A Spherical Representation for Recognition of Free-Form Surfaces

Martial Hebert, Katsushi Ikeuchi, Hervé Delingette

Abstract—**We introduce a new surface representation for recognizing curved objects. Our approach begins by representing an object by a discrete mesh of points built from range data or from a geometric model of the object. The mesh is computed from the data by deforming a standard shaped mesh, for example, an ellipsoid, until it fits the surface of the object. We define local regularity constraints that the mesh must satisfy. We then define a canonical mapping between the mesh describing the object and a standard spherical mesh. A surface curvature index that is pose-invariant is stored at every node of the mesh. We use this object representation for recognition by comparing the spherical model of a reference object with the model extracted from a new observed scene. We show how the similarity between reference model and observed data can be evaluated and we show how the pose of the reference object in the observed scene can be easily computed using this representation.**

We present results on real range images which show that this approach to modelling and recognizing 3D objects has three main advantages:

1) **First, it is applicable to complex curved surfaces that cannot be handled by conventional techniques.**
2) **Second, it reduces the recognition problem to the computation of similarity between spherical distributions; in particular, the recognition algorithm does not require any combinatorial search.**
3) **Finally, even though it is based on a spherical mapping, the approach can handle occlusions and partial views.**

Index Terms—**Object recognition, deformable surfaces, range data, pose registration, 3D modeling, surface models, free-form surfaces.**

I. Introduction

Recognition of curved objects is one of the key issues in computer vision. It is present not only in traditional applications such as industrial object recognition and face recognition, but also in emerging applications such as navigation and manipulation in natural environments. The key to building a practical recognition system is to define suitable object representations.

Traditionally, there are two ways to represent objects for recognition: local and global. Local methods attempt to represent objects as a set of primitives such as faces or edges. Most early local methods handle polyhedral objects and report effective and encouraging results. Representative systems include [9], [16], and [11]. Few systems can handle curved surfaces. Some systems include early work in which primitive surfaces enclosed by orientation discontinuity boundaries are extracted from range data [17]. Other systems determine primitive surfaces which satisfy planar or quadric equations [7]. Techniques based on differential geometry such as [3] segment range images using Gaussian curvatures.

The global methods assume one particular coordinate system attached to an object and represent the object as an implicit or parametric function in this coordinate system. The resulting representation is global in that the implicit function represents the entire shape of the object or of a large portion of the object. Generalized cylinder and superquadrics [18] are representative of this group. The extended Gaussian image (EGI) [10], [12] is another global representation for which recognition is performed by correlating spherical maps. This representation does require the extraction of primitive shapes from input data but cannot be used in the presence of occlusion.

Recently, new approaches have been developed based on the idea of fitting a bounded algebraic surface of fixed degree to a set of data points [19], [20]. In this case, recognition proceeds by comparing the polynomials describing observed and stored surfaces [8]. Although encouraging results have been obtained in this area, more research is needed in the areas of bounding constraints, convergence of surface fitting, and recognition before this approach becomes practical. Occlusion remains a problem since there is no guarantee that the polynomial computed from a partial view is similar to the polynomial computed from a complete model of the object.

All these approaches attempt to fit some known parametric surface, either locally or globally, to the object. Another class of approaches attempts to match sets of points directly without any prior surface fitting. An example is the work by Besl and Kay [2] in which the distance between point sets is computed and minimized to find the best transformation between model and scene. This approach does not require any surface segmentation or surface fitting. Recent results show that these algorithms can perform remarkably well by using numerical techniques for minimizing distances between two arbitrary point sets. The main drawback of this approach is that, like any minimization technique, it requires a initial guess of the transformation between model and scene.

In order to overcome the limitations of existing representations, we have designed a new approach that uses as a starting point a combination of several traditional object recognition and representation methods. Our approach begins with a combination of the point set matching and the original EGI approach. As in the case of the point set matching, we want to avoid fitting analytical surfaces to represent an object. Instead,

Manuscript received Jan. 21, 1993; revised April. 4, 1995; recommended for acceptance by L. Shapiro.

M. Hebert, and K. Ikeuchi are with the Robotics Institute, Carnegie Mellon University.

H. Delingette is with Project EPIDAURE, INRIA Sophia-Antipolis, France.

IEEECS Log Number P95089.

we use a representation that simply consists of a collection of points, or nodes, arranged in a mesh covering the entire surface of the object. This has the advantage that the object can have any arbitrary shape, as long as that shape is topologically equivalent to the sphere. To avoid problems with variable density of nodes on the mesh, we need to define regularity constraints that must be enforced when the mesh is built. Constructing meshes that fit input data and that satisfy regularity constraints is possible based on the optimization techniques originally introduced in [21] and [13]. We use an extension of the deformable surface algorithms introduced in [5] to compute the meshes.

As in the EGI algorithms, each node of the mesh is mapped onto a regular mesh on the unit sphere, and a quantity that reflects the local surface curvature at the node is stored at the corresponding node on the sphere. Instead of using a discrete approximation of the curvature, we develop a new measure of curvature, the simplex angle, which is entirely defined from a node and its neighbors in the mesh without any reference to the underlying continuous surface. We call the corresponding spherical representation the spherical attribute image (SAI).

To determine whether two objects are the same, we only need to compare the corresponding spherical distributions. The overall approach is illustrated in Fig. 1. A regular mesh is computed from input sensor data; a simplex angle is computed at each node of the meshes and the meshes are mapped onto a sphere, the SAI. A fundamental difference between the SAI and other global representations is that a unique mesh, up to rotation, translation, and scale, can be reconstructed from a given SAI. In the case of the EGI, for example, this property is true only for convex objects. Another fundamental difference is that the SAI preserves connectivity in that patches that are connected on the surface of the input object are still connected in the spherical representation. The latter is the main reason why our approach can handle arbitrary non-convex objects and in the presence of occlusion.

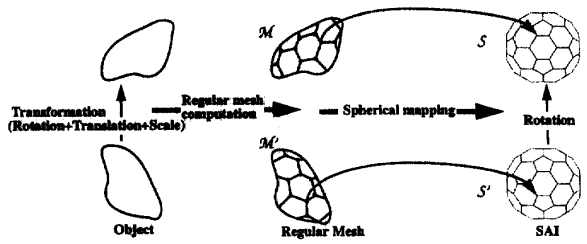


Fig. 1: Object recognition using SAIs.

