

# A Survey of Methods for Recovering Quadrics in Triangle Meshes

SYLVAIN PETITJEAN  
LORIA-CNRS & INRIA Lorraine

---

In a variety of practical situations such as reverse engineering of boundary representation from depth maps of scanned objects, range data analysis, model-based recognition and algebraic surface design, there is a need to recover the shape of visible surfaces of a dense 3D point set. In particular, it is desirable to identify and fit simple surfaces of known type wherever these are in reasonable agreement with the data. We are interested in the class of quadric surfaces, i.e. algebraic surfaces of degree 2, instances of which are the sphere, the cylinder and the cone. A comprehensive survey of the recent work in each subtask pertaining to the extraction of quadric surfaces from triangulations is presented.

Categories and Subject Descriptors: G.2 [Mathematics of Computing]: General; I.3 [Computing Methodologies]: Computer Graphics; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.4 [Computing Methodologies]: Image Processing and Computer Vision; I.4.6 [Image Processing and Computer Vision]: Segmentation

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Shape recovery, local geometry estimation, mesh fairing, data fitting, geometry enhancement

---

## 1. INTRODUCTION

Broadly speaking, the class of problems this paper examines can be stated as follows: given a piecewise-linear surface, identify, to the extent possible, the regions of the surface well described by curved patches drawn from a given simple class of shapes. Shape recovery problems of this sort occur in diverse scientific and engineering application domains, including:

- *Reverse engineering*: Efficiently manufacturing curved objects is an important issue in modern industry. Indeed, more and more industrial products are being designed with free-form surfaces. When an object has been designed with a CAGD (Computer-Aided Geometrical Design) system, subsequent manipulation and modification can be performed easily, the object being made up of parts of simple geometric shapes. This is not so, however, if a physical prototype is produced and modified directly or if some existing object has not been originally described using a CAGD system. There is thus

---

Author's address: LORIA (Laboratoire lorrain de recherche en informatique et ses applications), Campus scientifique, BP 239, 54506 Vandœuvre-lès-Nancy cedex, France.

E-mail address: Sylvain.Petitjean@loria.fr

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2002 ACM 0000-0000/2002/0000-0001 \$5.00

a need to create geometric models of existing objects. This process constitutes the geometric aspect of reverse engineering (cf. [Várady et al. 1997] for an introduction). An important step in the reverse engineering of solid models from 3D depth maps of scanned objects is the segmentation of the input data, i.e. the grouping of the points in the original dataset into subsets each of which logically belong to a single primitive surface (see Figure 1).



Fig. 1. Reverse engineering of a steering arm [Thompson et al. 1999]. a. Original part. b. Sensed 3D position points. c. Exploded view of the features making up the reverse engineered part.

- *Surface reconstruction and geometry enhancement:* The problem of reconstructing surfaces from a sparse set of unorganized points arises in various contexts such as computer graphics, computer vision and computational chemistry (see [Mencl and Müller 1998] for a survey of ongoing research). Most currently known methods output a polyhedral mesh interpolating the input data ([Boissonnat and Cazals 2000] is a notable exception). But this may not be enough. For instance, people in the CAGD community consider computing a piecewise-linear surface from a point cloud as only a first step of a more global process and what they mean by reconstruction is fully recovering the geometric structure of the data, i.e. the surfaces underlying the point set, be them low-degree surfaces or more complex rational B-splines.

For other applications, a partial recovery of simple shapes (spheres, cylinders, ...) is sufficient. As an example, it was recently shown in the field of global illumination that it is both physically more accurate and computationally more efficient to render low-degree parametric surfaces directly than to tessellate them in dozens of triangles as in traditional radiosity approaches [Alonso et al. 2001] (see Figure 2). But since the bulk of the models known to the computer graphics community are piecewise-linear models, it is important to be able to recover surfaces of simple geometric type in triangulated surfaces. This is all the more so true that most man-made objects can be modeled exactly or at least well approximated by a small set of simple surface patches. Indeed, according to Nourse et al. [1980], 85 percent of all mechanical pieces are well described by patches of planes, cones, spheres and cylinders. If in addition toroidal surfaces are allowed, then this primitive set encompasses 95 percent of conventional, unsculptured parts in industrial environments [Requicha and Voelcker 1982].

- *Model-based recognition and object registration:* Model-based recognition is the task of determining which, if any, of a given set of objects appears in a given 2D image or image sequence. Thus, object recognition is a problem of matching models from a database with representations of those models extracted from image data. To deal with the many possible transformations that an object may undergo in the imaging process, a popular

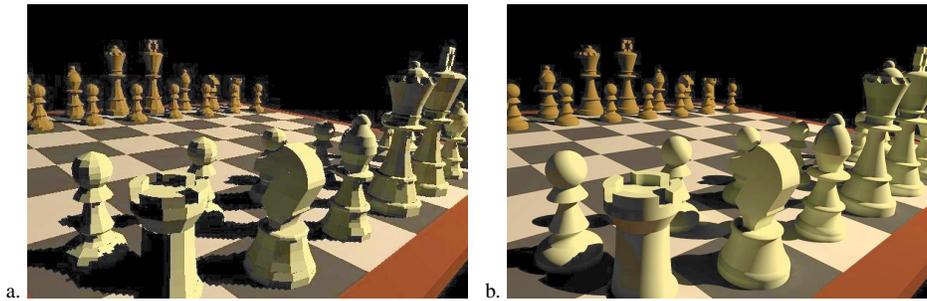


Fig. 2. Illuminating a piecewise-linear mesh (a.) and the quadrics-based model recovered (b.) [Alonso et al. 2001].

approach is to find measurements of the object that are invariant to these types of transformations [Besl and Jain 1986]. If the objects considered are free-form and described by high-order surfaces, matching may prove difficult to achieve. One possible approach is to divide both the base models and the measured data into low-degree surface patches and use the component patches as features in the object recognition system [Bolle and Cooper 1986; Nguyen et al. 1999]. Such a strategy can also be useful to register objects in the input data, i.e. to estimate their pose.

- *Matching overlapping range images:* The data produced by laser range scanning systems typically consists of a rectangular grid of distances from the sensor to the object being scanned. If more sophisticated systems are capable of digitizing some types of curved objects (e.g. cylindrical objects), the scanning of topologically more complex objects (like those having handles) cannot be accomplished directly. Clearly, multiple view points must be used. Matching overlapping range images is a key problem in the construction of CAGD models from full range images (reverse engineering) and for recovering scene descriptions from multiple views in computer vision.

To put a set of overlapping range images in a common 3D reference frame, the traditional approach consists in first determining the motion parameters used to describe the correspondence between points in adjacent images and then reconstructing a composite surface by mapping each range data point into a common reference frame [Soucy and Ferrie 1997; Eggert et al. 1998]. For motion parameter calculation, it is usually necessary to estimate local surface geometry and recover simple surface patches.

- *Analytical representation of segmented volume data:* The development of Magnetic Resonance and Computer Tomography imaging techniques has led to the ability to create 3D data sets and to view, for instance, areas of the human anatomy not previously reachable without invasive procedures. If one is interested not just in viewing the data but also exploring and analyzing them (like for instance modeling the blood flow through a diseased artery), then tools are needed to accurately and precisely recover the shape of objects in an image. Analytically representing the digital shapes obtained by segmenting volumetric industrial or medical images can help in this respect (see, e.g., [Sanderson 1996]).
- *Surface sketching and algebraic surface design:* A major task of CAGD is to automate the design process of such industrial objects as car parts and airplane wings, usually represented by smooth meshes of curves and surfaces. Ideally, one would like to po-

sition a number of key points and curves and let the system infer the interpolating or approximating shape [Bajaj et al. 1993]. Related is interactive surface sketching, where free-form surfaces are created according to a sketch made by the user with the help of a stylus or a mouse [Sachs et al. 1991].

While shape recovery algorithms addressing these problems have often been crafted on a case by case basis to exploit partial structure in the data or in the problem formulation, there are common themes to all methods. In particular, the following tasks always appear, in one form or another: **estimation** (computing the local surface geometry by way of differential parameters such as normals, curvatures, . . .), **segmentation** (dividing the point set into subsets having similar geometric characteristics), **classification** (deciding to which surface type – e.g. spherical vs. cylindrical – segmented points belong) and **reconstruction** (finding the surface best fitting a set of points).

This idealized separation of the tasks is an oversimplification. For instance, differential parameters are more reliable when estimated from a fitted surface, but reconstruction needs segmentation to be done first and segmentation is usually driven by estimates of the curvatures and other differential parameters. In the field of computer vision, researchers have advocated carrying out the three stages of segmentation, classification and fitting simultaneously rather than sequentially [Besl and Jain 1988]. For the sake of clarity of exposition, we shall however follow the natural order proposed above.

Given a polyhedral surface as input, most shape recovery methods start by estimating the local surface geometry at each point. The importance of such intrinsic properties such as surface curvatures for describing shape has been recognized early [Besl and Jain 1986]. Indeed, such properties are unaffected by the choice of the coordinate system and the particular parameterization of the surface. For all practical purposes, the local geometry at a point of a differential surface  $S$  is well captured by a second-order surface which locally “looks” like  $S$ . This is because differential objects like tangents, normals, curvatures and inflections make use of first- and second-order derivatives only.

If this initial step is more or less common to all shape recovery methods, the remaining steps depend heavily on the class of surfaces to be looked for in the data. The search for feature lines along which the mesh is to be segmented is intimately linked to the type of curved surfaces one wants to recover, even though some features, like sharp edges, have to be identified regardless of which surfaces one is after.

In this paper, we are interested in second-order surfaces [Blinn 1997]. This class of surfaces includes such shapes as ellipsoids, hyperboloids, cones, cylinders and paraboloids (see Figure 3). We survey the shape recovery methods that have been specifically tailored for extracting quadric surfaces or that can be nicely adapted to that class of surfaces. We focus on triangulations more than on range images. Shape recovery methods developed for data from structured light sensors such as the laser range finder or depth images from stereo pairs (see [Arman and Aggarwal 1993] for a survey) rarely apply to triangle meshes. Indeed, data from these techniques result in 3D points having a natural parameterization (the one given by the grid on which the points are aligned) which is exploited by shape extraction algorithms. For instance, Jiang and Bunke [1999] use the natural structure of range images to perform a segmentation based on a scan-line algorithm. By contrast, triangulated surfaces have no such natural parameterization [Stokely and Wu 1992].

