Towards Sensor-Derived Models of Objects

F.P. Ferrie J. Lagarde P. Whaite

Computer Vision and Robotics laboratory
McGill Research Center for Intelligent Machines
McGill University
Montréal, CANADA

Abstract

In order to cope with an environment, a robot needs to know where things are located, about the size and shape of objects, and ultimately what the objects are. This paper is about computing such information and presents an overview of our current work in building geometric descriptions of objects from sensor data. Its particular focus is a computational methodology which addresses the constituent problems of reconstruction and parts decomposition.

1. Introduction

In order to cope with an environment, a robot needs to know where things are located, about the size and shape of objects, and ultimately what objects are. This paper is about computing such information and presents an overview of our current work in building geometric descriptions of objects from sensor data. The kinds of descriptions we seek are volumetric and part oriented. Objects are represented as conjunctions of volumetric elements, where each element is used to approximate the shape of its corresponding part [Ferrie 86; Ferrie & Levine 85,88; Pentland 86]. The idea is to proceed in bottom-up fashion using only general constraints about the structure of objects in computing these descriptions. Differential geometry can provide these constraints as well as serve as a unifying framework within which to address the constituent problems of reconstruction [Blake & Zisserman 87], inference [Ferrie & Levine 88] and multiple view integration [Ferrie & Levine 87].

In contrast, the current trend in the literature seems to be towards so-called model-based vision where information extracted from sensor data is used to index into a database of object descriptions [Grimson & Lozano-Perez 84; Bhanu 84; Bhanu 87; Bhanu & Ho 87; Bolles & Cain 82]. Presumably, once a description is matched to invariant features of sensor data, the necessary attributes of an object can be found by look-up. But there are some shortcomings with this approach, primarily the difficulty in creating generic object models [Biederman 85] and the complexity incurred by explicitly representing each instance of an object in the domain of observation [Grimson & Lozano-Perez 84]. While model-based vision does indeed work, we believe that there is more to be gained in the long run, both in terms of generality and insight into the larger vision task, by investigating the problem of how basic structure and shape information can be inferred from sensor data.

Our previous work focused on the problem of computing coarse descriptions of objects from intensity and range data [Ferrie & Levine 88]. By trading off loss of fine detail against smoothing, it was shown that it is possible to obtain coarse, but stable descriptions of the form shown in Figure 1b from the data shown in Figure 1a. While adequate for certain tasks, the earlier approach was limited in that it had no provision for evolving descriptions from coarse to fine, and was also limited to measurements from a single sensor. This paper describes an evolution of the earlier methodology which extends the computational framework to remove these limitations. By augmenting the local representation and applying minimization techniques, it is possible to reliably estimate features at finer resolution and combine estimates from different sensors. Recent work by Zucker and his co-workers [Zucker et al. 88] suggests that a similar approach can also be used in the aggregation of local features.

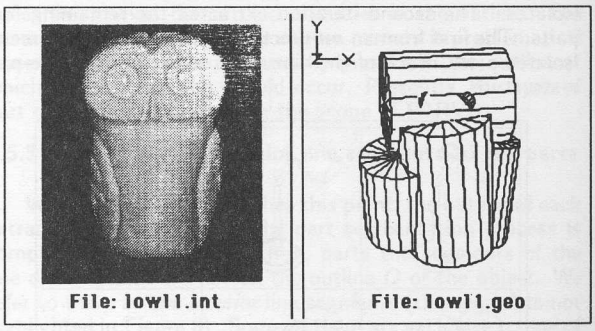

Figure 1a & 1b (a) Image of a stone owl sculpture (b) Rendition computed from (a) using simple volumetric primitives

We begin in Section 2 by outlining a paradigm developed earlier for deriving volumetric descriptions of objects from sensor data [Ferrie & Levine 88]. This is followed in Section 3 by a brief review of the mathematical tools used for computation as well as the representations we use for objects. The principal contribution of this paper follows next in Section 4 and describes the computational approach used to extract surface features and determine the parts decomposition. The testbed used for implementing our models and performing experiments is briefly described in Section 5 with the results of some experiments shown in Section 6.

2. Images to Surfaces to Objects: A Paradigm

Objects can be described at many different levels of abstraction, depending on what properties need to be made

explicit. For example, one might be interested in coarse geometric properties of the form shown in Figure 1, which make basic features of size and shape explicit [Ferrie & Levine 88]. The basis for this representation is the notion of a *parts decomposition*, in which the surfaces of an object are partitioned into regions corresponding to natural parts [Hoffman & Richards 84]. If such a partitioning of a surface can be found, then the task remains as to how to infer the more global geometric properties from each of these regions. Differential geometry plays a dual role here. Local properties such as surface curvature can provide cues about how to partition a surface along natural part boundaries [Hoffman & Richards 84; Ferrie & Levine 88]. Inferences about the more global properties of a region on a surface can be drawn by applying principles of differential geometry *in the large* [Hilbert & Cohn-Vossen 52].

However, in order to be able to apply these tools, one requires a *stable* representation of the surfaces of an object in a form suitable for analysis. Whether initial surface estimates are recovered indirectly from the properties of an image, or else acquired by direct surface measurement (e.g. a laser rangefinder), such estimates are only samples of a surface as well as being subject to noise and quantization error. Thus, there is a considerable gap between sensor measurements and the stable analytic forms required for analysis. We refer to this as the problem of reconstruction [Blake & Zisserman 87]. Other factors that must be considered in deriving object models are the availability of measurements from different kinds of sensors as well as the fact that scenes are rarely static, i.e. the observer and/or the object being viewed can move [Ferrie & Levine 87].

The above considerations lead to the following paradigm for deriving object models:

1. Reconstruction: The surfaces of an object are recovered from samples estimated from sensor data. Important properties of a surface such as the location of discontinuities and features of the differential geometry are computed as by-products of the reconstruction process. The reconstruction process may also encompass the integration of information from different viewpoints and/or sensors.
2. Parts Decomposition: A correspondence is determined between naturally defined parts of an object and regions of the reconstructed surface. We refer to these regions as surface patches.
3. Inference: Given a repertoire of basic shapes for representing the 3-D geometry of parts (i.e. volumetric primitives), the one that most closely resembles the structure of a particular patch is selected. This selection is made by inferring the larger structure of the patch from its local properties. The parameters of the selected part model are determined by fitting it to the corresponding patch.