The paper is organized as follows: In Section II, we describe a simple representation of closed 2D curves which we extend to 3D surfaces. In Section III, we show how to obtain SAIs from range data. In Section IV, we describe the SAI matching. Finally, we address the problem of occlusion and partial models and present several results of recognition in complex scenes in Section V.

II. Representing 3D Surfaces

In this section we extend the concepts of curvature indicator, local and global regularity, and circular representation to 3D surfaces. We consider the case of representing surfaces topologically equivalent to the sphere. We first develop the basic concepts by describing a simple representation for 2D curve and generalize those concepts to three dimensions. Detailed presentations of the basic results on semi-regular tessellations, triangulations, and duality used in this section can be found in [15], [22], and [23].

A. Representing 2D Curves

A standard approach to representing and recognizing contours is to approximate contours by polygons, and to compute a quantity that is related to the curvature of the underlying curve. The similarity between contours can then be evaluated by comparing the distribution of curvature measurement at the vertices of the polygonal representations.

The curvature of a discrete curve at each node of the polygonal approximation can be approximated by the angle φ between consecutive segments. The relation between φ and the curvature k is $k \approx \varphi/l$ as the density of points increases. Like the curvature, the angle φ is independent of rotation and translation. In addition, φ is independent of scale.

One problem is that if the lengths of the segments representing the curve are allowed to vary, the value of φ depends not only on the shape of the curve but also on the distribution of points on the curve. In particular, it is important for the same curve shape to generate the same value of φ to enable the comparison of discrete curves. One way to avoid this problem is to impose a local regularity condition on the distribution of vertices. The local regularity condition simply states that all the segments must have the same length. An equivalent definition is that two segments the projection of any node $\boldsymbol{P}$ onto the line joining its two neighbors $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ coincides with the midpoint of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$.

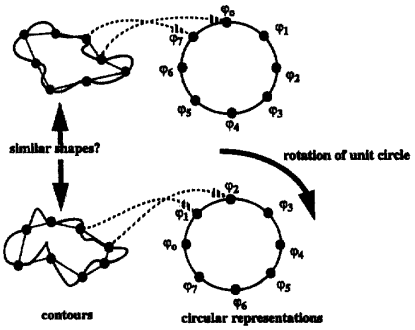


Fig. 2: Comparing contours in representation space.

The last step in representing 2D contours is to build a circular representation that can be used for recognizing contours. Let us assume that the contour is divided into N segments with vertices $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_N$, and with corresponding angles $\varphi_1, \ldots, \varphi_N$. Let us divide the unit circle using N equally spaced vertices

$C_1, \ldots, C_N$. Finally, let us store the angle φ_i associated with P_i at the corresponding circle point C_i. The circular representation of the contour is invariant by rotation, translation, and scaling. This property allows for comparing contours by deciding that two contours are identical if there exists a rotation of the unit circle that brings the contours representation in correspondence (Fig. 2). The unicity property is true because of the local regularity condition and because of the invariance of φ.

B. Mesh Topology and Regularity

The most natural discrete representation of a surface is a triangulation, that is, a polyhedron with triangular faces whose vertices are on the surface. Each face defines a plane which is the local approximation of the surface. It is desirable for many algorithms to have a constant number of neighbors at each node. Use a class of meshes that are constructed as the dual of triangulations of the surface. The mesh should be viewed as a graph of points with the desired connectivity, i.e., each node has always exactly three neighbors; the triangulation may be viewed as a polyhedral approximation of the object.

As mentioned in the previous section, global regularity can easily be achieved in two dimensions since a curve can always be divided into an arbitrary number of segments of equal length. The equivalent in three dimensions would be a mesh covering a closed surface such that the distance between vertices is constant and is the dual of a triangulation, that is, each node has exactly three neighbors. Unfortunately, it is well known that only approximate global regularity can be achieved in three dimensions since only three types of meshes are strictly regular in three dimensions.

The approach that we use is recursive subdivision of the dodecahedron which yields a mesh that is "almost" regular in that all but 12 pentagonal cells have hexagonal connectivity. The triangulation is constructed by subdividing each triangular face of a 20 face icosahedron into N^2 smaller triangles. The final mesh is built by taking the dual of the $20N^2$ faces triangulation, yielding a mesh with the same number of nodes. For the experiments presented in this paper, we used a subdivision frequency of $N = 7$ for a total number of nodes of 980.

The next step in going from two to three dimensions is to define a notion of local regularity that leads to invariance properties of the mesh and curvature indicator definition similar to the properties used for 2D curves. The definition of local regularity in three dimensions is a straightforward extension of points, with Q being the projection of P on the plane defined by

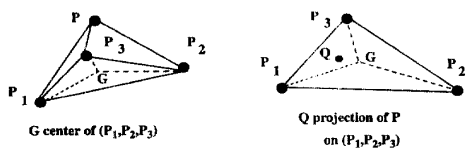


Fig. 3. Local regularity in three dimensions.

P_1, P_2, and P_3 (Fig. 3). The local regularity condition simply states that Q coincides with G. This is the same condition as in two dimensions, replacing the triangle (P_1, P_2, P) by the tetrahedron (P_1, P_2, P_3, P). The local regularity condition is invariant by rotation, translation, and scaling because it is purely local and involves only relative positions of the nodes with respect to each other, not absolute distances.

C. Discrete Curvature Measure: Simplex Angle

The last step in building a discrete surface representation is to define an indicator of curvature that can be computed from a mesh with the appropriate regularity properties. We propose a definition in terms of angular variation between neighbors in the mesh according to the definition used in the case of 2D contours. We need to define some notation (Fig. 4a). Let P be a node of the mesh, P_1, P_2, P_3 its three neighbors, O the center of the sphere circumscribed to the tetrahedron (P, P_1, P_2, P_3), Z the line passing through O and through the center of the circle circumscribed to (P_1, P_2, P_3). Now, let us consider the cross section of the surface by the plane Π containing Z and P. The intersection of Π with the tetrahedron is a triangle. One vertex of the triangle is P, and the base opposite to P is in the plane (P_1, P_2, P_3) (Fig. 4b). We define the angle φ_0 as the angle between the two edges of the triangle intersecting at P. By definition, φ_0 is the discrete curvature measure at node P. We call φ_0 the simplex angle at P, since it is the extension to a 3D simplex, the tetrahedron, of the notion introduced for a 2D simplex, the triangle.

