and the left part of the nose was interpolated. This clearly demonstrates how the use of both features and data points enables a hierarchical description of the object.

2.4 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:

$$E_{data} = W\left(\frac{\|\vec{P}C_{data}\|}{K}\right)$$

W(x) is a function that is quadratic if (x<1) and in *1/r* if (x>1) (Figure 4(a)). *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:

$$\vec{F}_{data} = -\nabla E_{data} = \frac{\frac{dW}{dx}\left(\frac{\|PC_{data}\|}{K}\right)}{\|\vec{P}C_{data}\|K} \cdot \vec{P}C_{data}$$

where *dW/dx* is the derivative function of *W(x)*. Figure 4(b) shows the curve *dW/dx*. 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 negligeable.

2.5 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 tessellated icosahedron as a structure. Each face of the icosahedron is subdivided to yield arbitrary 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. We use the center of each triangle as a node, every node having therefore three neighbors. If we write$\vec{r}^{\,i}_{t}$ as the position of the node i at the time t, then the discrete version of the motion equation is:

$$\vec{r}^{\,i}_{t+1} = \vec{r}^{\,i}_{t} + (1-k)\cdot(\vec{r}^{\,i}_{t} - \vec{r}^{\,i}_{t-1}) + \vec{F}_{smoothness} + \vec{F}_{data} + \sum_{i=0}^{n} \vec{F}^{i}_{Feature}$$

The surface is initialized as a sphere at t=0 and is deformed by applying repeatedly the equation of motion at each node. Forces $\vec{F}_{data}$ and $\vec{F}^{i}_{feature}$ are computed at each node independently in a parallel manner. $\vec{F}_{smoothness}$ 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 $\vec{F}_{data}$. It is theoretically in O(lm) where l is the number of data points and m the number on nodes. In practice, by using the geometry of the sensor, we are able to improve greatly this computational time. 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:

Our solution is to change the relative influence of both types of deformations over time: initially, the surface is mostly subject to 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 $D_{\mathrm{ref}}(t)$at time *t*.

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

By setting $D_{\mathrm{ref}}(t)$= $D^i_0 \cdot (T_0 - t)$, as t increases, the surface has to get closer and closer to be attracted by the feature. Therefore the influence of the features becomes smaller during the deformation.

In order to avoid discontinuities, we use a formulation that acts smoothly when *D(Feature,P)* is closed to $D_{\mathrm{ref}}(t)$. In the current implementation, we model features as 3-D 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, $E^i_{feature}$ (Figure 5.b) is:

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

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 (Figure 2(a)). Therefore *U*(*x*) is C^1 continuous, leading to an expression of the force that is C^0 continuous.
- $l_i\|\vec{PF_i}\|$ represents the distance between the point *P(u,v)* of the surface and the feature segment of length l_{i} and 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 center of the surface at its initial position. If the surface is initialized as a sphere, it is the distance to the center of the sphere.

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 $E^i_{feature}$:

$$\vec{F}^i_{feature} = \frac{-l_i}{D_0 \cdot (T_0 - t) \cdot \|\vec{PF_i}\|} \cdot \frac{dU}{dx}\left(\frac{l_i\|\vec{PF_i}\|}{D^i{}_0 \cdot (T_0 - t)}\right) \cdot \vec{PF_i}$$

where *dU/dx* is the derivative of *U(x)* (Figure 3(a)).

Figure 2(b) and 3(b) show the variation of the energy and force with respect to the time and *D(Feature,P)*; there is no singularity when *t* is close to T_0. By continuity, we can set $E^i_{feature}(t{=}T_0)$ = cste and $\vec{F}^i_{feature}(t{=}T_0)$= 0.

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

2.2 Smoothness Energy

We use the bivariate generalized spline functionals of the first and second order as a smoothness measure[1218]. 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., $\vec{r}_u = \frac{\partial}{\partial u}\vec{r}(u, v, t)$), the energy is defined by:

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

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

The constants α_1 and α_2 controls the relative smoothness of the surface. Let Δ_{uv} denotes the Laplacian operator with respect to (u,v). The corresponding force is therefore:

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

2.3 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. 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:

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

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,...), 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 (Figure 1(b)). On the contrary, in 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 (Figure 1(c)).

tle or no data is available. This is similar to the regularization approach to surface reconstruction from sparse data[9,18]. 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[19,18,22] which characterizes the surface smoothness. The relative importance of external and internal forces is a trade-off between accuracy and smoothness.

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.

2.1.General equations of motion.

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

- **Smoothness energy $E_{\text{smoothness}}$**: 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 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[101720] 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[7], the surface reaches a stable equilibrium when the Lagrangian of the system of forces reaches a minimum. The Lagrangian of the system is: $\boldsymbol{L = T - D - E_{\text{smoothness}} - E_{\text{feature}} - E_{\text{data}}}$. A deformable surface is parametrized by two spatial coordinates (*u*,*v*) and by the time *t*. A surface point P has coordinates:

$$\vec{OP} = \vec{r}(u, v, t) = \begin{bmatrix} X(u, v, t) \\ Y(u, v, t) \\ Z(u, v, t) \end{bmatrix}$$

With these notations, using the calculus of variations on L(*u*,*v*,*t*), one gets [17]):

$$\mu \cdot \frac{\partial^2}{\partial t}\vec{r}(u, v, t) = -k \cdot \frac{\partial}{\partial t}\vec{r}(u, v, t) + \vec{F}_{smoothness} + \vec{F}_{feature} + \vec{F}_{data}$$

where μ is the mass density of the surface and k is the damping factor.

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 “Representation by 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 correctly, the deformable surface had to be initialized close to the line of symmetry and it did not address the problem of segmenting the input image. Terzopoulos and Metaxas[19] in a recent paper combined the elementary shape and the deformable surface approaches to gain a local/global representation.

Our approach is to represent shapes by deformable surfaces as well. 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 is robust enough so that it does not require a precise segmentation of the scene as an input.

Section 2 describes the theory of deformable surfaces. Section 3 describes the implementation and applications.

2 Deformable surfaces

The basic surface description is a free-form deformable surface subject to three types of forces, external, internal, and inertial forces that follow the laws of a dynamic mechanical system. The surface is therefore as well a function of time *t:* initialization corresponds to *t*=0 and final shape corresponds to t=T_0.

External forces are generated by input data points and input features. External forces must be carefully designed to obtain an optimal shape description that uses features to constrain global shape and data points to fine-tune local shape. Earlier work on surface reconstruction using splines required a complete mapping between data and surface points to be defined beforehand. This constraint restricts the geometry of the problem and requires the surface to be initially very close to the solution.

Internal forces ensures that the surfaces 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 lit-

Deformable surfaces : a free-form shape representation[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 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 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. 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.

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, 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 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

[1] 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, expressed or implied of those agencies.