Note that we are primarily interested in extracting quadrics from discretized data coming from physical objects with exactly quadric boundaries. Indeed, close studies of the

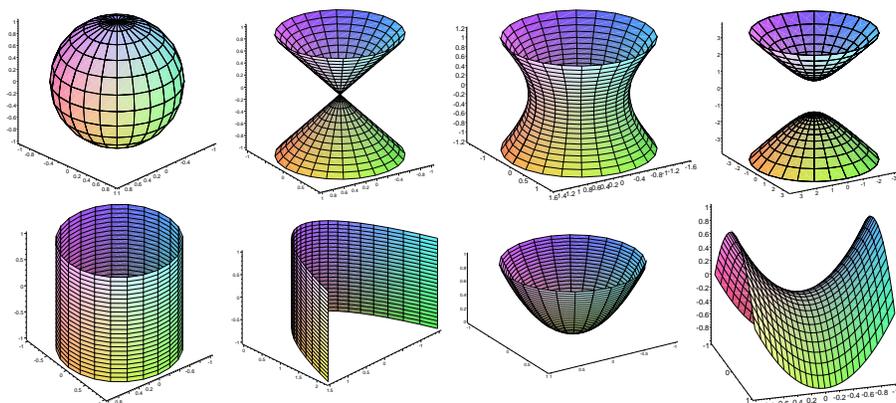


Fig. 3. Some instances of quadric surfaces. First row: ellipsoid, cone, hyperboloid of one sheet, hyperboloid of two sheets. Second row: elliptic cylinder, parabolic cylinder, paraboloid, hyperbolic paraboloid.

problem have led to the conclusion that implicit quadrics are inappropriate for approximating arbitrary data [Moore and Warren 1991; Sapidis and Besl 1995]. In other words, quadric-based segmentation fails to produce acceptable results when the data originate from arbitrarily shaped curved objects (for instance the final regions may be small and have many gaps).

The organization of the paper follows the succession of tasks introduced above. After recalling some notions of differential geometry (Section 2), we review in Section 3 some of the best methods for estimating local surface geometry, i.e. essentially normal and principal curvatures. After differential parameters have been estimated at each data point, we examine in Section 4 how they can be made consistent over the entire triangulation. Then we move on to the segmentation and classification tasks in Section 5. Finally, Section 6 is devoted to the final reconstruction and fitting step, before concluding.

## 2. DIFFERENTIAL GEOMETRY: A FLAVOR

As already indicated, differential geometry provides a convenient basis for describing the local behavior of a surface in the vicinity of some particular point. It has been widely used in computer vision and object recognition as a tool for describing surfaces. Differential geometry parameters are also essential elements of shape recovery methods. Indeed, many methods for estimating the local geometry of a surface approximating a set of points start by computing some differential properties of that surface.

In this section, we recall a few key notions of differential geometry (the interested reader should refer to [do Carmo 1976] for proofs and additional details).

In what follows,  $S$  is assumed to be a surface embedded in  $\mathbb{R}^3$ , described by an arbitrary parameterization of two variables  $\mathbf{X}(u, v)$  and smooth in the vicinity of point  $\mathbf{p}$ .

### 2.1 Fundamental forms and shape operator

Up to orientation of the surface, the unit normal  $\mathbf{n}$  to  $S$  at  $\mathbf{p}$  is given by:

$$\mathbf{n} = \frac{\mathbf{X}_u \times \mathbf{X}_v}{\|\mathbf{X}_u \times \mathbf{X}_v\|},$$

where subscripts indicate partial derivatives and  $\times$  denotes the cross product. The first and second *fundamental forms* of  $S$  are defined as

$$I(u, v, du, dv) = d\mathbf{X} \cdot d\mathbf{X} = d\mathbf{u}^T \mathcal{G} d\mathbf{u}, \quad II(u, v, du, dv) = -d\mathbf{X} \cdot d\mathbf{n} = d\mathbf{u}^T \mathcal{D} d\mathbf{u},$$

where  $d\mathbf{u} = (du, dv)^T$  and

$$\mathcal{G} = \begin{pmatrix} \mathbf{X}_u \cdot \mathbf{X}_u & \mathbf{X}_u \cdot \mathbf{X}_v \\ \mathbf{X}_u \cdot \mathbf{X}_v & \mathbf{X}_v \cdot \mathbf{X}_v \end{pmatrix}, \quad \mathcal{D} = \begin{pmatrix} \mathbf{n} \cdot \mathbf{X}_{uu} & \mathbf{n} \cdot \mathbf{X}_{uv} \\ \mathbf{n} \cdot \mathbf{X}_{uv} & \mathbf{n} \cdot \mathbf{X}_{vv} \end{pmatrix}.$$

The first fundamental form measures the small amount of movement on the surface for a given small movement in the space of the parameters  $(u, v)$ . It is invariant to surface parameterization changes and to translations and rotations of the surface. It does not depend on the way the surface  $S$  is embedded in 3D space, and is therefore an *intrinsic* property of the surface. By contrast, the second fundamental form, which measures the change in the unit normal for a movement in parameter space, depends on the embedding of  $S$  and is thus an *extrinsic* property of the surface.

For  $\mathbf{t}$  a vector tangent to  $S$  at  $\mathbf{p}$ , i.e.  $\mathbf{t} \cdot \mathbf{n} = 0$ , define the *shape operator* or *Weingarten map*  $\beta$  by

$$\beta(\mathbf{t}) = -\nabla_{\mathbf{t}} \mathbf{n},$$

where  $\nabla_{\mathbf{t}} \mathbf{n}$  is the *directional derivative* of  $\mathbf{n}$  in the direction  $\mathbf{t}$  (also called *covariant derivative* of  $\mathbf{n}$  by  $\mathbf{t}$ ):

$$(\nabla_{\mathbf{t}} \mathbf{n})(\mathbf{p}) = \lim_{\tau \rightarrow 0} \frac{\mathbf{n}(\mathbf{p} + \tau \mathbf{t}) - \mathbf{n}(\mathbf{p})}{\tau}.$$

The shape operator is a linear operator  $\Gamma_{\mathbf{p}} \rightarrow \Gamma_{\mathbf{p}}$ , where  $\Gamma_{\mathbf{p}}$  is the space tangent to  $S$  at  $\mathbf{p}$ . In matrix form, it writes down as

$$\beta(\mathbf{t}) = \mathcal{G}^{-1} \mathcal{D} \mathbf{t}.$$

Let  $S$  be the matrix  $\mathcal{G}^{-1} \mathcal{D}$ . One can view the matrix  $S$  as the entity that determines surface shape by relating the intrinsic geometry of the surface to the Euclidean geometry of 3D space. It is a generalization of the curvature of plane curves.

## 2.2 Principal curvatures and directions

Let  $\mathbf{t}$  again be a tangent vector at  $\mathbf{p}$ . The *normal curvature* of  $S$  at  $\mathbf{p}$  in the direction  $\mathbf{t}$  is the curvature of the plane curve formed by the intersection of the plane defined by  $\mathbf{t}$  and  $\mathbf{n}$  with the surface. It is defined by:

$$\kappa_n(\mathbf{t}) = \frac{\beta(\mathbf{t}) \cdot \mathbf{t}}{\|\mathbf{t}\|^2}.$$

The *principal curvatures*  $\kappa_1$  and  $\kappa_2$  at  $\mathbf{p}$  are the maximum and minimum values of  $\kappa_n$  respectively. The unit directions  $\mathbf{e}_1$  and  $\mathbf{e}_2$  for which these values are reached are called the *principal directions* of  $S$  at  $\mathbf{p}$ .  $\kappa_1$  and  $\kappa_2$  are the eigenvalues of the shape operator.  $\mathbf{e}_1$  and  $\mathbf{e}_2$  are the corresponding eigenvectors. With  $\mathbf{n}$ , they form an orthonormal frame at  $\mathbf{p}$ , the *principal coordinate frame*. This frame is well-defined except at *umbilic points*, which are points at which the principal curvatures are equal.

If  $\theta$  is the angle from  $\mathbf{e}_1$  to  $\mathbf{t}$  in the orientation of the tangent plane to  $S$  at  $\mathbf{p}$ , the expression of the second fundamental form in the basis  $\{\mathbf{e}_1, \mathbf{e}_2\}$  is:

$$\kappa_n(\theta) = \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta.$$

It is known as the *Euler formula*. Also, if  $C$  is a curve lying on  $S$  of curvature  $\kappa$ ,  $\mathbf{N}$  its normal vector and  $\varphi$  the angle between  $\mathbf{n}$  and  $\mathbf{N}$  at  $\mathbf{p}$ , then the normal curvature of  $S$  at  $\mathbf{p}$  in the direction of the tangent  $\mathbf{t}$  to  $C$  is

$$\kappa_n = \kappa \cos \varphi.$$

This is *Meusnier's theorem*.

Useful shape descriptors can be derived from the principal curvatures. The *mean curvature*  $H$  is defined as the average of the normal curvatures:

$$H = \frac{1}{2\pi} \int_0^{2\pi} \kappa_n(\theta) d\theta. \quad (1)$$

Using the Euler formula, this reduces to  $H = (\kappa_1 + \kappa_2)/2$ . The *Gaussian curvature*  $K$  is defined as the product of the two principal curvatures:  $K = \kappa_1 \kappa_2$ . In other words,  $K$  is the determinant of  $\mathcal{S}$  and  $H$  is its half-trace. A point of the surface is called *elliptic* if  $K > 0$ , *hyperbolic* if  $K < 0$ , *parabolic* if  $K = 0$  and  $H \neq 0$ , and *planar* if  $K = H = 0$ .