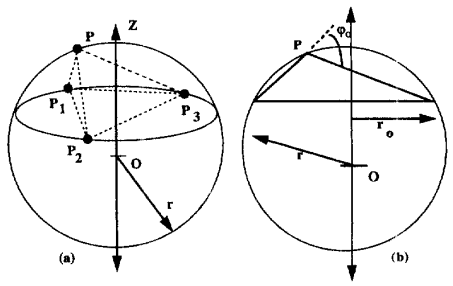


Fig. 4. Definition of the simplex angle.

The simplex angle varies between $-\pi$ and π. The angle is 0 for a flat surface, and is large in absolute value if P is far from the plane of its three neighbors. The simplex angle is negative if the surface is locally concave, positive if it is convex, assuming that the set of neighbors is oriented such that the normal to the plane they form is pointing toward the outside of the object. This behavior of the simplex angle is consistent with the intuitive notion of local "curvature" of a surface. The simplex angle is invariant by rotation, translation, and scaling since it depends only on the relative positions of the nodes in the neighborhood, not on their absolute positions. An experimental comparison of φ_0 with other measure of curvature is described in [5].

In the rest of the paper, we denote by g the function that maps a node to its simplex angle; the simplex angle φ_0 at a node P will be denoted by $g(P)$.

D. Spherical Attribute Image

We now extend the circular representation developed in two dimensions to a spherical representation in three dimensions. Let $\mathcal{M}$ be mesh of points on a surface such that it has the topology of the quasi-regular mesh of Section II.B. Let $\mathcal{S}$ be a reference mesh with the same number of nodes on the sphere of unit one. We can establish a one-to-one mapping h between the nodes of $\mathcal{M}$ and the nodes of $\mathcal{S}$. The mapping h depends only on the topology of the mesh and the number of nodes. Specifically, for a given size of the mesh $M = 20 \times N^2$, where N is the frequency of the mesh (Section II.B), we can define a canonical numbering of the nodes that represents the topology of any M-mesh. In other words, if two nodes from two different M-meshes have the same index, so do their neighbors. With this indexing system, $h(\boldsymbol{P})$, where $\boldsymbol{P}$ is a node of the spherical mesh, is the node of the object mesh that has the same index as $\boldsymbol{P}$.

Given h, we can store at each node $\boldsymbol{P}$ of $\mathcal{S}$ the simplex angle of the corresponding node on the surface $g(h(\boldsymbol{P}))$. The resulting structure is a quasi-regular mesh on the unit sphere, each node being associated with a value corresponding to the simplex angle of a point on the original surface. As an analogy to the EGI, we call this representation the spherical attribute image(SAI). In the remainder of the paper, we will denote by $g(\boldsymbol{P})$ instead of $g(h(\boldsymbol{P}))$ the simplex angle associated with the object mesh node $h(\boldsymbol{P})$ since there is no ambiguity.

The fundamental property of the SAI is that it unambiguously represents an object up to a rotation. More precisely, if $\mathcal{M}$ and $\mathcal{M}'$ are two meshes on the same object with the same number of nodes both satisfying the local regularity condition, then the corresponding SAIs $\mathcal{S}$ and $\mathcal{S}'$ are identical up to a rotation of the unit sphere. Strictly speaking, this is true only as the number of nodes becomes very large because the nodes of one sphere do not necessarily coincide with the nodes of the rotated version of the other sphere. (This problem is addressed in Section IV.A.) One consequence of this property is that two SAIs represent the same object if one is the rotated version of the other.

III. Building Intrinsic Representations from 3D Data

In the previous sections, we have defined the notion of locally regular mesh and its associated SAI. In this section, we describe the algorithm developed for computing such a mesh from input data. The general approach is to first define an initial mesh near the data points and to deform until it satisfies two conditions: It must be close to the input object, and it must satisfy the local regularity condition. The first condition ensures that the resulting mesh is a good approximation of the object, while the second condition ensures that a valid SAI can be derived from the mesh.

A. Mesh Deformation

Given input sensor data and a reference mesh, the problem is to deform the mesh so that it fits the data. In order to perform the fitting, we developed a technique based on deformable surfaces in which each node of the mesh is subjected to a force that is a function of its distance to the closest data point. Given an initial shape, e.g., a sphere surrounding the object, the mesh deforms itself iteratively until it reaches a stable configuration. In addition to the data forces, smoothness and inertia forces are incorporated into the deformation model in order to ensure that the resulting mesh is smooth and that the iterations converge. This algorithm is described in detail in [5].

In order to use the mesh for matching, we need to enforce the local regularity constraint described earlier. This is done by defining a new force $\boldsymbol{F}_g$ at every node $\boldsymbol{P}$. $\boldsymbol{F}_g$ is a function of the distance between the projection of $\boldsymbol{P}$ onto the plane formed by its three neighbors and the center of mass the neighbors. The effect of this force is to "pull" each node of the mesh in the direction in which it best satisfies the local regularity constraint. Except for the addition of $\boldsymbol{F}_g$, the mesh generation algorithms are identical to the algorithms introduced in [5].

B. From Mesh to SAI

Once a regular mesh is created from the input data, a reference mesh with the same number of nodes is created on the unit sphere. The value of the angle at each node of the mesh is stored in the corresponding node of the sphere.