3. Representation

The paradigm for deriving object models is based on computing three distinct representations: the local neighbourhood of a point described by its augmented Darboux frame ξ_P, the sets of frames S_i which describe the surfaces of each part (known collectively as the parts decomposition), and the volumetric elements V_j corresponding to each part.

3.1 Mathematical Preliminaries

The representation for surfaces makes use of some basic concepts from differential geometry. An important property of a surface S in this regard is *normal curvature* κ_n. Let T_p be defined as the plane tangent to a point $P \in S$, and N_P as the unit normal vector to S at P (Figure 2). Now let Π_N be a plane containing N_P and notice that for any particular direction $\mathrm{v} \in T_p$, Π_N intersects S in a contour called a *normal section*. The normal curvature κ_n at a point P in the given direction is defined to be the curvature of its normal section with its value given by

$$\kappa_n = -\langle dN_P(\mathrm{v}), \mathrm{v} \rangle = \mathrm{II}_P(\mathrm{v}) \ , \qquad (1)$$

where v is a unit vector that defines the orientation of Π_N, and $\mathrm{II}_P(\mathrm{v})$ is the second fundamental form [do Carmo 1976]. As Π_N is rotated about N_P, κ_n can take on maximum and minimum values corresponding to the principal curvatures κ_M and $\kappa_{\mathcal{M}}$ respectively. The directions associated with each are referred to as the principal directions M and $\mathcal{M}$.

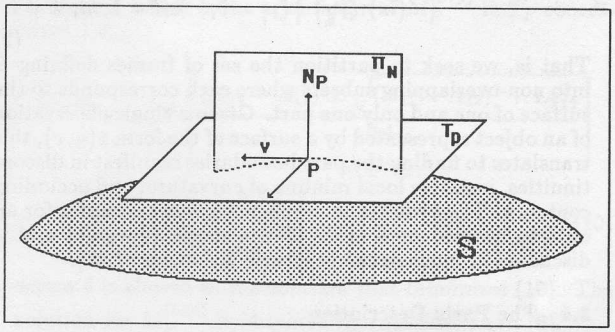

Figure 2 A surface and its tangent plane

One can estimate these quantities with a local approximation of S of the form $\bar{x}(u,v) = (u, v, h(u,v))$, where $\bar{x}(u,v)$ is approximated using points *sampled* from S. Methods for local estimation have been described elsewhere, e.g. [Besl & Jain 86; Fan et al. 86; Ferrie & Levine 88; Sander 88] to name but a few.

3.2 Local Representation of a Surface

Sander [88] introduced a local representation for surfaces that offers a number of advantages with respect to problems of surface reconstruction and parts decomposition. The local structure of a surface at a point P is represented by its augmented Darboux frame, $\xi_P = (P, \kappa_M, \kappa_{\mathcal{M}}, M_P, \mathcal{M}_P, N_P)$, where κ_M and $\kappa_{\mathcal{M}}$ are the principal curvatures, M_P and $\mathcal{M}_P$ the principal direction vectors, and N_P the unit normal vector respectively at point $P \in S$. We use this representation in two ways. First, it determines a parabolic quadric surface $\bar{x}(u,v)$ in local coordinates at each point $P \in S$. This is used in a constrained least-squares minimization process which results in a stable interpretation of the surface and its features [Sander 88]. Second, as a result of this minimization process, the frames make explicit the principal curvatures and directions at each point on the quantized surface. Local minima in the principal curvatures provide important cues for parts decomposition [Hoffman & Richards 84].

3.3 The Parts Decomposition

A parts decomposition is a segmentation of a surface into regions that correspond to different parts of an object. The criteria we use in forming the parts decomposition is based on the Hoffman & Richards [84] theory. More formally, Let $S \in R^3$ define a set of augmented Darboux frames corresponding to the surfaces of an object O. We define a surface patch S_i as a partition of S such that

$$S = \bigcup_{i=1}^{n} S_i, \text{ s.t. } (i)\ S_i \subset S$$
$$(ii)\ S_i \bigcap_i S_j = \emptyset, \quad i,j = 1,n;\ i \neq j.$$

where

$$S_i = \bigcup_{k=1}^{m} \xi_{P_k}, \text{s.t. } (iii)\ \xi_{P_k} \subset S_i$$
$$(iv)\ \xi_{P_k} \bigcap \xi_{P_l} = \emptyset, \quad k,l = 1,m;\ k \neq l. \tag{2}$$

That is, we seek to partition the set of frames defining S into non-overlapping subsets where each corresponds to the surface of one and only one part. Given a single observation of an object represented by a surface of the form $\bar{x}(u,v)$, this translates to finding the part boundaries manifest in discontinuities, negative local minima of curvature, and occluding contours. The problem is significantly more complex for an object observed from more than a single viewpoint and is discussed in more detail in [Ferrie & Levine 87,88].

3.4 The Parts Description

Each part is described at two different levels of abstraction, first by a set of frames S_i at the surface level, and second by a volumetric primitive V_j at the level of part geometry. The computational task is to infer the appropriate representation V_j for each S_i and determine the associated parameters, i.e. given some object O,

$$O \approx V = \bigcup_{j=1}^{m} V_j, \quad V_j \in \Gamma. \tag{3}$$

Any shape descriptor can be used to represent a part provided that (i) there is a mechanism for selecting a particular $V \in \Gamma$ where more than one choice exists, (ii) that its parameters can be computed from a given surface description, and (iii) that it ultimately reflects the features of interest [Pentland 86; Ferrie & Levine 88].

4. Computational Aspects

The computational problems that have to be solved follow directly from the representations just outlined, i.e. estimation of ξ_P from sensor data to describe the surface S, the partitioning of S into $\bigcup_i S_i$ to identify part surfaces, and the inference of V_j from S_i to describe part geometry. In previous work each of these problems was solved by local analysis based on differential geometry [Ferrie & Levine 88]. To make this analysis work reliably, averaging was introduced that smoothed out fine details but preserved the essential structure of objects at coarse scales. In this paper we show how reconstruction based on *curvature consistency* [Sander 88] can be be used to reliably estimate local features without loss of detail. This approach has been successfully applied to computing the first two components of the representation described above.