### 2.3 Principal quadric

A neighborhood of a smooth point  $\mathbf{p}$  on  $S$  can be represented in the form  $z = h(x, y)$ , where  $\mathbf{p}$  is the origin of the local coordinate frame and the  $z$  axis is directed by the normal  $\mathbf{n}$  at  $\mathbf{p}$ .  $h$  is a differentiable function and by Taylor's expansion at  $\mathbf{p}$ , we have:

$$h(x, y) = \frac{1}{2}(h_{xx}^{\mathbf{p}}x^2 + 2h_{xy}^{\mathbf{p}}xy + h_{yy}^{\mathbf{p}}y^2) + R(x, y) \quad \text{with} \quad \lim_{(x,y) \rightarrow (0,0)} \frac{R(x, y)}{x^2 + y^2} = 0,$$

where  $h_{xx}^{\mathbf{p}}$  means  $h_{xx}$  evaluated at  $\mathbf{p}$ . (Note that  $h_x^{\mathbf{p}} = h_y^{\mathbf{p}} = 0$  because  $\mathbf{n}$  is along the  $z$  axis.) The quadratic surface  $Q$  defined as the zero-set of the equation

$$z = \frac{1}{2}(h_{xx}^{\mathbf{p}}x^2 + 2h_{xy}^{\mathbf{p}}xy + h_{yy}^{\mathbf{p}}y^2)$$

approximates  $S$  up to order 2 and is called the *principal* or *osculating* quadric of  $S$  at  $\mathbf{p}$ . At a hyperbolic point,  $Q$  is a hyperbolic paraboloid. At an elliptic point, it is an elliptic paraboloid. At a parabolic point, it is a parabolic cylinder. And at a planar point, it is a plane.

Locally,  $S$  can be parameterized by  $\mathbf{X}(x, y) = (x, y, h(x, y))^T$  with  $\mathbf{n} = (0, 0, 1)^T$ . In turn, this means that

$$\mathcal{G} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathcal{D} = \begin{pmatrix} h_{xx}^{\mathbf{p}} & h_{xy}^{\mathbf{p}} \\ h_{xy}^{\mathbf{p}} & h_{yy}^{\mathbf{p}} \end{pmatrix}, \quad \mathcal{S} = \begin{pmatrix} h_{xx}^{\mathbf{p}} & h_{xy}^{\mathbf{p}} \\ h_{xy}^{\mathbf{p}} & h_{yy}^{\mathbf{p}} \end{pmatrix}.$$

In other words, since  $\kappa_1$  and  $\kappa_2$  are the eigenvalues of  $\mathcal{S}$ , the principal quadric has the following equation in the principal coordinate frame:

$$z = \frac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2).$$

The principal quadric thus encodes all the relevant differential information of the surface  $S$  at  $\mathbf{p}$ .

## 3. LOCAL SURFACE GEOMETRY ESTIMATION

Estimating the local surface geometry at a vertex  $\mathbf{p}$  of a piecewise-linear surface amounts to computing what Sander and Zucker called the *augmented Darboux frame* [Sander and

Zucker 1990] at  $\mathbf{p}$ , i.e.

$$\Delta_{\mathbf{p}} = (\mathbf{p}, \mathbf{e}_1, \mathbf{e}_2, \mathbf{n}, \kappa_1, \kappa_2).$$

Computing Darboux frames is a prerequisite to most segmentation and shape recovery methods. It is also a key ingredient of surface re-parameterization, geometry-based subdivision, mesh simplification [Heckbert and Garland 1999] and surface denoising [Desbrun et al. 1999] algorithms. In fact, many surface-oriented applications require an approximation of the first- and second-order differential properties of a piecewise-linear surface with as much accuracy as possible.

Estimating the Darboux frames of a subjacent, unknown, piecewise-smooth surface from a polyhedral approximation is difficult because of the inherently discrete nature of the data. In addition, differential quantities such as principal curvatures and directions, involving second-order partial derivatives, are very sensitive to measurement and quantization errors. In practice, this means that estimates computed on a local basis must be improved in a second global stage of processing (Section 4).

This section examines some of the most powerful methods for locally estimating the differential properties of a triangle mesh<sup>1</sup>. There are three ways to take the information provided by the surface into account:

1. forget the point-to-point connectivity and estimate local geometry using only the 3D vertex set;
2. use the mesh connectivity as a purely topological attribute;
3. interpolate the data using the mesh structure.

We believe the two extremes (1. and 3.) are poor ways of approaching the problem considered. Indeed, methods of the first kind (see, e.g., [Várady et al. 1998]) may have a hard time defining surface orientation and normal in regions where the vertex set is not dense enough, while methods of the third kind need to build an interpolating mesh with high-order continuity (as in [Samson and Mallet 1997], where Bézier patches are used to build a mesh with  $G^2$  continuity), which is hard to achieve. Consequently, this paper focuses on methods of the second type only.

In many methods, vertex normal and principal directions are computed in a single pass. Some iterative methods, however, need an initial estimate of the normal. So we start by indicating how vertex normal can be estimated based on the normals of the incident faces (§ 3.1). We then examine how the principal quadric – and thus the Darboux frame – can be directly estimated by local surface fitting (§ 3.2). We next present some methods based on interpretations in the “discrete” domain of results of differential geometry (§ 3.3), before turning our attention to techniques having a signal processing flavor, which are largely immune to the inherent difficulties of computing differential quantities of quantized objects (§ 3.4). We conclude by saying a word on the comparison between these different methods (§ 3.5).

In what follows,  $T$  is assumed to be an oriented and consistent triangulated surface with boundary [Foley et al. 1992]. In other words, neighboring triangles have their normals pointing to the same side of the surface. For  $\mathbf{p}$  a vertex of  $T$ , we call *1-ring neighborhood*

<sup>1</sup>Though we use triangulations for our presentation, some of the methods presented below are also applicable to more general piecewise-linear surfaces.

of  $\mathbf{p}$  the set of triangles incident to  $\mathbf{p}$ . We denote by  $N(\mathbf{p})$  the set of 1-ring neighbor vertices  $\mathbf{p}_i$  of  $\mathbf{p}$  and  $m$  the cardinal of  $N(\mathbf{p})$  (the valence of  $\mathbf{p}$ ).

### 3.1 Vertex normal estimate

It is commonly suggested to compute the normal at a vertex  $\mathbf{p}$  of a piecewise-linear surface as a (possibly weighted) average of the normals of the faces adjacent to  $\mathbf{p}$ , i.e.

$$\mathbf{n} = \frac{\sum_{i=1}^m w_i \mathbf{n}_i}{\left\| \sum_{i=1}^m w_i \mathbf{n}_i \right\|},$$

where the  $\mathbf{n}_i$  are the unit normals to the triangles in the 1-ring neighborhood of  $\mathbf{p}$ .

Various kinds of averages have been proposed in the literature, in particular arithmetic, area-weighted and angle-weighted averages [Meek and Walton 2000]. Gouraud [1971] takes  $w_i = 1$ , i.e. he considers unweighted averages. Obviously, this definition depends heavily on the local meshing around  $\mathbf{p}$  and two different meshings may result in two different normals. Arguing that the normal vector should be independent of the shape or length of the adjacent faces, Thürmer and Wüthrich [1998] propose an angle-weighted average, i.e.  $w_i = \theta_i$ , where  $\theta_i$  is the angle of the  $i$ -th face at vertex  $\mathbf{p}$  (see Figure 7.c). However, this estimate may be worse than the unweighted average when all vertices are on a smooth surface. Max [1999] derives weights which give the exact normal at  $\mathbf{p}$  if the triangulation is a (possibly irregular) tessellation of a sphere, i.e.

$$w_i = \frac{\sin \theta_i}{\|\mathbf{pp}_i\| \|\mathbf{pp}_{i+1}\|}.$$

Others have proposed taking weights proportional to the areas (or the areas of subregions) of the incident faces. For instance, to stay in line with what is advocated in [Desbrun et al. 2000], one can use weights equal to the areas of the local barycentric cells, as in Figure 7.a.

### 3.2 Fitting for principal quadric estimation

To robustly estimate the local surface geometry of  $T$ , a good start may be to compute the principal quadric at each vertex  $\mathbf{p}$ . We show in this section how the principal quadric can be directly estimated by local surface fitting. Our exposition largely follows that of [McIvor and Valkenburg 1997].

Note that since the differential parameters to be estimated are functions of derivatives up to second order, it is intuitively sufficient to fit second-order surfaces. This is confirmed by experimental results. For instance, Krsek et al. [1998] report little advantage working with third- and fourth-order surfaces over second-order surfaces.

**3.2.1 General approach.** Let  $\mathbf{x}_w$  be a point with coordinates expressed in the global (world) coordinate frame in which the data is obtained. Let  $\mathbf{x} = (x, y, z)^T$  be the coordinates of that point in the principal coordinate frame associated with a given point  $\mathbf{p}$ . Then

$$\mathbf{x} = \mathcal{R}(\mathbf{x}_w - \mathbf{p}_w),$$

where  $\mathcal{R}$  is a rotation matrix called the *attitude matrix*.