The SAI building algorithm is illustrated in Fig. 5. Fig. 5a shows three views of a green pepper from which three 240×256 range images were taken using the OGIS range finder. The images are merged, and an initial description of the

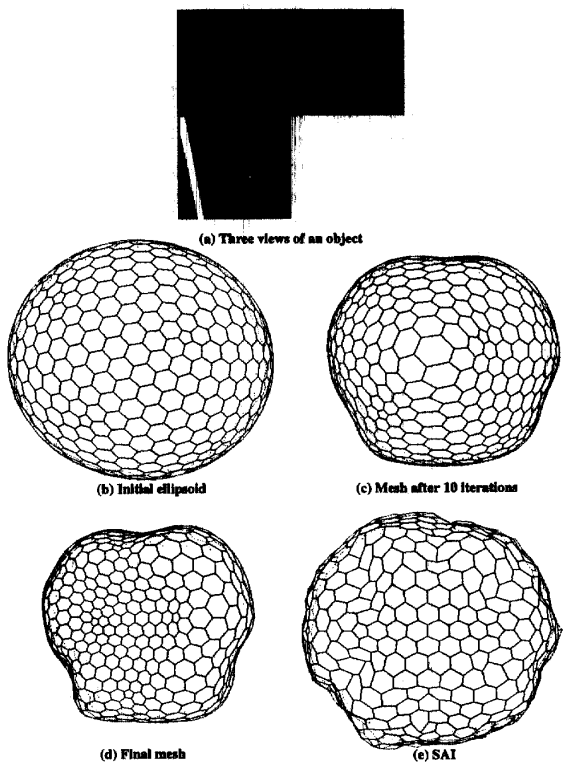


Fig. 5: Building SAI from range data.

object is produced using the deformable surface algorithm. In this example, the three views of the object are precisely calibrated with respect to a common coordinate system. As a result, the coordinates of the data points from all three images are expressed in the same reference frame, thus avoiding a difficult registration problem. In general, it would be important to automatically register views to build the models. This could be done using the SAI formalism as well. In the context of this paper, however, we concentrate on the recognition aspects rather than on the model building aspects. Fig. 5b and Fig. 5c show the initial mesh mapped on the ellipsoid and the mesh at an intermediate stage. Fig. 5d shows the final regular mesh on the object. Fig. 5e shows the corresponding SAI. The meshes are displayed as depth-cued wireframes.The SAI is displayed by placing each node of the sphere at a distance from the origin proportional to the angle stored at that node.

IV. Matching Objects

We now address the matching problem: Given two SAIs, determine whether they correspond to the same object. If so, find the rigid transformation between the two instances of the object. As discussed earlier, the representations of a single object with respect to two different reference frames are related by a rotation of the underlying sphere. Therefore, the most straightforward approach is to compute a distance measure between the SAIs which is minimum for the best rotation. Once the rotation is determined, the full 3D transformation can be computed.

A. Finding the Best Rotation

Let $\mathcal{S}$ and $\mathcal{S}'$ be the spherical representations of two objects. Denoting by $g(\boldsymbol{P})$, *resp.* $g'(\boldsymbol{P})$, the value of the simplex angle at a node $\boldsymbol{P}$ of $\mathcal{S}$, *resp* $\boldsymbol{P}$ of $\mathcal{S}'$, $\mathcal{S}$ and $\mathcal{S}'$ are representations of the same object if there exists a rotation $\boldsymbol{R}$ such that:

$$g'(\boldsymbol{P}) = g(\boldsymbol{RP}) \tag{1}$$

for every point $\boldsymbol{P}$ of $\mathcal{S}'$. Since the SAI is discrete, $g(\boldsymbol{RP})$ is not defined because, in general, $\boldsymbol{RP}$ will fall between nodes of $\mathcal{S}$. We define a discrete approximation of $g(\boldsymbol{RP})$, $G(\boldsymbol{RP})$, as follows: Let $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, $\boldsymbol{P}_3$, and $\boldsymbol{P}_4$ be the four nodes of $\mathcal{S}$ nearest to $\boldsymbol{RP}$. $G(\boldsymbol{RP})$ is the weighted sum of the values $g(\boldsymbol{P}_i)$. Formally:

$$G(\boldsymbol{RP}) = \sum_{1}^{4} W(\|\boldsymbol{RP} - \boldsymbol{P}_i\|) g(\boldsymbol{P}_i) \tag{2}$$

where $W(d)$ is a weighting function that is 1 if $d = 0$, and 0 if d is greater than the average distance between nodes. This definition of G amounts to computing an interpolated value of g using the four nearest nodes.

The problem now is to find this rotation using the discrete representation of $\mathcal{S}$ and $\mathcal{S}'$. This is done by defining a distance $D(\mathcal{S}, \mathcal{S}', \boldsymbol{R})$ between SAIs as the sum of squared differences between the simplex angles at the nodes of one of the spheres and at the nodes of the rotated sphere. Formally, the distance is defined as:

$$D(S, S', \boldsymbol{R}) = \sum_{S} \left(g'(\boldsymbol{P}) - G(\boldsymbol{RP})\right)^2 \tag{3}$$

The minimum of D corresponds to the best rotation that brings $\mathcal{S}$ and $\mathcal{S}'$ in correspondence. The simplest strategy is to sample the space of all possible rotations, represented by three angles (φ, θ, ψ), and to evaluate D for each sample $(\varphi_i, \theta_i, \psi_i)$. This approach is obviously expensive; Section IV.C presents better strategies.

It is important to note that the rotation is not the rotation between the original objects; it is the rotation of the representations. An additional step is needed to compute the actual transformation between objects as described below.

B. Computing the Full Transformation

The last step in matching objects is to derive the transformation between the actual objects, given the rotation between their SAIs. The rotational part of the transformation is denoted by $\boldsymbol{R}_o$, the translational part by $\boldsymbol{T}_o$. Given an SAI rotation $\boldsymbol{R}$, for each node $\boldsymbol{P}'$ of $\mathcal{S}'$ we compute the node $\boldsymbol{P}$ of $\mathcal{S}$ that is nearest to $\boldsymbol{RP}'$. Let $\boldsymbol{M}$, *resp.* $\boldsymbol{M}'$ be the point on the object corresponding to the node $\boldsymbol{P}$ of $\mathcal{S}$, *resp.* $\boldsymbol{P}'$. A first estimate of the transformation is computed by minimizing the sum of the distances between the points $\boldsymbol{M}$ of the first object and the corresponding points $\boldsymbol{R}_o\boldsymbol{M}' + \boldsymbol{T}_o$ of the second object. Formally, the expression to minimize is:

$$E_o(\boldsymbol{R}_o, \boldsymbol{T}_o) = \sum \|\boldsymbol{R}_o \boldsymbol{M}'_o + \boldsymbol{T}_o - \boldsymbol{M}\|^2 \tag{4}$$

The sum in this expression is taken over the set of all the nodes of the mesh. The resulting transformation is only an approximation because it assumes that the nodes from the two meshes correspond exactly. We use an additional step to refine the transformation by looking for the node $\boldsymbol{M}$ closest to $\boldsymbol{M}'$ for every node of the mesh and by computing again the minimum of $E(\boldsymbol{R}, \boldsymbol{T})$.