4.1 Estimating the Initial Frame

Most approaches to estimating $\xi_P\ \forall P \in S$ are based on a local least-squares approximation of the local neighbourhood of P [Besl & Jain 86; Fan et al. 86]. This approach can work provided that the neighbourhood is void of discontinuities, adequately sampled, and that noise and quantization error is sufficiently limited. But these constraints are difficult to maintain with real data. For example one can try to improve least-squares approximation by drawing on a larger sample size, but this results in a smoothing of local features and possibly the inclusion of discontinuities. A number of approaches have been proposed to adapt the sampling neighbourhood to minimize these effects, e.g. [Ferrie & Levine 88], but remain largely data dependent.

Instead we choose to view local least-squares as a means of obtaining *initial* estimates of ξ_P. Using this approach one can compute the parameters of a parabolic quadric $f(u,v) = au^2+buv+cv^2$ and determine N, κ_M, $\kappa_\mathcal{M}$, M, and $\mathcal{M}$ using standard methods [do Carmo 76]. The size of the least-squares approximation neighbourhood is selected to be large enough to provide reasonable estimates, yet small enough to minimize smoothing. In addition, the placement of the sampling neighbourhood is set to avoid possible discontinuities [Ferrie & Levine 88]. The problem now is to refine these local estimates into a correct interpretation of the underlying surface.

4.2 Iterative Refinement of the Darboux Frame

Estimates of $\xi_P\ \forall P \in S$ can be improved by applying minimization procedures that incorporate additional information about surface properties. One approach, characterized by methods of variational calculus [Courant & Hilbert 53; Blake & Zisserman 87], employs constraints derived from *global* properties in the minimization. Another, based on differential geometry, approaches the problem from the opposite viewpoint by employing *local* constraints [Sander 88; Parent & Zucker 85]. We have adopted the latter approach in our work largely for two reasons. First, the minimization problem can be completely specified in terms of local properties, resulting in a single interpretation of the surface. Second, the computational formulation of the problem results in an efficient iterative algorithm that converges rapidly [Sander 88].

Sander [88] applied the notion of curvature consistency [Parent & Zucker 85] to the problem of inferring differential structure from three-dimensional images. The essential ideas of Sander's algorithm can be explained with the aid of Figure 3. Let ξ_P and $\xi_{Q\alpha}$ be two adjacent frames on a surface S, and S_P and $S_{Q\alpha}$ their associated parabolic quadrics respectively. If ξ_P and $\xi_{Q\alpha}$ are assumed to lie in the same neighbourhood of S, then an estimate of ξ_P in terms of $\xi_{Q\alpha}$, $\xi_{P\alpha}$, can be derived from $\xi_{Q\alpha}$ by transporting $\xi_{Q\alpha}$ along $S_{Q\alpha}$ to ξ_P [Sander 88]. Now let $R(\xi_P^{(i)}, \xi_{P\alpha}^{(i)})$ be a measure of mismatch between ξ_P and $\xi_{P\alpha}, \alpha \in \mathcal{N}$ at

iteration step i, where $\mathcal{N}$ defines the local support (neighbourhood) of ξ_P. The algorithm minimizes $R(\xi_P^{(i)}, \xi_{P\alpha}^{(i)})$ by updating $\xi_P^{(i+1)}$ with the least mean square error fit to $\mathcal{N}$.

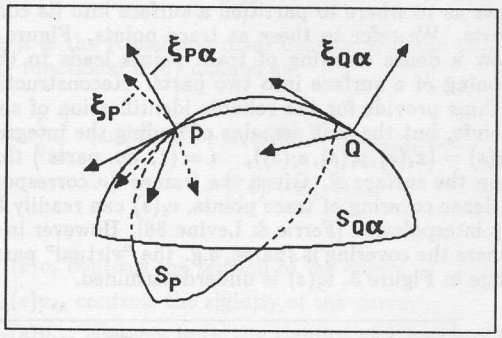

Figure 3 Extrapolating from $\xi_{Q\alpha}$ to $\xi_{P\alpha}$

What curvature consistency means in this context is that the least mean square error fit is subject to constraints defined on $\mathcal{N}$. These constraints fall out of assumptions about the local structure of $\mathcal{N}$, i.e. that $\mathcal{N}$ can be characterized by a C^2 surface. This assumption implies the following constraints which must be formulated as part of the minimization procedure:

$$\left(N_P^{(i)} \cdot N_P^{(i)}\right) = 1 \quad \left(M_P^{(i)} \cdot M_P^{(i)}\right) = 1 \quad \left(M_P^{(i)} \cdot N_P^{(i)}\right) = 0 \tag{4}$$

Sander's algorithm decomposes the minimization into two components corresponding to refinement of the local surface normal $N_P^{(i)}$ and principal direction vector $M_P^{(i)}$ respectively[1]. Sander used the method of Lagrange multipliers to realize the following minimization functional:

$$J = E_1 + E_2$$

$$E_1 = \sum_{\alpha=1}^{n} \|N_P - N_{P\alpha}\|^2 + (\kappa_M - \kappa_{MP\alpha})^2 + (\kappa_{\mathcal{M}} - \kappa_{\mathcal{M}P\alpha})^2 + \lambda((N_P \cdot N_P) - 1)$$

$$E_2 = \sum_{\alpha=1}^{n} \|M_P - M_{P\alpha}\|^2 + \lambda_1(M_P^2 - 1) + \lambda_2(M_P \cdot N_P) \tag{5}$$

where superscripts (i) have been omitted for clarity.