Estimating all the parameters of the principal quadric at  $\mathbf{p}$  at the same time may not be easy. In general, this task is separated into two subproblems: estimating the surface normal and estimating the principal directions. This is achieved by making use of a *rotated*

*principal quadric*, which is the principal quadric expressed in a coordinate frame  $\mathbf{x}' = (x', y', z')^T$  related to the principal frame by a rotation about the surface normal:

$$\mathbf{x} = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x}'. \quad (2)$$

The rotated principal quadric has the form

$$z' = a'x'^2 + b'x'y' + c'y'^2 \quad (3)$$

and the associated shape operator matrix is

$$\mathcal{S} = \begin{pmatrix} 2a' & b' \\ b' & 2c' \end{pmatrix}.$$

Its eigenvalues and eigenvectors are the principal curvatures and directions. A straightforward calculation gives:

$$\begin{aligned} \kappa_1 &= a' + c' + \sqrt{(a' - c')^2 + b'^2}, & \kappa_2 &= a' + c' - \sqrt{(a' - c')^2 + b'^2}, \\ \alpha &= \frac{1}{2} \text{atan2}(b', a' - c'), & K &= 4a'c' - b'^2, & H &= a' + c'. \end{aligned} \quad (4)$$

The transformation from the world coordinate frame to a rotated principal frame is then

$$\mathbf{x}' = \mathcal{R}'(\mathbf{x}_w - \mathbf{p}_w). \quad (5)$$

One useful rotated principal frame is defined by choosing  $\mathcal{R}' = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)^T$  as follows:

$$\mathbf{r}_3 = \mathbf{n}, \quad \mathbf{r}_1 = \frac{(\mathcal{I} - \mathbf{n}\mathbf{n}^T)\mathbf{i}}{\|(\mathcal{I} - \mathbf{n}\mathbf{n}^T)\mathbf{i}\|}, \quad \mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1, \quad (6)$$

where  $\mathbf{i}$  is along the first axis in the global coordinate frame and  $\mathcal{I}$  is the identity matrix. In other words, rotation  $\mathcal{R}'$  aligns  $x'$  with the projection of  $\mathbf{x}_w$  onto the tangent plane defined by  $\mathbf{n}$ .

A typical method (e.g. [Sander and Zucker 1990; Stokely and Wu 1992; Ferrie et al. 1993; Hamann 1993]) for estimating the principal quadric then goes as follows:

1. Estimate the surface normal at  $\mathbf{p}$  with one of the definitions of § 3.1 or by finding the plane best fitting  $\mathbf{p}$  and its 1-ring neighbors.
2. Construct the rotation matrix  $\mathcal{R}'$  using (6).
3. Map the world data to the rotated principal frame with (5).
4. Fit the mapped data to the rotated principal quadric (3), and solve the resulting system, giving  $a', b', c'$ .
5. Compute  $\kappa_1, \kappa_2, \alpha$  from  $a', b', c'$  using (4).
6. Estimate the attitude matrix by concatenating the transformations (2) and  $\mathcal{R}'$ .

The coefficients of the rotated principal quadric (Step 4) are obtained by solving the following over-determined system of linear equations:

$$\begin{pmatrix} x_1^2 & y_1^2 & x_1y_1 \\ \vdots & \vdots & \vdots \\ x_n^2 & y_n^2 & x_ny_n \end{pmatrix} \begin{pmatrix} a' \\ b' \\ c' \end{pmatrix} = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}.$$

This system can be solved using a least-squares method.

**3.2.2 Improvements.** Several improvements can be made to this basic principal quadric estimation scheme.

First, it is clear that with the above method, the estimation of the Darboux frame relies heavily on the accuracy of the estimation of the surface normal. Inaccuracies in the surface normal estimate can be allowed by fitting an extended quadric instead of the rotated principal quadric. Compared to the basic method described above, Steps 1-3 and 6 are unchanged and Steps 4-5 now become [McIvor and Valkenburg 1997]:

4. Fit the mapped data to the extended quadric

$$\hat{z} = a' \hat{x}^2 + b' \hat{x}\hat{y} + c' \hat{y}^2 + d' \hat{x} + e' \hat{y},$$

giving a new estimate of the surface normal at  $\mathbf{p}$ :

$$\mathbf{n} = \frac{(-d', -e', 1)^T}{1 + d'^2 + e'^2}.$$

From this, the rotation needed to align  $\hat{z}$  with  $\mathbf{n}$  can be calculated. The data is mapped into this new coordinate frame and a new quadric fitted. The iteration stops when the incremental change in the direction of the normal falls below some tolerance level.

5. Estimate the differential parameters as follows:

$$K = \frac{4a'c' - b'^2}{(1 + d'^2 + e'^2)^2}, \quad H = \frac{a' + c' + a'e'^2 + c'd'^2 - b'd'e'}{(1 + d'^2 + e'^2)^{3/2}},$$

where  $a', b', c', d', e'$  are the parameters of the last quadric fitted.

An iterative method similar to this one is described in [Fitzgibbon and Fisher 1993].

A further improvement is obtained by adding a zero-order term to the quadric to be fitted. This is the same as removing the constraint that the point  $\mathbf{p}$  has to lie on the fitted surface [McIvor and Valkenburg 1997].

An alternative to the linear iterative scheme above is the following 7-dimensional non-linear optimization problem [McIvor and Valkenburg 1997]:

$$\min_{\tilde{\mathcal{R}}, a', b', c', f'} \left\| \tilde{z} - (a'\tilde{x}^2 + b'\tilde{x}\tilde{y} + c'\tilde{y}^2 + f') \right\|, \quad (7)$$

where  $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z})^T = \tilde{\mathcal{R}}(\tilde{\mathbf{x}}_w - \tilde{\mathbf{p}}_w)$ , where the parameter  $f'$  is added to accommodate noise in the data at  $\mathbf{p}$ . Note that Eq. (7) has 4 linear ( $a', b', c', f'$ ) and 3 non-linear (the three Euler angles of  $\tilde{\mathcal{R}}$ ) parameters, so special techniques for separable least-squares can be used.

### 3.3 Surface description from differential geometry

Apart from the above fitting techniques, many methods have been proposed for estimating local surface geometry based on results from differential geometry. We elaborate here on some of the most interesting ones.

**3.3.1 Approximation by curves.** Fast methods for estimating differential parameters result from approximating planar sections of the surface by simple curves.

One such method is the circle fitting algorithm [Martin 1998]. Let  $\mathbf{p}$  be the point at which principal curvatures are to be estimated. The algorithm roughly goes as follows:

- Choose triples of points, each having  $\mathbf{p}$  and two other vertices in the 1-ring neighborhood of  $\mathbf{p}$  (one on each side of  $\mathbf{p}$ ) in common. Let  $l$  be the number of such triples ( $l$  is necessarily  $\geq 3$ ).
- Compute the circle  $C_j$  interpolating each triple,  $j = 1, \dots, l$ . Let  $k_j$  be the curvature of  $C_j$  (i.e. the inverse of the radius of  $C_j$ ) and  $\mathbf{n}_j$  the normal of the plane containing  $C_j$ .
- For each  $C_j$ , the line through  $\mathbf{p}$  and tangent to  $C_j$  (let  $\mathbf{t}_j$  be its direction) can be considered as being tangent to the surface. The normal  $\mathbf{n}$  at  $\mathbf{p}$  is perpendicular to each such tangent line. Estimates of this normal are obtained either by taking tangents  $\mathbf{t}_j$  pairwise and computing the average of the cross products, or by finding the plane  $P$  best fitting all tangents (least-squares fit).
- Let  $\mathbf{t}'_j$  be the projection of  $\mathbf{t}_j$  on  $P$ , i.e.

$$\mathbf{t}'_j = \mathbf{t}_j - (\mathbf{t}_j \cdot \mathbf{n})\mathbf{n}.$$

Each  $C_j$  can be considered as being contained in the surface. So the normal curvature  $\kappa_n(\mathbf{t}'_j)$  of the surface at  $\mathbf{p}$  in the direction  $\mathbf{t}'_j$  is, by Meusnier's theorem,  $\kappa_n(\mathbf{t}'_j) = k_j \cos \varphi_j$ , where  $\varphi_j$  is the angle between  $\mathbf{n}$  and  $\mathbf{n}_j$ .

- Let  $\mathbf{t}_0$  be an arbitrary reference direction in  $P$  and  $\theta_0$  the angle between the first principal direction and  $\mathbf{t}_0$ . Let  $\theta_j$  be the angle between  $\mathbf{t}'_j$  and  $\mathbf{t}_0$ . Then, using the Euler formula, we are left with a set of equations

$$\kappa_n(\mathbf{t}'_j) = \kappa_1 \cos^2(\theta_j - \theta_0) + \kappa_2 \sin^2(\theta_j - \theta_0)$$

in the variables  $\kappa_1, \kappa_2$  and  $\theta_0$  which can be solved using a least-squares method. Rotating  $\mathbf{t}_0$  by an angle of  $\theta_0$  and  $\theta_0 + \frac{\pi}{2}$  then gives  $\mathbf{e}_1$  and  $\mathbf{e}_2$  respectively.

A closely related method is the quadratic complexity algorithm of [Chen and Schmitt 1992], which is also based on fitting circles to triples of points. However, given the point  $\mathbf{p}$  at which curvature is to be measured and one of its neighbors  $\mathbf{p}_i$ , not all triples  $(\mathbf{p}, \mathbf{p}_i, \mathbf{p}_j)$  with  $\mathbf{p}_j$  on the opposite side of  $\mathbf{p}$  are taken into account. The strategy is to define a measure of geometric oppositeness  $M = (\mathbf{p} - \mathbf{p}_i) \cdot (\mathbf{p}_j - \mathbf{p})$ , compute  $M$  for all possible triples, sort the results in decreasing order and keep only the triples corresponding to the first  $l$  values of  $M$  (there are at least 3). This way, the curves  $C$  interpolating the retained triples are close to the normal section and the angle  $\varphi$  between  $\mathbf{n}$  and the normal to the plane containing  $C$  is close to 0 or  $\pi$ . Since Meusnier's theorem involves  $\cos \varphi$ , this limits the effects of computation error for  $\varphi$  and allows for some noise in vertex position to be taken into account.