C. Reducing the Search Space

As mentioned in Section IV.A, the exhaustive search approach is computationally expensive. A more efficient approach is based on the observation that the only rotations for which $D(\mathcal{S}, \mathcal{S}', \boldsymbol{R})$ should be evaluated are the ones that correspond to a valid list of correspondences $\{(\boldsymbol{P}_i, \boldsymbol{P}'_j)\}$ between the nodes $\boldsymbol{P}_i$ of $\mathcal{S}$ and the nodes $\boldsymbol{P}'_j$ of $\mathcal{S}'$. Fig. 6a illustrates the idea of correspondences between nodes: Node $\boldsymbol{P}_1$ of the first SAI is put in correspondence with node $\boldsymbol{P}'_{i1}$ of $\mathcal{S}'$ and its two neighbors, $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$, are put in correspondence with two neighbors of $\boldsymbol{P}'_{i1}$, $\boldsymbol{P}'_{i2}$ and $\boldsymbol{P}'_{i3}$, respectively. This set of three correspondences defines a unique rotation of the spherical image. It also defines a unique assignment for the other nodes, that is, there is a unique node $\boldsymbol{P}'_{ij}$ corresponding to a node $\boldsymbol{P}_i$ of $\mathcal{S}$, given the initial correspondences. Moreover, there is only a small number of such initial correspondences, or, equivalently, there is a small number of distinct valid rotations of the unit sphere. In fact, the number of rotations is $3K$ if K is the number of nodes.

Based on this observation, the SAI matching algorithm can be decomposed into two stages: a pre-processing phase and a run-time phase. During pre-processing, we generate the data

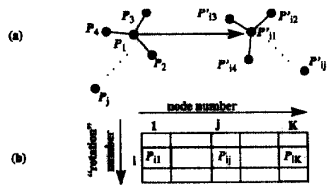


Fig. 6. Efficient matching algorithm; (a) valid correspondence between nodes; (b) table of correspondences.

structure shown in Fig. 6b. The data structure is a two dimensional array in which each row corresponds to a possible rotation of the SAI, and in which column j of row i is the index of the node $\boldsymbol{P}_{ij}$ corresponding to node $\boldsymbol{P}_j$ and correspondence number i. At run time, the distance is evaluated for each row of the array:

$$D_i(S,S',R)=\sum\left(g(\boldsymbol{P}_j)-g(\boldsymbol{P}_{ij})\right)^2$$

The row that produces the minimum D_i gives the best correspondence between nodes of the mesh, $\{(\boldsymbol{P}_j, \boldsymbol{P}'_{ij})\}$, which is used for computing the full transformation between the object meshes as described in the next section. It is important to note that this algorithm tries all possible rotations of the SAIs up to the resolution of the mesh. Consequently, it is guaranteed to find the global optimum of D and it does not require an initial estimate of the transformation. This validates our initial claims of global optimality and pose-independence of the algorithm. This is an efficient algorithm because all that is required at run time is to look up the correspondence table, to compute sum of square differences of corresponding nodes and to sum them. Our preliminary implementation of this approach shows that the computation time can be reduced to a few seconds on a Sparc workstation for K = 980. Initial results also show that the resulting optimal pose is the same as the one obtained by exhaustive search.

D. Example

Fig. 7 shows three views of the same object as in Fig. 5 placed in a different orientation. A model is built from the three corresponding range images using the approach described in Section III.B. Fig. 8 shows the value of the SAI distance measure. The distance measure is displayed as a function of φ and θ only since the distance is a function of three angles that cannot be displayed easily. The displayed value at (φ, θ) is the minimum value found for all the possible values of ψ. The resolution of the graph is 10^o in both φ and θ. This display shows that there is a sharp minimum corresponding to the rotation that brings the SAI in correspondence. Fig. 9 and Fig. 10 illustrate the result of the matching. Fig. 9a shows the superimposition of the cross-sections of both models before matching; Fig. 9b shows the same cross-sections after transformation of the second model using the result of the SAI matching. Fig. 10 shows one of the models backprojected in the image of the other using the computed transformation. Fig. 10a is the original image; Fig. 10b is the backprojected model. This example shows that the transformation is correctly computed in that the average distance between the two models after transformation is on the order of the accuracy of the range sensor.

Fig. 7. Three views of the object of Fig. 5 in different orientations.

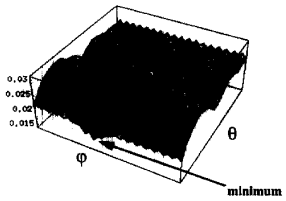


Fig. 8. Graph of distance between SAIs as a function of φ and θ.

Fig. 9. Overlaid cross-sections of the two models; (a) before matching; (b) after matching.

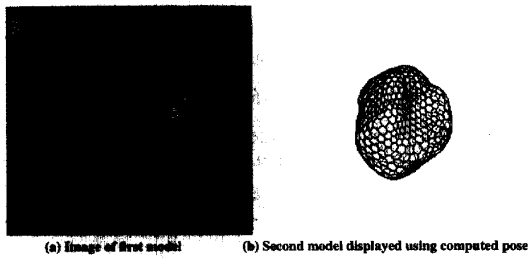


Fig.10. Display of the model in the computed pose.

V. Partial Views and Occlusion

Up to now we have assumed that we have data covering the entire surface of the object, as in Fig. 5. This assumption is appropriate for building reference models of objects. During the recognition phase, however, only a portion of the object is visible in the scene. The matching algorithm of Section IV must be modified to allow for partial representations. The algorithm used for extracting the initial surface model is able to distinguish between regions of the mesh that are close to input surfaces or to data points, and parts that are interpolated between input data. The first type of region is the visible part of

the mesh, and the second type is the occluded part of the mesh. Therefore, even though the representation is a mesh mapped on a closed surface, it is always possible to determine which nodes of the mesh represent valid data.