However, E_2 as formulated is dependent on the direction of M_P, which can lead to ambiguities in the resulting minimization. We avoid these difficulties by re-formulating E_2 as follows. Express M in tangent plane coordinates as

$$M_P = \bar{b}_1 \cos\theta + \bar{b}_2 \sin\theta, \ (0, 2\pi) \text{ s.t. } 1) \ \bar{b}_1, \bar{b}_2 \in T_P \quad 2) \ \|\bar{b}_1\| = \|\bar{b}_2\| = 1 \cdot \quad 3) \ (\bar{b}_1 \cdot \bar{b}_2) = 0 \tag{6}$$

[1] Since $\mathcal{M}_P^{(i)}$ is orthogonal to $M_P^{(i)}$, it need not be considered.

Then

$$E_2 = \min_{\theta} \sum_{\alpha=1}^{n} \left[1 - (M_P(\theta) \cdot M_{P\alpha})^2\right]. \tag{7}$$

The technical details relating to the determination of $\xi_{P\alpha}$ and the minimization of E_1 and E_2 are beyond the scope of this paper, but are discussed at length in [Sander 88] and [Ferrie & Lagarde 88]. However we do summarize the resulting updating rules for N_P, κ_{MP}, $\kappa_{\mathcal{M}P}$:

$$N_P^{(i+1)} = \frac{(\sum_{\alpha=1}^{n} N_{x_{P\alpha}}^{(i)}, \sum_{\alpha=1}^{n} N_{y_{P\alpha}}^{(i)}, \sum_{\alpha=1}^{n} N_{z_{P\alpha}}^{(i)})}{\sqrt{(\sum_\alpha N_{x_{P\alpha}}^{(i)})^2 + (\sum_\alpha N_{y_{P\alpha}}^{(i)})^2 + (\sum_\alpha N_{z_{P\alpha}}^{(i)})^2}} \tag{8}$$

$$\kappa_M^{(i+1)} = \sum_{\alpha=1}^{n} \frac{\kappa_{MP\alpha}}{n} \qquad \kappa_{\mathcal{M}}^{(i+1)} = \sum_{\alpha=1}^{n} \frac{\kappa_{\mathcal{M}P\alpha}}{n} \tag{9}$$

The updating rule for $M_P^{(i+1)}$ is obtained indirectly in terms of θ in (8), which is the orientation of $M_P^{(i+1)}$ in T_P coordinates, i.e.,

$$\theta^{(i+1)} = \tan^{-1}\left[\frac{(A_{22} - A_{11}) \pm \sqrt{(A_{11} - A_{22})^2 + 4A_{12}^2}}{2A_{12}}\right]$$

$$A_{ij} = \sum_{\alpha=1}^{n} (M_{P\alpha} \cdot \bar{b}_i)(M_{P\alpha} \cdot \bar{b}_i), \tag{10}$$

where θ is chosen as the solution that minimizes (10). The solution for $M_P^{(i+1)}$ is obtained by substitution in (6).

4.3 Analysis of Surface Features

The point of the minimization strategy is to obtain a description of the surface S that is *stable* with respect to further interpretation [Blake & Zisserman 87]. This allows for a more direct interpretation of features and specifically avoids having to deal with the problem at the level of feature interpretation, e.g. [Boulanger 87]. Determination of features related to obtaining the surface partition is a case in point.

Hoffman & Richards [84] argue that a natural basis for surface decomposition is the principle of transversality regularity. Simply stated, the interpenetration of two arbitrarily shaped surfaces (i.e. corresponding to different parts) results in a contour of concave discontinuity of their tangent planes. In the context of smooth surfaces, our principal focus, this translates into the partitioning of S into parts at loci of negative minima of each principal curvature along its associated family of lines of curvature [Hoffman & Richards 84]. We will refer to such loci as *critical points*. Thus, it is important to have stable estimates of the principal curvatures and directions at each point on S. The following procedure is used to determine the loci of critical points.

Let $\kappa_M(x,y)$ and $\kappa_{\mathcal{M}}(x,y)$ represent stable estimates of the principal curvatures corresponding to S sampled on the discrete grid (x,y), i.e. $S(x,y)$, with corresponding principal directions $M(x,y)$ and $\mathcal{M}(x,y)$. The directional derivatives

in these directions are $\kappa'_M(x,y)|_M$ and $\kappa'_M(x,y)|_M$ respectively. Then a point P is deemed to be a critical point iff

$$\kappa'_M(x,y)|_M = 0 \quad \text{AND} \quad \kappa_M(x,y) < 0$$

OR

$$\kappa'_M(x,y)|_M = 0 \quad \text{AND} \quad \kappa_M(x,y) < 0. \qquad (11)$$

Another important feature of a surface is Gaussian curvature $K_P = \kappa_{MP} \times \kappa_{MP}$, which relates to the generic form of a surface in the vicinity of P. Table 1 below shows some of the inferences that may be drawn from an examination of Gaussian curvature.

Model	Signs (κ_M,κ_m)	Ratios (κ_M,κ_m)
Sphere	(+,+),(-,-)	1
Ellipsoid	(+,+),(-,-)	$\propto$ eccentricity
Cylinder	(+,0),(-,0)	∞,0
Hyperboloid	(+,-)	$\propto$ eccentricity
Planar	(0,0)	undefined

Figure 5 Generic Surface Classification

Of particular interest are points of inflection on S where the sign of Gaussian curvature changes. Critical points and points of inflection are important features in surface inference and reconstruction [Ferrie & Levine 87,88]. In practice, application of (11) in the identification of critical points, and the localization of inflection points in $K(x,y)$ are problems of finding the zero-crossings on a discrete grid. This is easily accomplished by local approximation of the near-neighbourhood of each point by a planar surface $f(u,v) = au + bv + c$ and the analytic verification that $f(u,v) = 0$ for some element in this neighbourhood [Ferrie & Levine 88]. Again, it is important to emphasize that the relatively straightforward determination of critical and inflection points is made possible by having a stable representation of S in terms of ξ_P.

The same is true for occluding contours of S, points of which can be identified by the angle between the surface normal N and view vector V. Furthermore, if S is assumed to be smooth and in orthographic projection to the viewer, then the Z component of N will roll off to zero along the occluding contour. It is this latter property that we use in identifying the trace points of the occluding contour.