**3.3.2 Surface normal changes.** Another method considering local approximation by circular arcs is presented in [Karbacher and Häusler 1998]. Assume that the sampling density is high enough so as to neglect variations of surface curvature between adjacent sample points. Let  $\mathbf{n}_i, \mathbf{n}_j$  be the normals estimated at neighboring vertices  $\mathbf{p}_i, \mathbf{p}_j$  (again with any of the definitions of § 3.1). Consider the circle  $S_{ij}$  going through  $\mathbf{p}_i$  and  $\mathbf{p}_j$  and tangent to the surface at these points (i.e.  $S_{ij}$  is orthogonal to  $\mathbf{n}_i$  at  $\mathbf{p}_i$  and to  $\mathbf{n}_j$  and  $\mathbf{p}_j$ ). Let  $\mathbf{p}_{ij}$  be the center of  $S_{ij}$  and  $d_{ij}$  the distance between the two vertices (Figure 4). Then the curvature of  $S_{ij}$  can be estimated as:

$$c_{ij} \approx \pm \frac{\angle \mathbf{p}_i \mathbf{p}_{ij} \mathbf{p}_j}{d_{ij}} \approx \pm \arccos(\mathbf{n}_i \cdot \mathbf{n}_j),$$

where  $c_{ij} > 0$  for a concave area and  $c_{ij} < 0$  for a convex area. The principal curvatures at  $\mathbf{p}_i$  are then estimated as the extrema of  $c_{ij}$  over all vertices  $\mathbf{p}_j$  neighboring  $\mathbf{p}_i$ :

$$\kappa_1 \approx \max_j (c_{ij}), \quad \kappa_2 \approx \min_j (c_{ij}).$$

Knowing  $\kappa_1, \kappa_2$  and a set of normal curvatures at  $\mathbf{p}_i$  (the  $c_{ij}$ ) in several different directions, one can deduce the principal directions by applying Meusnier's theorem and solving the resulting system.

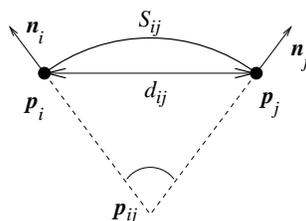


Fig. 4. Approximating the surface by a mesh of circular arcs.

Other authors have considered estimating differential parameters as functions of surface normal changes. The method of [Hoschek et al. 1998] relies on an estimate of the angular variation of the normal close to a particular vertex, while [Krsek et al. 1998] discuss the use of the angular deficit at a vertex as a measure of the curvature. Unfortunately, these two methods may not get the magnitude of the curvature right and they may have troubles with noisy data.

Now consider the following general approach [Flynn and Jain 1989]. For any pair of triangles which share an edge  $\mathbf{p}_i \mathbf{p}_j$ , compute the curvature of the sphere passing through the four vertices involved. (Set this curvature to zero if the four points are coplanar.) The sign of the curvature is taken to be positive if the center of the sphere is on the same side of the surface as the normals at  $\mathbf{p}_i$  and  $\mathbf{p}_j$ , negative otherwise. An estimated curvature value for a given triangle is then taken as the average of the curvatures obtained when it is paired with each of the triangles with which it shares an edge. However, results can be affected by noise in the data.

To compensate for the effect of errors in the positions of triangle vertices, Sacchi et al. [1999] proposed the following algorithm. First, compute an interpolated normal at each vertex of the triangulation as the weighted average of the normals of the incident triangles, with weights equal to the areas of the triangles. Take as compensated normal of a triangle the weighted average of the three interpolated normals at the vertices of the triangle, using as weight, for each vertex, the sum of the areas of the incident faces. Similarly, define the compensated center of each triangle as the weighted average of the vertices using the same weights as before. For a pair of triangles with compensated normals  $\mathbf{n}_1$  and  $\mathbf{n}_2$  and compensated centers  $\mathbf{c}_1$  and  $\mathbf{c}_2$ , take as estimate of the curvature of their common edge

$$c_{12} = \frac{\|\mathbf{n}_1 \times \mathbf{n}_2\|}{\|\mathbf{c}_1 - \mathbf{c}_2\|}.$$

For a given triangle, we thus obtain three curvature values. Pairing the compensated normal of the triangle with the interpolated normals at the three vertices of the triangle and

applying Meusnier's theorem gives three additional curvature values. The estimated mean curvature of a triangle is then taken as the average of the maximum and the minimum of the six curvature measures.

Application of this curvature estimation method on the a surface representing a technical device is shown on Figure 5.

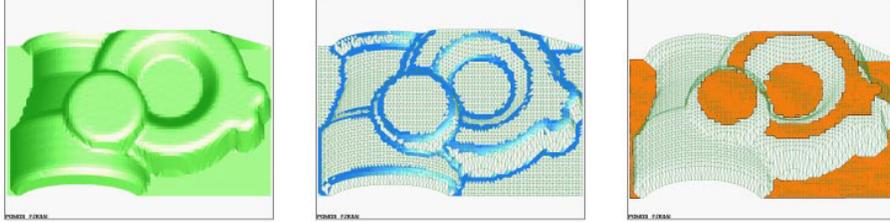


Fig. 5. From left to right: a triangle mesh, regions of high mean curvature (in blue) and identification of planar regions [Sacchi et al. 1999].

3.3.3 *Tensor of curvature.* Using the notations of Section 2, consider the following symmetric  $3 \times 3$  matrix

$$\mathcal{M} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \kappa_n(\mathbf{t}) \mathbf{t} \mathbf{t}^T d\theta,$$

where  $\mathbf{t} = \cos \theta \mathbf{e}_1 + \sin \theta \mathbf{e}_2$ . Taubin [1995b] proves that  $\mathcal{M}$  can be factored as follows:

$$\mathcal{M} = \mathcal{E}_{12}^T \begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix} \mathcal{E}_{12},$$

where  $\mathcal{E}_{12} = [\mathbf{e}_1, \mathbf{e}_2]$  is the  $3 \times 2$  matrix constructed by concatenating the column vectors  $\mathbf{e}_1$  and  $\mathbf{e}_2$  and

$$\kappa_1 = 3m_1 - m_2, \quad \kappa_2 = 3m_2 - m_1. \quad (8)$$

That is, the eigenvectors of  $\mathcal{M}$  are  $0, m_1, m_2$  and the corresponding eigenvectors are  $\mathbf{n}, \mathbf{e}_1, \mathbf{e}_2$ . In other words, an estimation of the tensor of curvature (i.e. the map which assigns each point  $\mathbf{p}$  of  $S$  to  $\kappa_n(\mathbf{t})$ , the normal curvature of  $S$  at  $\mathbf{p}$  in the direction of the unit vector  $\mathbf{t}$ ) follows from an estimation of  $\mathcal{M}$ .

Taubin [1995b] proposes the following algorithm, which is both linear in time and space, to perform this estimation. Let  $\mathbf{p}$  be a vertex of  $T$ ,  $N(\mathbf{p})$  its set of 1-ring neighbor vertices. Let  $\mathbf{n}$  be an estimate of the unit normal at  $\mathbf{p}$ . Then the matrix  $\mathcal{M}$  can be approximated as follows:

$$\mathcal{M}_{\mathbf{p}} = \sum_{\mathbf{p}_i \in N(\mathbf{p})} w_i \kappa_n^i \mathbf{t}_i \mathbf{t}_i^T,$$

where the weight  $w_i$  is chosen to be proportional to the sum of the surface areas of the triangles incident to both  $\mathbf{p}$  and  $\mathbf{p}_i$  (there is only one such triangle if  $\mathbf{p}$  and  $\mathbf{p}_i$  are both on the boundary of  $T$ , two otherwise) and such that  $\sum_{\mathbf{p}_i \in N(\mathbf{p})} w_i = 1$ ,  $\mathbf{t}_i$  is the normalized projection of  $\mathbf{p}_i - \mathbf{p}$  onto the tangent plane  $\langle \mathbf{n} \rangle^\perp$  and  $\kappa_n^i$  is an estimate of the normal

curvature in the direction  $\mathbf{t}_i$ :

$$\mathbf{t}_i = \frac{(\mathbf{p}_i - \mathbf{p}) - [(\mathbf{p}_i - \mathbf{p}) \cdot \mathbf{n}]\mathbf{n}}{\|(\mathbf{p}_i - \mathbf{p}) - [(\mathbf{p}_i - \mathbf{p}) \cdot \mathbf{n}]\mathbf{n}\|}, \quad \kappa_n^i = \frac{2\mathbf{n} \cdot (\mathbf{p}_i - \mathbf{p})}{\|\mathbf{p}_i - \mathbf{p}\|^2}.$$

Justification for this approximation of  $\kappa_n$  is best explained with the help of Figure 6: the radius  $R_i$  of the osculating circle through  $\mathbf{p}$  and  $\mathbf{p}_i$  must be such that there is a right angle at vertex  $\mathbf{p}_i$ :

$$(\mathbf{p} - \mathbf{p}_i) \cdot (\mathbf{p} - \mathbf{p}_i - 2R_i\mathbf{n}) = 0.$$

Since  $\kappa_n^i$  is just the inverse of  $R_i$ , the approximation follows<sup>2</sup>. (Note that there is an error in the formula given by [Taubin 1995b].)

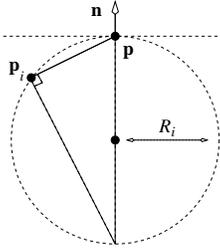


Fig. 6. Osculating circle for edge  $\mathbf{p}\mathbf{p}_i$ .