The situation is illustrated in Fig. 11 in the case of a two dimensional contour. In Fig. 11a a contour is approximated by a mesh of eight points. The mesh is assumed to be regular, that is, all the points of the mesh are equidistant. Let $L = 8l$ be the total length of the mesh. Fig. 11b shows the same contour with one portion hidden. The occluded portion is shown as a shaded curve. The visible section is approximated by a regular mesh of eight nodes of length $L_1 = 8l_1$. Since the occluded part is interpolated as a straight line, the length of this mesh is smaller than the total length of the mesh on the original object. Conversely, the length of the part of the representation corresponding to the visible part, L_2 shown in Fig. 11d, is greater than the length of the same section of the curve on the original representation, L^* shown in Fig. 11c. In order to compute the distance D defined in Section IV, the SAI of the observed curve must be scaled so that it occupies the same length on the unique circle as in the reference representation of the object. If L^* were known, the scale factor would be:

$$k = \frac{L^*}{L_2} \tag{5}$$

In reality, L^* is not known because we do not yet know which part of the reference curve corresponds to the visible part of the observed curve. To eliminate L^*, we use the relation:

$$\frac{L_1}{L} = \frac{L^*}{2\pi} \tag{6}$$

This relation simply expresses the fact that the ratios of visible and total length in object and representation spaces are the same, which is always true when the mesh is regular. Since the left-hand side involves only known quantities, total length of model and observed visible length, L^* can be eliminated by combining (5) and (6):

$$k = \frac{2\pi}{L_2}\frac{L_1}{L} \tag{7}$$

The situation is similar in three dimensions in which case the lengths are replaced by areas A, A_1, A_2, A^*. Relation (7) becomes:

$$k = \frac{4\pi}{A_2}\frac{A_1}{A} \tag{8}$$

The direct extension from two to three dimension is only an approximation because the equivalent of (6), $A_1/A = A/4\pi$, holds only if the area per node is constant over the entire mesh. In practice, however, the area per node is nearly constant for a mesh that satisfies the local regularity condition.

Once k is computed, the appropriate scaling needs to be applied to the SAI. If $\boldsymbol{C}$ is the center of the visible region on the representation sphere, a node $\boldsymbol{P}$ such that θ is the angle $(\boldsymbol{OP}, \boldsymbol{OC})$ is moved to the point $\boldsymbol{P}'$ on the great circle that contains $\boldsymbol{P}$ and $\boldsymbol{C}$ such that:

$$1-\cos\theta' = k(1-\cos\theta) \tag{9}$$

where θ' is the angle $(\boldsymbol{OP}', \boldsymbol{OC})$ and k is the scale factor.

We now show two examples of recognition in the presence of occlusion. In the first example, a range image of an isolated object is taken. A complete model of the object is matched with the SAI representation from range data. Fig. 12 shows the intensity image of the object. Only about 30% of the object is visible in the image. The remaining 70% of the representation built from the image is interpolated and is ignored in the estimation of the SAI distance. Fig. 13a shows the mesh used as a model together with the data points used for building the mesh, and Fig. 13b shows the corresponding SAI. Fig. 14 displays the graph of the distance between SAIs as function of rotation angles. Fig. 14a shows two views of the distance as a function of φ and θ. Fig. 14b shows the same function displayed in φ-ψ space. These displays demonstrate that there is a well defined minimum at the optimal rotation of the SAIs. Fig. 15 shows the model backprojected in the observed image using the computed transformation. In this example, the reference model was computed by taking three registered range images of the object as in the example of Fig. 5.

In the second example, the reference model is built form data points computed from a CAD model. The observed scene is shown in Fig. 16. The result of the matching is shown in Fig. 17.

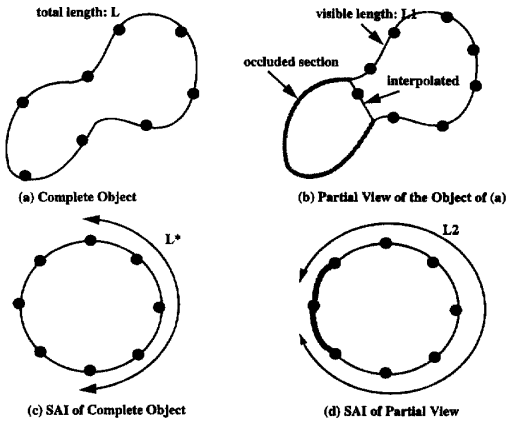


Fig. 11. Matching partial representation in two dimensions.

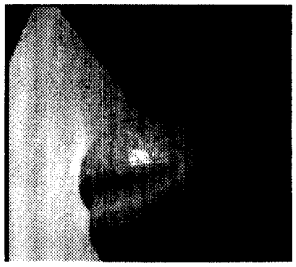


Fig. 12. Input image.

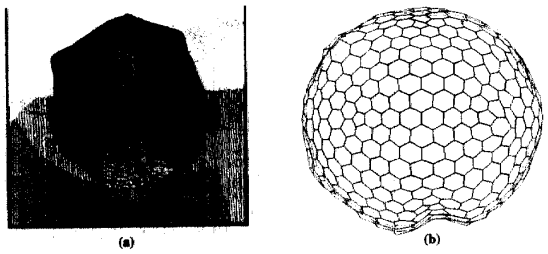


Fig. 13. Reference model; (a) overlay of the mesh and the range data used for building the model; (b) corresponding SAI.

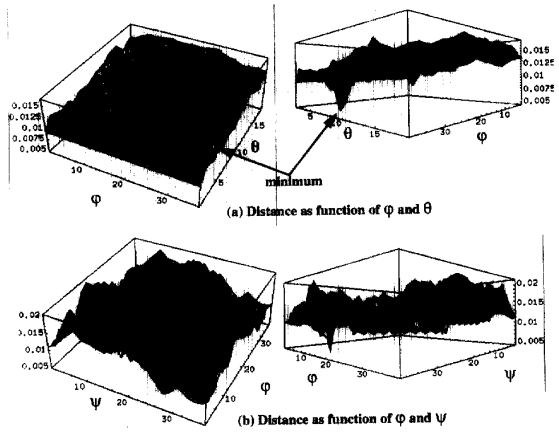


Fig. 14. Sum of squared differences of SAIs as function of rotation angles.