One surface feature that is conspicuous by its absence is the surface discontinuity, which can be thought of as the limiting case of a critical point. The problem of locating surface discontinuities is a topic of considerable research, particularly as it relates to surface reconstruction [Blake & Zisserman 87; Leclerc 89]. We are currently pursuing this problem in the context of the curvature consistency algorithm used to reconstruct the surface. For the purposes of the present discussion, however, it is assumed that either the locations of such points can either be found by alternate means, or that surface measurements introduce sufficient smoothing such that a discontinuity can be treated as a critical point.

4.4 Determining the Surface Partition

Critical points, points on the occluding contour and points where the surface is discontinuous are important cues as to where to partition a surface into its constituent parts. We refer to these as trace points. Figure 4 shows how a dense covering of trace points leads to the partitioning of a surface into two parts. Reconstruction algorithms provide for the reliable identification of such trace points, but the task remains of finding the integral curves $v_i(s) = (x_i(s), y_i(s), z_i(s)), \quad i = (1, \text{no. parts})$ that partition the surface S. Given the frames ξ_P corresponding to a dense covering of trace points, $v_i(s)$ can readily be found by interpolation [Ferrie & Levine 88]. However in the case where the covering is sparse, e.g. the "virtual" partitioning edge in Figure 5, $v_i(s)$ is underdetermined.

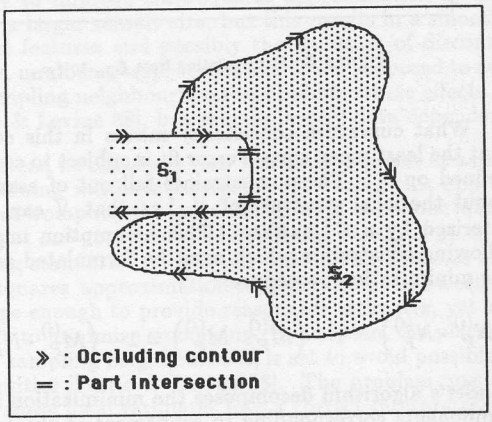

Figure 4 The partition of a surface determined by trace points

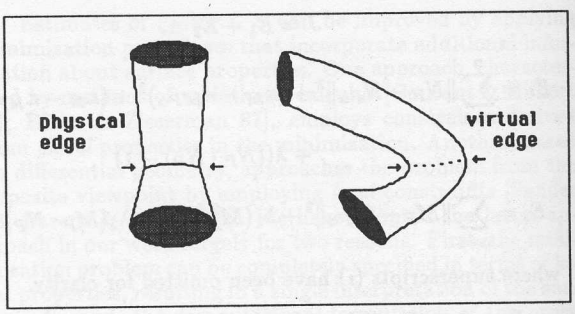

Figure 5 Partitions determined by "dense" and "sparse" coverings of trace points

In such a case one must introduce new constraints that relate to the expected structure of $v_i(s)$. We are currently investigating the application of energy-minimizing spline fitting in the inference of these curves [Kass et al. 88; Terzopoulos 87a,87b; Zucker et al. 88]. Our problem formulation is directly analogous to that described in Zucker et al., [88], where the task is to find a global covering of plane curves through a 2-D tangent field. The difference is that we seek a covering of space curves through a 3-D tangent vector field. Let $v(s,t) = (x(s,t), y(s,t), z(s,t)), \quad 0 \leq s \leq$

1 represent a deformable curve with kinetic energy functional $T(v)$ defined as

$$T(v) = \frac{1}{2}\int_0^1 \mu |v_t|^2 ds, \tag{12}$$

where μ is the (constant) mass density, and the potential energy functional $U(v)$ defined as

$$U(v) = \frac{1}{2}\int_0^1 (\omega_1(s)|v_s|^2 + \omega_2(s)|v_{ss}|^2 + \omega_3(s)|v_{sss}|^2 + I(v) + S(v))ds, \tag{13}$$

where

- $\omega_1(s)v_s$ controls the tension of the curve;
- $\omega_2(s)v_{ss}$ controls the rigidity of the curve;
- $\omega_3(s)v_{sss}$ controls both the rigidity and twisting force (torsion) of the curve;
- I(v) is the potential field coming from the tangent field information (ξ_P);
- S(v) is a force between neighbouring curves that operates when they are in close proximity.

The space curves that we seek are described by those functions $x(s,t), y(s,t)$ and $z(s,t)$ for which

$$\int_{t_0}^{t_1} T(v) - U(v) \quad dt \tag{14}$$

is a minimum. Zucker et al. [88] describe solution methods obtained from the calculus of variations as well as how to obtain the various parameters and potential fields for the 2D case. We are currently investigating an extension of their methodology to the 3D case.

4.5 Fitting Part Models

The set of contours, $\{v_i(s)\}$, effectively partition S into $\bigcup_i S_i$. For each $i = 1$, no. parts, we now seek to infer a corresponding volumetric element V_j that best characterizes its 3-D shape. Different subscripts are used to signify the fact that one or more surface patches S_i can map to a single volumetric element V_j, e.g. where a surface is occluded or where multiple viewpoints are involved [Ferrie & Levine 88]. This really does not pose a problem as multiple instantiations can be resolved at the level of volumetric description using the "same space" assumption [Ferrie & Levine 88]. That is, if two or more parts with similar parametric descriptions occupy the same positions and orientations in 3-D, then they are treated as multiple instantiations of the same part. This reduces the problem of fitting part models to (1) finding the $V_j \in \Gamma$ that best characterizes a particular S_i, and (2) determining the parameters of V_j by minimizing

$$|V_j(x,y,z) - S_i(x,y,z)|. \tag{15}$$

There are a number of different approaches to solving this problem. Pentland [86], for example, used a single superquadric model to represent *all* parts, reducing the problem to a single minimization. Ferrie & Levine [88] used the largest elliptical or cylindrical sub-region of S_i to characterize the geometry of the part according to Table 1, and took advantage of model symmetries in reducing the complexity of the minimization where possible. For example, Figures 6a & 6b show how the centers of curvature, which are easily determined from $\{\xi_P\}_{S_i}$, project into points and axes of symmetry for spheres and cylinders. The same holds true for symmetric ellipsoids, reducing the complexity of the minimization problem that has to be solved. Where such symmetries do not exist, we use a conventional least-squares approach to determine model parameters. The weakness of our present method, however, is that it does not make use of $\{\xi_P\}_{S_i}$ in the least-squares minimization, i.e., it uses the initial raw data sampled from S_i. A better approach, currently under investigation, is the parameterization of model primitives in terms of ξ_P.