By construction,  $\mathbf{n}$  is an eigenvector of  $\mathcal{M}_{\mathbf{p}}$  associated with the eigenvalue 0. To compute the remaining eigenpairs,  $\mathcal{M}_{\mathbf{p}}$  is restricted to the tangent plane  $\langle \mathbf{n} \rangle^\perp$  using a Householder transformation and then diagonalized. Let  $\mathbf{i} = (1, 0, 0)^T$  be the first coordinate vector and let

$$\mathcal{W}_{\mathbf{p}} = \frac{\mathbf{i} + \varepsilon\mathbf{n}}{\|\mathbf{i} + \varepsilon\mathbf{n}\|}, \quad \mathcal{Q}_{\mathbf{p}} = \mathcal{I} - 2\mathcal{W}_{\mathbf{p}}\mathcal{W}_{\mathbf{p}}^T,$$

where  $\varepsilon = -1$  if  $\|\mathbf{i} - \mathbf{n}\| > \|\mathbf{i} + \mathbf{n}\|$ ,  $\varepsilon = 1$  otherwise.  $\mathcal{Q}_{\mathbf{p}}$  is the Householder matrix, an orthogonal matrix having  $\varepsilon\mathbf{n}$  as its first column. The other two columns ( $\tilde{\mathbf{e}}_1$  and  $\tilde{\mathbf{e}}_2$ ) define an orthonormal basis of the tangent space. Thus:

$$\mathcal{Q}_{\mathbf{p}}^T \mathcal{M}_{\mathbf{p}} \mathcal{Q}_{\mathbf{p}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \tilde{m}_1 & \tilde{m}_3 \\ 0 & \tilde{m}_3 & \tilde{m}_2 \end{pmatrix}.$$

The  $2 \times 2$  nonzero minor can be diagonalized, given an angle  $\eta$  such that

$$\mathbf{e}_1 = \cos \eta \tilde{\mathbf{e}}_1 - \sin \eta \tilde{\mathbf{e}}_2, \quad \mathbf{e}_2 = \sin \eta \tilde{\mathbf{e}}_1 + \cos \eta \tilde{\mathbf{e}}_2$$

<sup>2</sup>This formula leads to a straightforward calculation of  $H$  and  $K$  as follows [Watanabe and Belyaev 2000]. Applying a trapezoid approximation to Eq. (1) yields:

$$H = \frac{1}{2\pi} \int_0^{2\pi} \kappa_n(\theta) d\theta = \kappa_n\left(\frac{\theta_m + \theta_1}{2}\right) + \cdots + \kappa_n\left(\frac{\theta_{m-1} + \theta_m}{2}\right),$$

where  $\theta_i$  is the angle of the  $i$ -face at  $\mathbf{p}$ . This gives an estimate of  $H$ . A similar calculation leads to:

$$\frac{1}{2\pi} \int_0^{2\pi} \kappa_n^2(\theta) d\theta = \frac{3H^2 - K}{2}.$$

Applying again a trapezoid rule gives an estimate of  $K$ .

are the remaining eigenvectors of  $\mathcal{M}_{\mathbf{p}}$ , i.e., the estimates of the principal directions of the surface at  $\mathbf{p}$ . The principal curvatures are obtained from the two corresponding eigenvalues of  $\mathcal{M}_{\mathbf{p}}$  using Eq. (8).

Note that the above method, even though it was described for a surface triangulation, can be extended to a discrete set of points, as recently proposed by Gopi et al. [2000] in the context of surface reconstruction.

**3.3.4 Spatial averages and 1-ring patches.** Recently, Desbrun et al. [2000] have argued that the best way of extending to discrete meshes the definitions of differential quantities in the continuous case is by computing spatial averages around vertices. For instance, the average Gaussian curvature at vertex  $\mathbf{p}$  of  $T$  should be defined in its discrete form as:

$$K(\mathbf{p}) = \frac{1}{A} \iint_A K \, dA,$$

for  $A$  a properly chosen area around  $\mathbf{p}$ . The authors show that strong analogies between the continuous and the discrete case are obtained by restricting the averaging domain at vertex  $\mathbf{p}$  to a family of special regions  $A_M$  (called *finite volumes*) contained within the 1-ring neighborhood of  $\mathbf{p}$ , with piecewise-linear boundaries crossing the mesh edges at their midpoints. Two main types of finite volumes are used. In the first, the point inside each triangle of the 1-ring neighborhood of  $\mathbf{p}$  is the barycenter of the triangle (Figure 7.a). In the second, this inside point is the circumcenter of the triangle and the finite volume is recognized as the local Voronoi cell (Figure 7.b).

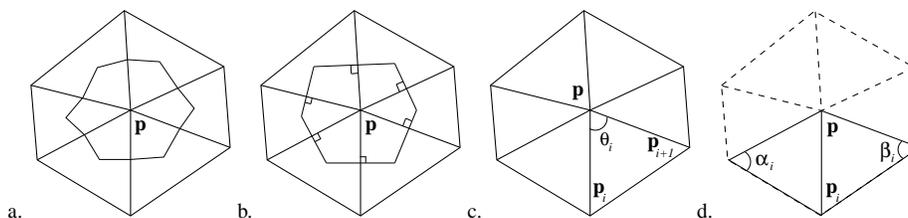


Fig. 7. Local regions around a vertex [Desbrun et al. 2000]. a. Finite volume region using barycentric cells. b. Local region using Voronoi cells. c. External angles of a Voronoi region. d. 1-ring neighbors and angles opposite to an edge.

A discrete equivalent of the Gaussian curvature is obtained by applying the Gauss-Bonnet theorem to the finite volume  $A_M$ . Roughly speaking, this theorem says that given a region  $R$  of a surface, the *integral curvature*  $\iint_R K \, dA$  measures the solid angle filled by all unit normals to  $R$ , translated to one point. In the discrete domain, this means that

$$\iint_{A_M} K \, dA = 2\pi - \sum_{\mathbf{p}_i \in N(\mathbf{p})} \theta_i,$$

where  $\theta_i$  is the angle of the  $i$ -th face at vertex  $\mathbf{p}$  (see Figure 7.c). This formula holds for any surface patch within the 1-ring neighborhood whose boundary crosses the edges at their midpoint. But one finite volume region must be chosen to provide an accurate estimate of the spatial average. Desbrun et al. [2000] show that the Voronoi cells provide provably

tight error bounds under mild assumptions of smoothness. If the 1-ring neighborhood of  $\mathbf{p}$  is made only of non-obtuse triangles, then the local patch has area

$$A_{\text{Voronoi}} = \frac{1}{8} \sum_{\mathbf{p}_i \in N(\mathbf{p})} (\cot \alpha_i + \cot \beta_i) \|\mathbf{p}_i - \mathbf{p}\|^2.$$

where  $\alpha_i$  and  $\beta_i$  are the two angles opposite to the edge  $\mathbf{p}\mathbf{p}_i$  as depicted in Figure 7.d. To ensure a perfect tiling of the triangulation  $T$  and an optimal accuracy in the presence of obtuse triangles in the 1-ring neighborhood, the authors advocate the use of *mixed areas*: if a triangle is obtuse, take its barycenter as inside point; otherwise, take its circumcenter. Denoting the area of the corresponding local patch by  $A_{\text{Mixed}}$ , an estimate of the Gaussian curvature at  $\mathbf{p}$  is:

$$K(\mathbf{p}) = \frac{1}{A_{\text{Mixed}}} \left( 2\pi - \sum_{\mathbf{p}_i \in N(\mathbf{p})} \theta_i \right).$$

Resorting to the Gauss-Bonnet theorem to find a discrete equivalent of the Gaussian curvature is not new and has been proposed in the past notably in [Lin and Perry 1982; Pinkall and Polthier 1993; Alboul and van Damme 1996]. For instance, Lin and Perry [1982] proposed to use the following as approximation of the Gaussian curvature:

$$K(\mathbf{p}) = \frac{3}{A} \left( 2\pi - \sum_{\mathbf{p}_i \in N(\mathbf{p})} \theta_i \right),$$

where  $A$  is the total area of the triangles in the 1-ring neighborhood of  $\mathbf{p}$ .

To compute estimates of the mean curvature and vertex normal at  $\mathbf{p}$ , the authors introduce the *Laplace-Beltrami operator*  $\mathcal{K}$ . On a piecewise-smooth surface  $S$ , this operator maps a point  $\mathbf{p}$  to the vector  $\mathcal{K}(\mathbf{p}) = 2H_{\mathbf{p}}\mathbf{n}_{\mathbf{p}}$ . On a triangulation  $T$ , the integral of the Laplace-Beltrami operator over the finite volume  $A_M$  can be transformed into a line integral over the boundary of the finite volume. Computing this line integral gives

$$\iint_{A_M} \mathcal{K} \, dA = \frac{1}{2} \sum_{\mathbf{p}_i \in N(\mathbf{p})} (\cot \alpha_i + \cot \beta_i) (\mathbf{p}_i - \mathbf{p}).$$

As before, this expression holds even for 1-ring neighborhoods with obtuse triangles. To provide an estimate of the Laplace-Beltrami operator at  $\mathbf{p}$ , the mixed area is chosen:

$$\mathcal{K}(\mathbf{p}) = \frac{1}{2A_{\text{Mixed}}} \sum_{\mathbf{p}_i \in N(\mathbf{p})} (\cot \alpha_i + \cot \beta_i) (\mathbf{p}_i - \mathbf{p}).$$

This vector, after normalization, provides a reasonable estimate of the vertex normal at  $\mathbf{p}$ . As for the mean curvature, it can be estimated by taking half of the magnitude of  $\mathcal{K}(\mathbf{p})$ . When the mean curvature at  $\mathbf{p}$  is zero, the vertex normal is obtained by averaging the 1-ring face normal vectors (as in § 3.1).

The mean quadrature has an interesting interpretation as a quadrature of the integral of Eq. (1):

$$H(\mathbf{p}) = \sum_{\mathbf{p}_i \in N(\mathbf{p})} w_i \kappa_n^i,$$

where  $\kappa_n^i$  is an estimate of the normal curvature along the edge  $\mathbf{p}\mathbf{p}_i$  (the same approximation used in [Taubin 1995b] in the estimation of the tensor of curvature) and the  $w_i$  are

weights which sum to one for each  $\mathbf{p}$  having no obtuse triangle in its 1-ring neighborhood:

$$\kappa_n^i = \frac{2\mathbf{n} \cdot (\mathbf{p}_i - \mathbf{p})}{\|\mathbf{p}_i - \mathbf{p}\|^2}, \quad w_i = \frac{1}{8A_{\text{Mixed}}} (\cot \alpha_i + \cot \beta_i) \|\mathbf{p}_i - \mathbf{p}\|^2.$$

To determine the principal directions, the idea is to use the  $\kappa_n^i$  as samples of the  $2 \times 2$  curvature tensor  $\mathcal{S}$ . In other words, the goal is to find  $\mathcal{S}$  satisfying

$$\mathbf{d}_i^T \mathcal{S} \mathbf{d}_i = \kappa_n^i, \quad i = 1, \dots, m,$$

where  $\mathbf{d}_i$  is the unit direction in the tangent plane of the edge  $\mathbf{p}\mathbf{p}_i$ :

$$\mathbf{d}_{i,j} = \frac{(\mathbf{p}_i - \mathbf{p}) - [(\mathbf{p}_i - \mathbf{p}) \cdot \mathbf{n}] \mathbf{n}}{\|(\mathbf{p}_i - \mathbf{p}) - [(\mathbf{p}_i - \mathbf{p}) \cdot \mathbf{n}] \mathbf{n}\|}.$$

A least-squares approximation of  $\mathcal{S}$  is found by minimizing the error  $E$

$$E(\mathcal{S}) = \sum_i w_i \left| \mathbf{d}_i^T \mathcal{S} \mathbf{d}_i - \kappa_n^i \right|,$$

subject to the constraints that the determinant of  $\mathcal{S}$  is  $K(\mathbf{p})$  and its trace is  $2H(\mathbf{p})$ . The eigenvectors of  $\mathcal{S}$  then give estimates of the principal directions at  $\mathbf{p}$ . An illustration is given in Figure 8.