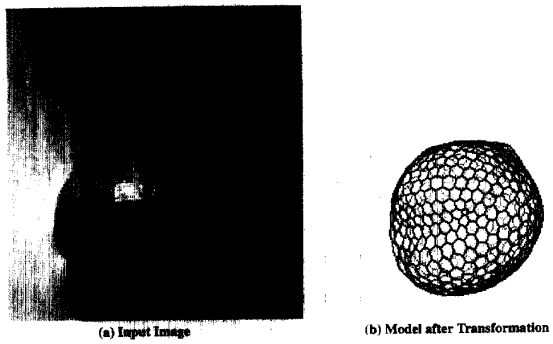


Fig. 15. Display of model using the pose computed from the matching.

Fig. 16. Input image.

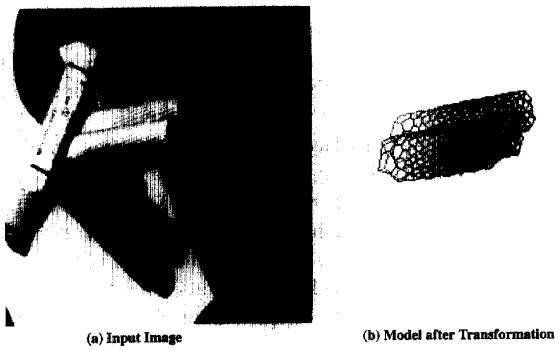


Fig.17. Display of model using the pose computed from the matching.

VI. Performance

For a more quantitative evaluation of the registration, Fig. 18 lists statistics on the registration errors. Those errors were computed on matching three views of the object shown in Fig. 13. The table lists the minimum, maximum, average, and standard deviation of the registration error at the nodes of the mesh. The registration error is defined as the distance between a mesh node and the closest data point after registration. The errors are listed in millimeters in the table. The errors were computed from 869 visible nodes on the object out of 980 nodes on the entire mesh. The mean error is on the order of 0.1 mm which is also the maximum resolution of the range sensor. The standard deviation is on the order of 0.2 mm, reflecting the fact that the error is distributed in a relatively uniform manner. The large maximum error is due to "border effects." Specifically, a node at the edge of the visible part of the mesh may not overlap exactly with a region of the data set, thus causing a large error to be reported. This occurs only at a few isolated nodes at the border.

Error type	Error value (mm)
Min	0.0086
Max	1.11
Mean	0.105
Standard deviation	0.115
Number of points	869

Fig. 18. Statistics of the distances between registered model and data points using the model of Fig. 13.

These numbers show that the registration errors are on the order of the resolution of the sensor, in this case 0.1mm. This shows, in particular, that the node correspondences found through SAI matching are correct and the estimation of the pose based on the correspondences is basically as accurate as it can be given the finite sensor resolution.

VII. Conclusion

In this paper, we introduced a new approach for building and recognizing models of curved objects. The basic representation is a mesh of nodes on the surface that satisfies certain regularity constraints. We introduced the notion of a simplex angle as a curvature indicator stored at each node of the mesh. We showed how a mesh can be mapped into a spherical representation in canonical manner, and how objects can be recognized by computing the distance between spherical representations.

The SAI representation has many desirable properties that make it very effective as a tool for 3D object recognition. Firstly, the SAI is stable with respect to translation, rotation, and scaling of the object. This is not true of most other commonly used representations. This invariance allows the recognition algorithm to compare shapes through the computation of distances between SAIs without requiring explicit matching between object features or explicit computation of object pose.

Secondly, the SAI preserves connectivity between parts of the object in that nodes that are neighbors on the object mesh are also neighbors on the SAI. Thus the SAI does not exhibit the same ambiguity problem for non-convex objects as the EGI and CEGI representations.

Finally, the SAI representation can handle partial views and occluded objects. The basic approach is to measure the area of the visible portion of an object observed in a scene, and deform the SAI mesh model so that the percentage of the sphere corresponding to the visible area is the same in both model and scene SAIs. This approach can be used because a connected visible region of an object corresponds to a connected region on the corresponding SAI.

Results show that the SAI representation is successfully used to determine the pose of an object in a range image including occlusion and multiple objects. This approach is particularly well suited for applications dealing with natural objects. Typically, conventional object modeling and recognition techniques would not perform well due to the variety and complexity of shapes that may have to be handled. The approach is general enough that it can also convert manually built models to the SAI representation.

Acknowledgments

This work was sponsored in part by the National Science Foundation under IRI-9224521, and in part by the Avionics Laboratory, Wright-Patterson AFB, under F33615-90-C-1465 and F33615-93-1-1282.

This research was conducted while H. Delingette was a visiting scholar at the Robotics Institute, and partially supported by a Lavoisier Fellowship.

References

[1] A.D. Aleksandrov and V.A. Zalgaller., "Intrinsic geometry of surfaces," *Translation of Mathematical Monographs Series*, AMS Publisher, 1967.

[2] P. Besl and N.D. Kay, "A method for registration of 3D shapes," *Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239, 1992.

[3] P. Besl and R. Jain, "Segmentation through symbolic surface descriptions," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, pp. 77-85, Miami, 1986.

[4] R.M. Bolle and B.C. Vemuri, "Three dimensional surface reconstruction methods," *Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 1-14, 1991.

[5] H. Delingette, M. Hebert, and K. Ikeuchi, "Shape representation and image segmentation using deformable surfaces," *Image and Vision Computing*, vol. 10, no. 3, pp. 132- 140, April 1992.

[6] H. Delingette, M. Hebert, and K. Ikeuchi, "Representation and recognition of free-form surfaces," Technical Report CMU-CS-92-124, School of Computer Science, Carnegie-Mellon Univ., 1992.

[7] O.D. Faugeras and M. Hebert, "The representation, recognition, and locating of 3D objects," *Int'l J. Robotics Research*, vol. 5, no. 3, pp. 27-52, 1986.

[8] D.A. Forsyth, J.L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell, "Invariant descriptors for 3D object recognition and pose," *Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 971-992, Oct. 1992.

[9] W.E.L. Grimson and T. Lozano-Perez, "Localizing overlapping parts by searching the interpretation tree," *Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 469-482, July 1987.