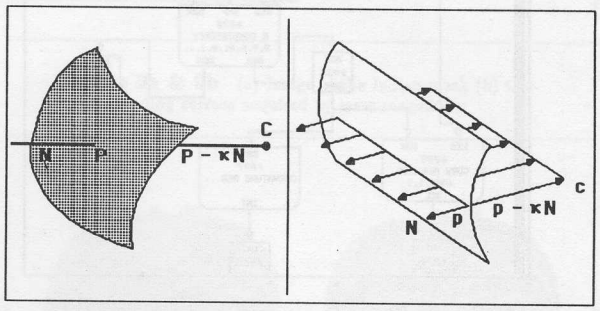

Figure 6a & 6b (a) Center of symmetry and (b) axis of symmetry determined from the loci of centers of curvature

5. The ShapeBench Environment

Part of our efforts have also been directed at the development of a flexible testbed in which the different aspects of the modeling problem can be investigated in a uniform environment. We believe that such tools are an essential part of the research effort.

5.1 A Directed Graph Representation – Arcs, Nodes and Sockets

Our paradigm has a repetitive processing pattern. Algorithms take data and produce new data which is used by other algorithms. We can think of a process flow along the arcs of a directed graph connected between alternating nodes of algorithm and data. A directed graph is a useful representation. When presented visually it gives the user a good overview of the organization of the paradigm and if we can arrange to manipulate the configuration graphically, say with a mouse, then we have a useful and intuitive "programming" method.

Figure 7 shows a Symbolics Lisp machine window in which is displayed the directed graph used to calculate curvature regions [Besl & Jain 86]. There are three basic structures on the graph: nodes, arcs, and sockets. The nodes are the rectangles, the arcs are the lines connecting the nodes, and the sockets are indicated by the three letter tags inside the input and output edges of the nodes. The sockets provide us with a mechanism to ensure that the arc connections between nodes are compatible.

All of the graphic objects are mouse sensitive. When the mouse cursor is positioned over an object it is highlighted to indicate that the right mouse button will display

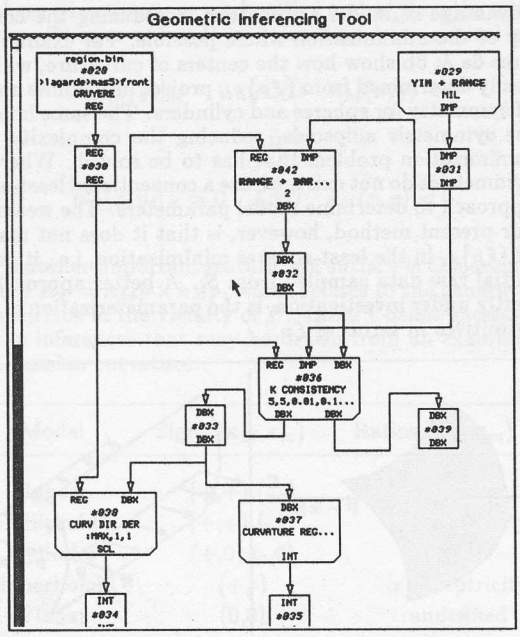

Figure 7 Directed graph display of the process used to calculate curvature regions [Besl & Jain 86].

a menu of operations available for the object. For example node menus always contain operations that can move or delete a node, the sockets menus have an operation to connect an arc, and the arcs an operation to delete the arc. The other mouse buttons, in combination with modify keys, provide shortcuts to the more frequently used operations from the menu. When the mouse cursor is not over an object the right button will display a list of commands. Some examples are commands to create new types of nodes, to save the graph to file, and to clear the graph.

5.2 Function Nodes

The function[2] nodes are indicated by the large rectangles in Figure 7 and are the structures responsible for executing the algorithms in our paradigm. They define node menu operations that permit the value of function parameters to be edited.

The sockets are considered to be part of the function node. The mouse sensitivity of the sockets changes according to context. For example and input will not be sensitive when connecting an arc with an input socket at the other end, or if the tags at the other end is not compatible. Thus we can restrict the kinds of connections made between nodes.

ShapeBench provides a Lisp macro to add new types of function nodes. The specification allows the user to define the inputs, the outputs, and parameters, followed by the code that implements the algorithm. For each input and output the user assigns a symbol, and a map data type. The symbol can be referred to in the algorithm code. Inputs and outputs may be declared as optional. Similarly parameters are also referred to by symbols, and a given a system data type and a default value. A mechanism is provided that allows the user to edit the parameter values.

5.3 Map Nodes

The map[3] nodes are indicated on the directed graph by the smaller square nodes. They are the structures which manage the data used by the algorithms. The map nodes are responsible for allocating, and deallocating the data structures produced by the function nodes. They also define addition node menu operations which can produce a graphic display of the data, for example as a two dimensional intensity image, or as a projected three dimensional image.

Lisp macros allow the user can define new map types. Map types are added to a data base from which is constructed a menu of choices when the command to create a new map node is invoked.

5.4 Updating the Graph

As it stands the graph is only a definition of the algorithm we are investigating. In order to run the algorithms we say that the graph has to be updated. Node menu items are provided that allow the user to update any given node, and a command menu item allows the user to update the entire graph. A node is updated by recursively updating nodes feeding it's inputs. When a node is updated it is time stamped. The time stamp is used by future updates to avoid unnecessary calculations. If a node is modified, for example the value of a parameter is changed, or the connections to it are changed, then the time stamp is changed to show that an update is required.