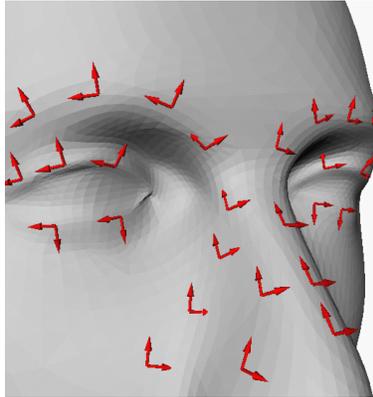


Fig. 8. Principal directions on a triangle mesh [Desbrun et al. 2000].

A different discrete formulation of the mean curvature is given in [Alboul and van Damme 1996]. Using Eq. (1) and the Euler formula, it is a simple matter to see that at any point of a piecewise-smooth surface  $S$  the mean curvature is the half-sum of the normal curvatures in any two orthogonal directions:

$$H = \frac{1}{2} (\kappa_n(\mathbf{t}) + \kappa_n(\mathbf{n} \times \mathbf{t})).$$

This formula can be applied to a triangulation  $T$ . For a point on an edge  $e$ , choose as one direction the direction along this edge and the orthogonal direction  $\perp_e$ . The curve defined by  $e$  is a straight line, so its curvature is zero. As for the curve defined by  $\perp_e$ , the equivalent of its curvature is the angle between the plane normals of the faces adjacent to

$e$ . The discrete mean curvature  $H(e)$  of  $e$  is thus one-half of this angle. For a domain  $U$ , the mean curvature is then:

$$H(U) = \sum_{\text{edge } e \subset U} H(e) \text{ length}(e \cap U).$$

The ratio of  $H(U)$  to the area of  $U$  gives an estimate of the average mean curvature over the part of  $T$  which corresponds to  $U$ . Using the notations above, and taking  $U = A_{\text{Mixed}}$ , this gives another estimate of the mean curvature at  $\mathbf{p}$ :

$$H(\mathbf{p}) = \frac{1}{4A_{\text{Mixed}}} \sum_{\mathbf{p}_i \in N(\mathbf{p})} \|\mathbf{p}_i - \mathbf{p}\| \varphi_i,$$

where  $\varphi_i$  is the dihedral angle between the two triangles meeting along the edge  $\mathbf{p}\mathbf{p}_i$ .

### 3.4 Covariance matrices

Many of the original methods for measuring surface normals and curvature at points of a range image used derivatives estimates which may not be robust under additive noise. In addition, the image considered may not have adequate ‘‘smoothness’’ to support the use of differential operators. This led several authors to explore an alternate signal processing basis for computing local shape measures, using covariance matrices [Liang and Todhunter 1990; Berkmann and Caelli 1994]. The main advantage of this covariance approach is that it provides ideal ways of treating signals embedded in additive Gaussian or white noise.

The covariance matrices method can be easily adapted to triangulations. Let again the vertices  $\mathbf{p}_i$  be the 1-ring neighbors of  $\mathbf{p}$ . Define the first-order,  $3 \times 3$ , surface covariance matrix at  $\mathbf{p}$  as:

$$C_I = \frac{1}{m} \sum_{i=1}^m (\mathbf{p}_i - \bar{\mathbf{p}}) (\mathbf{p}_i - \bar{\mathbf{p}})^T, \quad (9)$$

where  $\bar{\mathbf{p}} = \frac{1}{m} \sum_{i=1}^m \mathbf{p}_i$  is the mean position vector. This matrix may be seen as a discrete equivalent of the first fundamental form matrix  $\mathcal{G}$ . Two of its eigenvectors ( $\mathbf{t}_1$  and  $\mathbf{t}_2$ ) define the plane which minimizes, in the least-squares sense, the orthogonal distance from all points to that plane. As originally proposed in [Liang and Todhunter 1990], this plane is a reasonable approximation to the surface tangent plane at  $\mathbf{p}$ . Consequently, the other eigenvector forms the equivalent of the surface normal  $\mathbf{n}$ .

An analogous definition of the second fundamental form matrix follows by projecting the difference vector that points from  $\mathbf{p}$  to  $\mathbf{p}_i$  onto the ‘‘tangent plane’’ as determined by (9) and weighting the resulting vector by a measure of the orthogonal distance from point  $\mathbf{p}_i$  to the ‘‘tangent plane’’ [Berkmann and Caelli 1994]:

$$C_{II} = \frac{1}{m} \sum_{i=1}^m (\mathbf{y}_i - \bar{\mathbf{y}}) (\mathbf{y}_i - \bar{\mathbf{y}})^T,$$

where

$$\mathbf{y}_i = [(\mathbf{p}_i - \mathbf{p}) \cdot \mathbf{n}] \begin{pmatrix} (\mathbf{p}_i - \mathbf{p}) \cdot \mathbf{t}_1 \\ (\mathbf{p}_i - \mathbf{p}) \cdot \mathbf{t}_2 \end{pmatrix},$$

The eigenvectors of the  $2 \times 2$  matrix  $C_{II}$  can be considered as estimates of the principal directions at  $\mathbf{p}$ .

There are alternate ways of computing principal directions. They can for instance be estimated by calculating the covariance matrix of a certain normal map [Liang and Todhunter 1990]. Also, since they lie on the tangent plane anyway, they can be found as the eigenvectors of a  $2 \times 2$  covariance matrix  $C'_{II}$  built on the projections of the normal vectors, within a neighborhood of  $\mathbf{p}$ , onto the tangent plane [Berkmann and Caelli 1994].  $C'_{II}$  is defined as follows:

$$C'_{II} = \frac{1}{m} \sum_{i=1}^m (\mathbf{y}_i - \bar{\mathbf{y}}) (\mathbf{y}_i - \bar{\mathbf{y}})^T,$$

with

$$\mathbf{y}_i = \begin{pmatrix} \mathbf{n}_i \cdot \mathbf{t}_1 \\ \mathbf{n}_i \cdot \mathbf{t}_2 \end{pmatrix}$$

and  $\mathbf{n}_i$  the vertex normal estimated at  $\mathbf{p}_i$  with Eq. (9).

In the context of surface reconstruction from point clouds, related definitions were given in [Hoppe et al. 1992]. The tangent plane  $P$  associated with the data point  $\mathbf{p}$  is represented as a point  $\bar{\mathbf{p}}$ , called the center, together with a unit normal  $\mathbf{n}$ . Let  $N'(\mathbf{p})$  be the set of vertices within a certain distance of  $\bar{\mathbf{p}}$ . The center and unit normal are computed so that the plane  $P$  is the least-squares best fitting plane to  $N'(\mathbf{p})$ . In other words,  $\bar{\mathbf{p}}$  is taken to be the centroid of  $N'(\mathbf{p})$  and  $\mathbf{n}$  is determined using principal component analysis of the covariance matrix of  $N'(\mathbf{p})$ :

$$C'_I = \sum_{\mathbf{p}_i \in N'(\mathbf{p})} (\mathbf{p}_i - \bar{\mathbf{p}}) (\mathbf{p}_i - \bar{\mathbf{p}})^T.$$

Then  $\mathbf{n}$  is chosen to be (up to sign) the eigenvector corresponding to the smallest eigenvalue of  $C'_I$ . Another definition for the same mathematical object, though with a different formulation, is proposed in [Gopi et al. 2000].

Note that once vertex normals and principal curvatures are known, the normal curvatures  $\kappa_n^i$  in the directions of the 1-ring neighbors  $\mathbf{p}_i$  can be estimated with the formula of Taubin (§ 3.3.3) and the principal curvatures are found by solving the overdetermined system of linear equations

$$\kappa_n^i = \kappa_1 \cos^2 \theta_i + \kappa_2 \sin^2 \theta_i$$

obtained using the Euler formula, where  $\theta_i$  is the angle between  $\mathbf{e}_1$  and  $\mathbf{p}_i - \mathbf{p}$ .

Somewhat related to the covariance approach is the method proposed by Yoshimi and Tomita [1994]. Assume that the surface normals have been estimated at all vertices of the triangulation. Define a local region about a point  $\mathbf{p}$  (normal  $\mathbf{n}$ ) as being those neighboring points  $\mathbf{p}_i$  (normal  $\mathbf{n}_i$ ) at which the angle between  $\mathbf{n}$  and  $\mathbf{n}_i$  is less than a given constant. This region is generally a single closed region with an elliptic boundary, except when the Gaussian curvature at the point considered is zero. Consider the projection of this region onto the “tangent plane” at  $\mathbf{p}$ . The major and minor axes of the ellipse forming the boundary are the principal directions and the radii are related to the principal curvatures.

### 3.5 A word of conclusion

Despite the extensive use of piecewise-linear surfaces in computer graphics and the repeated need to estimate differential quantities, there is currently no consensus on the most appropriate way to approximate such simple geometric attributes as normals and principal

curvatures on discrete surfaces. No systematic comparison has been made between the most popular methods for computing local Darboux frames at vertices of triangle meshes.