[10] K. Ikeuchi, "Recognition of 3D objects using the extended gaussian image," *Int'l Joint Conf. on Artificial Intelligence*, pp. 595-600, 1981.

[11] K. Ikeuchi and K.S. Hong, "Determining linear shape change: Toward automatic generation of object recognition program," *Computer Vision, Graphics, and Image Processing: Image Understanding*," vol. 53, no. 2, pp. 154-170, March 1991.

[12] S.B. Kang and K. Ikeuchi, "Determining 3D object pose using the complex extended Gaussian image," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 580-585, June 1991.

[13] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int'l J. Computer Vision*, vol. 2, no. 1, pp. 321-331, 1988.

[14] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson., "Affine invariant model-based object recognition," *IEEE-Robatics and Automation Magazine*, vol. 6, no. 5, pp. 578-589, Oct. 1990.

[15] A.L. Loeb, Space Structures, Addison-Wesley, 1976.

[16] D.G. Lowe, "Three dimensional object recognition from single two dimensional images," *Artificial Intelligence*, vol. 31, 1987, pp. 355-395.

[17] M. Oshima and Y. Shirai, "Object recognition using 3D information," *Trans. Pattern Analysis and Machine Intelligence*, vol. 5, no. 4, pp. 353-361, July 1983.

[18] A.P. Pentland, "Perceptual organization and the representation of natural form," *Artificial Intelligence*, vol. 28, no. 2, pp. 293-331, 1986.

[19] G. Taubin, "Recognition and positioning of rigid objects using algebraic and moment invariants," PhD Dissertation, Brown Univ., 1990.

[20] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D.J. Kriegman, "Parametrizing and aces," *Proc. Computer Vision and Pattern Recognition*, June 1992.

[21] D. Terzopoulos, A. Witkin, and M. Kass, "Symmetry-Seeking 3D object recognition," *Int'l J. Computer Vision*, vol. 1, no. 1, pp. 211-221, 1987.

[22] M. Wenninger, *Polyhedron Models*. London. Cambridge Univ Press, London: 1971.

[23] M. Wenninger., *Dual Models*. London. Cambridge Univ. Press, 1983.

Martial Hebert is a senior researach scientist at the Robotics Institute, Carnegie Mellon University. Dr. Heberet did his graduate work at the Vision Laboratory, INRIA, France before joining the Robotics Institute in 1984.

Dr. Hebert developed techniques for recognizing 3D objects in arbitrary poses using descriptions based on surface primitives. This research introduced the use of direct method for the resolution of the pose estimation problem. He has developed techniques for using range data for navigation, and map building for autonomous mobile robots. Those techniques involve surface description, segmentation, sensor modeling, and matching. In particular, many of those techniques are the basis of the navigation for the unmanned ground vehicle project. More recently, he has been working on developing new representations for general curved objects for shape recognition and for model-building for multiple views. This research has led to the development of the spherical attribute image representation which has been shown to be well suited for representing and matching 3D shapes. Also, he has been developing new calibrated and weakly calibrated stereo techniques for navigation applications. The weakly calibrated stereo research, in particular, is one of the first applications of weak calibration techniques to mobile robot navigation. These techniques have been demonstrated in the context of a robot for planetary exploration. Finally, he is applying previous results in shape segmentation and recognition to problems of interior modeling for supervised teleoperation of robotic systems, in particular for the maintenance of nuclear facilities.

Dr. Hebert is an associate editor of the IEEE *Transactions on Robotics and Automation*.

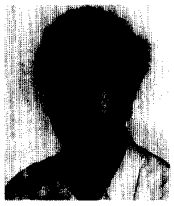
Katsushi Ikeuchi is a principal research scientist (research full professor) at the Computer Science Department and Robotics Institute at Carnegie-Mellon University. Dr. Ikeuchi received the B.Eng degree in mechanical engineering from the University of Tokyo in 1973, and the PhD degree in information engineering from the same university in 1978. After working at the AI Lab at MIT and the ETL of MITI in Japan, he joined the School of Computer Science at Carnegie Mellon in 1986.

Dr. Ikeuchi developed the "smoothness constraint," a constraint to force neighboring points to have similar surface orientations, by which he iteratively recovered shape from shading and shape from texture. He pioneered the use of specular reflections to recover surface orientations. Instead of discarding specular reflections, he actively used them for recovering shape and reflectance. Recently, this method evolved into photometric sampling, which determines not only the object shape but also surface characteristics. He has also been working to develop object representations for vision-guided manipulation for such tasks as bin picking of man-made objects and sampling of natural objects. The representations he has proposed include the EGI, the complex EGI, the spherical angle image, and a frame-based geometric/sensor modeling system (VANTAGE). Currently, he is developing vision algorithm compilers which automatically convert object and sensor models into vision programs.

He was program chairman for the 1993 IEEE CAD-Based Vision Workshop and will be general co-chairman of the 1995 Intelligent Robotics and Systems Conference, and program co-chairman of the 1996 International Conference on Computer Vision and Pattern Recognition. He served on the program committees of several international conferences.

Dr. Ikeuchi is on the editorial board of *IEEE Transactions on Robotics and Automation, International Journal of Computer Vision, Journal of Manufacturing Sysems* and the Optical Society of America.

Dr. Ikeuchi has received several awards including the David Marr Prize in computational vision, and an IEEE Outstanding Paper award. In addition, in 1992, his paper, "Numerical Shape from Shading and Occluding Boundaries," was selected as one of the most influential papers to have appeared in the *Journal of Artificial Intelligence* within the past 10 years.



Hervé Delingette received the Diplome d'Ingénieur from the École Centrale de Paris in 1989, and the PhD degree in the area of computer vision in 1995. He is currently a senior research scientist at the Epidaure project at INRIA, Sophia-Antipolis, France.

From 1989 to 1992, Dr. Delingette was a visiting scholar with the Vision Group at Carnegie-Mellon University. In 1992, he joined the Vision and Graphics group of Nippon Telegraph and Telephone, Yokusuka, Japan, and is now affiliated with Project EPIDAURE, INRIA, Sophia-Antipolis, France.

His research interests include image processing, deformable models, object recognition, and computer graphics.