6. Experimental Results

Figure 8a shows the surfaces of the owl image shown earlier in Figure 1a that were reconstructed using a shape-from-shading algorithm [Ferrie & Levine 86]. Local analysis of curvature (i.e. without applying the minimization algorithm) and contour interpolation resulted in the critical points shown in Figure 8b. In spite of the fact that principal curvatures and directions are computed locally, the results are quite stable. This is largely due to the fact that sufficient smoothing is applied to reduce noise, and the shape of the object is such that the effects of smoothing are minimal. As will be seen in the following example with directly acquired range data, local methods are insufficient to produce stable estimates for interpretation. Once critical points and occluding contours are located, the parts decomposition is determined (Figure 8c), resulting in six surface patches. Two of these correspond to the wings of the owl, another two to the eyes, one to the head, and the remaining patch to the torso. The largest elliptical or cylindrical region of each patch is then used to select a corresponding model primitive according to Table 1. For the case of the owl, a cylinder was automatically selected to represent each surface as shown in Figure 8d. Although this rendition is

[2] A function in the computational, not the mathematical sense.

[3] Historical roots from FORTRAN days when data was stored in two dimensional arrays.

rather coarse, it does describe the basic shape properties of the object without relying on any contextual knowledge. A more faithful rendition could be obtained by using a more flexible part model (e.g. a superquadric), but at the cost of increased computation.

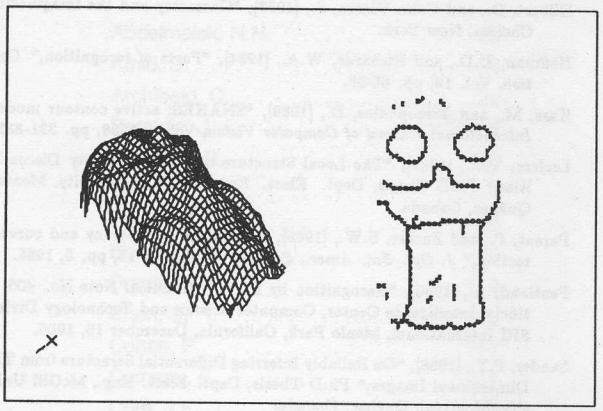

Figure 8a & 8b (a) Surfaces reconstructed from the owl image (b) Critical points on the reconstructed surface

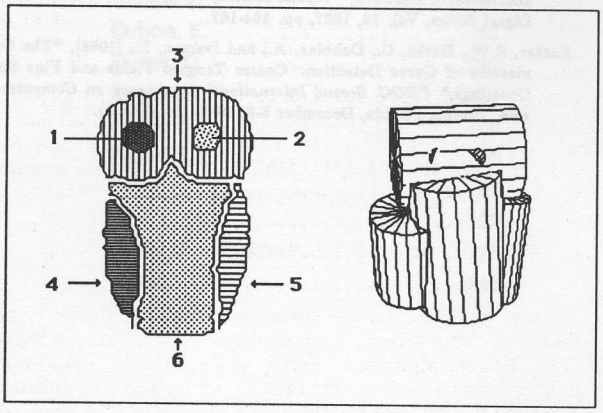

Figure 8c & 8d (c) Parts decomposition of the owl surface (d) Coarse geometric representation computed from the parts decomposition

The next example underscores the need for reconstruction prior to interpretation. Figure 9a shows the image of an Indian mask acquired with a laser rangefinder, and Figure 9b the corresponding surfaces. Local estimates of curvature and orientation tend to be unstable due to noise. For example, Besl and Jain [86] attempted to classify surfaces based on such local estimates using the signs of Gaussian, $K = \kappa_M \times \kappa_M$, and mean curvature, $H = \frac{\kappa_M + \kappa_M}{2}$. Figures 9c and 9d show the results of applying Besl and Jain's classification to estimates of curvature before and after application of the minimization algorithm respectively. Notice a considerable improvement in the results. A further demonstration of the stability of the surface representation after reconstruction can be seen in the identification of critical points on the surface of the mask. Figures 9e and 9f show the before and after results of applying the algorithm; again

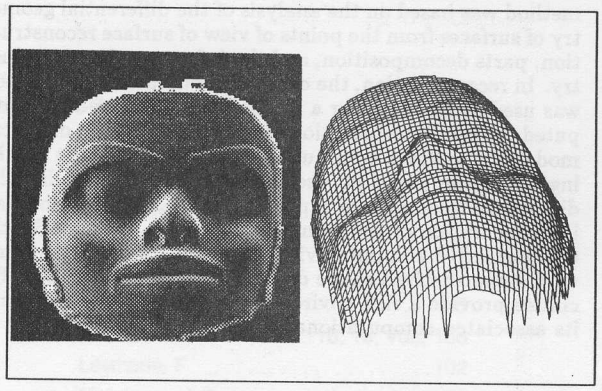

Figure 9a & 9b (a) Image of the Indian mask (b) Corresponding surface acquired by laser rangefinder

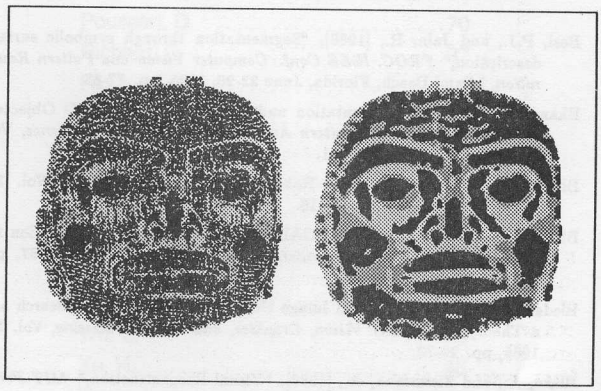

Figure 9c & 9d K-H decomposition before (c) and after (d) application of the minimization algorithm

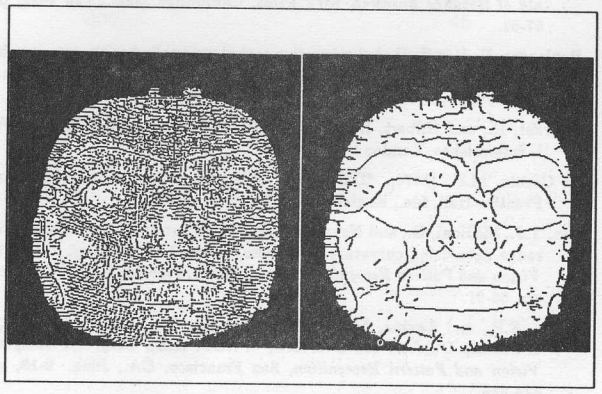

Figure 9e & 9f Critical points detected (e) before and (f) after application of the minimization algorithm

there is a considerable improvement. Observe how the contours formed by these points serve to partition the surface into its constituent parts.

7. Conclusions

In this paper we have described a bottom-up approach to obtaining descriptions of shape from sensor data. The

method was based on the analysis of the differential geometry of surfaces from the points of view of surface reconstruction, parts decomposition, and the inference of part geometry. In reconstruction, the concept of curvature consistency was used as the basis for a minimization scheme that computed the best interpretation of the surface in light of a local model of curvature continuity. We showed how the resulting algorithm could be used to accurately determine the differential geometry of range data. We also described an implementation of the resulting paradigm in ShapeBench, an interactive computer vision system for experimenting with sensor-derived shape descriptions. Together, these facilities provide a rich environment for studying shape and its associated computational problems.

References

Besl, P.J., and Jain, R., [1986], "Segmentation through symbolic surface description," *PROC. IEEE Conf. Computer Vision and Pattern Recognition*, Miami Beach, Florida, June 22-26, 1986, pp. 77-85.

Bhanu, B., [1984], "Representation and Shape Matching of 3D Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 3, pp. 340-351.

Bhanu, B., [1987], "CAD-Based Robot Vision," *IEEE Computer*, Vol. 20, No. 8, August 1987, pp. 13-16.

Bhanu, B., and Ho, C., [1987], "CAD-Based 3D Object Representation for Robot Vision," *IEEE Computer*, Vol. 20, No. 8, August 1987, pp. 19-35.

Biederman, I., [1985], "Human Image Understanding: Recent Research and a Theory," *Computer Vision, Graphics, and Image Processing*, Vol. 32, 1985, pp. 29-73.

Blake, A., and Zisserman, A., [1987], "Visual Reconstruction," *MIT Press*, Cambridge, Massachusetts, 1987.

Bolles, R.C., and Cain, R.A., [1982], "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method," *International Journal of Robotics Research*, MIT Press, Cambridge Mass., Fall 1982, pp. 57-82.

Boulanger, P., [1988], "Label relaxation technique applied to the topographic primal sketch," *PROC. Vision Interface 1988*, Edmonton, Canada, June 6-10 1988, pp. 158-162.

Courant, R., and Hilbert, D., [1953], *Methods of Mathematical Physics*, Vol. I, Interscience, London.

do Carmo, M.P., [1976], "Differential Geometry of Curves and Surfaces," Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Fan, T.J., Medioni, G., and Nevatia, R., [1986], "Description of surfaces from range data using curvature properties," *PROC. IEEE Conf. Computer Vision and Pattern Recognition*, Miami Beach, Florida, June 22-26, 1986, pp. 86-91.

Ferrie, F.P., and Levine, M.D., [1985], "Piecing Together the 3-D Shape of Moving Objects: An Overview," *PROC. IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA., June. 9-13, pp. 574-584.

Ferrie, F.P., [1986], "Reconstructing and Interpreting the 3D Shape of Moving Objects," Ph.D Thesis, Dept. Elect. Eng., McGill University, Montréal, Québec, Canada.

Ferrie, F.P. and Levine, M.D., [1986], "Where and Why Local Shading Analysis Works," *IEEE Trans. PAMI*, (to be published).

Ferrie, F.P., and Levine, M.D., [1987], "Integrating Descriptions From Multiple Views," *PROC. Workshop on Computer Vision*, Miami, Florida, December 1987.

Ferrie, F.P., and Levine, M.D., [1988], "Deriving Coarse 3D Models of Objects," *IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, University of Michigan, Ann Arbor, Michigan, June 5-9, 1988, pp. 345-353.

Ferrie, F.P., and Lagarde, J., [1988], "Curvature Consistency Improves Local Shading Analysis," Technical Report CIM-88-28, McGill University Research Center for Intelligent Machines, November 1988.

Grimson, W.E.L, and Lozano-Perez, T., [1984], "Model-Based Recognition and Localization from Sparse Range or Tactile Data," *International Journal of Robotics Research*, Fall 88, pp. 3-35.

Hilbert, D., and Cohn-Vossen, S., [1952], "Geometry and the Imagination," Chelsea, New York.

Hoffman, D.D., and Richards, W.A., [1984], "Parts of recognition," *Cognition*, Vol. 18, pp. 65-96.

Kass, M., and Terzopoulos, D., [1988], "SNAKES: active contour models," *International Journal of Computer Vision*, Vol. 1, 1988, pp. 321-332.

Leclerc, Y.G., [1989], "The Local Structure of Image Intensity Discontinuities," Ph.D Thesis, Dept. Elect. Eng., McGill University, Montréal, Québec, Canada.

Parent, P., and Zucker, S.W., [1985], "Curvature consistency and curve detection," *J. Opt. Soc. Amer., Ser. A*, Vol. 2, No. 13, pp. 5, 1985.

Pentland, A., [1986], "Recognition by Parts," Technical Note No. 406, Artificial Intelligence Center, Computer Science and Technology Division, SRI International, Menlo Park, California, December 16, 1986.

Sander, P.T., [1988], "On Reliably Inferring Differential Structure from Three-Dimensional Images," Ph.D Thesis, Dept. Elect. Eng., McGill University, Montréal, Québec, Canada.

Terzopoulos, D., [1987a], "On matching deformable models to images," *Topical Meeting on Machine Vision*, Technical Digest Series, Vol. 12, 1987, pp. 160-163.

Terzopoulos, D., [1987b], "On matching deformable models to images: Direct and iterative solutions," *Topical Meeting on Machine Vision*, Technical Digest Series, Vol. 12, 1987, pp. 164-167.

Zucker, S.W., David, C., Dobbins, A., and Iverson, L., [1988], "The Organization of Curve Detection: Coarse Tangent Fields and Fine Spline Coverings," *PROC. Second International Conference on Computer Vision*, Tampa, Florida, December 5-8, 1988, (to appear).