However, some interesting experiments can help us draw partial conclusions. McIvor and Valkenburg [1997] compare several methods for approximating the Darboux frames at points of a range image: finite differences (surface derivatives are estimated in terms of differences in depth between neighboring pixels), facet-based estimation (independent fitting of low-order functions to the 3 components of position in a small neighborhood of each data point, from which the differential properties of the surface can be computed – see [Lee et al. 1993; McIvor 1998]) and quadric surface fitting. The first two methods are of little interest to us: they rely heavily on the natural parameterization of range images given by the grid on which the points are aligned and are not applicable to general piecewise-linear surfaces. But surface fitting is a different story. According to the authors, the non-linear quadric fitting method (see § 3.2) has the best performance of all tested methods and also the greatest computational cost. As for the linear fitting methods, the number of terms used in the quadric makes little difference to the curvature estimate performance, although having first- and zero-order terms improves the surface normal estimates.

Other researchers have made empirical analyses of the performance of curvature estimation techniques [Flynn and Jain 1989; Trucco and Fisher 1995; Tang and Medioni 1999]. The overall common conclusion of their experiments is that qualitative properties (e.g. sign of Gaussian curvature) can be more reliably estimated than quantitative ones (e.g. curvature magnitude). Since qualitative information about the curvature field is more important anyway than quantitative ones for segmentation purposes, people have looked at methods for directly computing the curvature signs without computing their magnitude. For example, Angelopoulou and Wölff [1998] compute the sign of the Gaussian curvature, without surface fitting, local derivative computation, nor normal recovery, by checking the relative orientation of two simple closed curves.

Despite these very partial views, it seems that the best local geometry estimation techniques are those that are direct analogues in the discrete setting of formulas in the continuous case. The methods advocated in [Taubin 1995b; Desbrun et al. 2000] stand out in this respect. Further experiments are needed to back up this intuitive claim.

#### 4. DIFFERENTIAL PARAMETERS ESTIMATES IMPROVEMENT

Polyhedral surfaces extracted from volumetric data by isosurface construction algorithms or those resulting from laser range scanners may contain a good deal of noise and small-scale oscillations. These undesirable features can severely affect the estimation of differential properties and thus lead to poor segmentation and shape recovery. It is thus important to smooth out the high frequency details of noisy meshes while retaining the low frequency components. Ridding a mesh of its unnecessary details is known as *discrete fairing*.

But even if a mesh is “fair” enough, the accurate estimation of the principal directions and curvatures of the subjacent surface is still a difficult task. Differential parameters estimation is analogous to feature detection in conventional 2D images and suffers from the same problems, namely the sensitivity of local operations to noise and quantization [Hilton et al. 1995]. Better results can be obtained, for instance, by approximating the surface at  $p$  over a larger window or by extending the spatial averages to the 2-ring neighborhood. But this tends to smooth the results, and one of its effects is that the estimated curvatures will be of smaller magnitude than the actual curvatures. The situation is even worse for

directional properties (principal directions) which are difficult to robustly estimate without considering more global approaches to feature recovery. The *curvature consistency* method is one such global approach.

This section successively delves into discrete fairing (§ 4.1) and curvature consistency (§ 4.2).

#### 4.1 Surface fairing and mesh denoising

Smoothness refers to the mathematical notion of continuous differentiability and smoothing enforces a continuity of curvature. By contrast, fairness is a measure of aestheticism, of “well-shapedness” and fairing imposes a low variation of curvature. Classical fairing techniques use constrained energy minimization. For a surface  $S$ , the following fairness functional is frequently used:

$$E(S) = \iint_S (\kappa_1^2 + \kappa_2^2) \, dA.$$

In practice, even though this energy can be estimated on discrete meshes [Welch and Witkin 1994], the non-linear dependence of the principal curvatures on  $S$  leads people to work with the membrane  $E_m$  and thin-plane  $E_t$  functionals:

$$E_m(S) = \iint_S (\mathbf{X}_u^2 + \mathbf{X}_v^2) \, du \, dv, \quad E_t(S) = \iint_S (\mathbf{X}_{uu}^2 + 2\mathbf{X}_{uv}^2 + \mathbf{X}_{vv}^2) \, du \, dv.$$

Their variational derivatives correspond to the Laplacian  $\mathcal{L}(\mathbf{X}) = \mathbf{X}_{uu} + \mathbf{X}_{vv}$  and second Laplacian  $\mathcal{L}^2(\mathbf{X}) = \mathcal{L} \circ \mathcal{L}(\mathbf{X})$ . Fairing  $S$  is then achieved by integrating over time the diffusion equation:

$$\frac{\partial \mathbf{X}}{\partial t} = \lambda \mathcal{L}(\mathbf{X}), \quad (10)$$

where  $\lambda$  is a small positive constant. The diffusion flow allows to smooth the high frequencies in noisy meshes.

Now back to discrete meshes. At each vertex  $\mathbf{p}$  of  $T$ , the Laplacian can be linearly approximated by the *umbrella operator*:

$$\mathcal{U}(\mathbf{p}) = \frac{\sum_i w_i \mathbf{p}_i}{\sum_i w_i} - \mathbf{p},$$

where the summation is over the 1-ring neighborhood vertices of  $\mathbf{p}$  and the  $w_i$  are positive weights. Integrating the diffusion equation (10) amounts in this discrete setting to construct a sequence of meshes  $T^{(j)}$  with the local update rule:

$$\mathbf{p}^{(j+1)} \leftarrow \mathbf{p}^{(j)} + \lambda \mathcal{U}(\mathbf{p}^{(j)}),$$

for a discrete time step  $\Delta t = 1$ . This process is called *Laplacian smoothing*. It recursively moves each vertex of the mesh by a displacement equal to a positive scale factor times the average of the neighboring vertices. Possible choices for the weights  $w_i$  are  $w_i = 1$  (the corresponding umbrella operator is denoted  $\mathcal{U}_0$ ) or a function of the length of the edge  $\mathbf{p}\mathbf{p}_i$ ,  $w_i = \|\mathbf{p}_i - \mathbf{p}\|^\alpha$  ( $\mathcal{U}_1$ ).  $\alpha = -1$  produces good results. Laplacian smoothing is linear in both time and space for each filtering pass. But if  $\lambda$  is not small enough, ripples appear on the surface. This restriction means that for very large meshes, hundreds of iterations are

needed to produce a noticeable smoothing. In addition, the Laplacian flow may introduce unnatural deformations if the initial mesh is not regularly sampled.

People have advocated using a weighted average of the derivatives  $\mathcal{L}$  and  $\mathcal{L}^2$  to improve this basic scheme. By generalizing classical discrete Fourier analysis to two-dimensional discrete surface signals, Taubin [1995a; 1995c] shows that the combination  $(a+b)\mathcal{L} - ab\mathcal{L}^2$  can provide a Gaussian filtering that minimizes shrinkage and unnatural deformations. In particular, he proposes a weighted Laplacian smoothing flow of this form with two scale factors of opposite signs, the negative factor having the larger magnitude, which in the discrete setting translates to the following update rule:

$$\mathbf{p}^{(j+1)} \leftarrow \mathbf{p}^{(j)} - (\mu - \lambda)\mathcal{U}(\mathbf{p}^{(j)}) - \mu\lambda\mathcal{U}^2(\mathbf{p}^{(j)}),$$

where  $\mu > \lambda > 0$  and  $\mathcal{U}^2$  is the squared umbrella operator

$$\mathcal{U}^2(\mathbf{p}) = \frac{\sum_i w_i \mathcal{U}(\mathbf{p}_i)}{\sum_i w_i} - \mathcal{U}(\mathbf{p}).$$

The best smoothing results are obtained with the  $\mathcal{U}_0$  operator.

If this scheme, which is related to that of [Kobbelt 1997], improves on the result of the basic Laplacian smoothing flow, it still has a number of undesirable features, notably its lack of local shape control and the fact that it smoothes all small-scale features [Ohtake et al. 2000]. People have argued that the umbrella operator is not an adequate approximation of the Laplacian: in the case of meshes with irregular sampling, it may lead to geometric distortion during smoothing and numerical instability. Desbrun et al. [1999] advocate instead the use of the *mean curvature flow*. In the continuous case, it smoothes the surface by moving along the surface normal with a speed equal to the mean curvature:

$$\frac{\partial \mathbf{p}}{\partial t} = -H(\mathbf{p})\mathbf{n}(\mathbf{p}).$$

Going discrete, this gives the following local update rule:

$$\mathbf{p}^{(j+1)} \leftarrow \mathbf{p}^{(j)} - H(\mathbf{p}^{(j)})\mathbf{n}(\mathbf{p}^{(j)}).$$

The initial mesh  $T$  may represent a surface with sharp edges. If isotropic denoising is used, then the underlying geometry will be lost. A good smoothing scheme is one that preserves clear features like object boundaries and smoothes homogeneous regions. To keep important features intact, noise should only be directionally diffused. Since the presence of edges can be detected using estimates of the principal curvatures. Desbrun et al. [2000] propose to use a modified update rule of the mean curvature flow:

$$\mathbf{p}^{(j+1)} \leftarrow \mathbf{p}^{(j)} - wH(\mathbf{p}^{(j)})\mathbf{n}(\mathbf{p}^{(j)}),$$

where  $w$  is a smoothing weight defined as follows:

$$w = \begin{cases} 1 & \text{if } |\kappa_1| \leq \tau \text{ and } |\kappa_2| \leq \tau, \\ 0 & \text{if } |\kappa_1| > \tau \text{ and } |\kappa_2| > \tau \text{ and } K > 0, \\ \kappa_1/H & \text{if } |\kappa_1| = \min(|\kappa_1|, |\kappa_2|, |H|), \\ \kappa_2/H & \text{if } |\kappa_2| = \min(|\kappa_1|, |\kappa_2|, |H|), \\ 1 & \text{if } |H| = \min(|\kappa_1|, |\kappa_2|, |H|), \end{cases}$$

where  $\tau$  is a user-defined parameter. Applying this procedure to a noisy cube gives the result shown on Figure 9. For more complicated objects, a pass of curve smoothing is added to better straighten the edges.

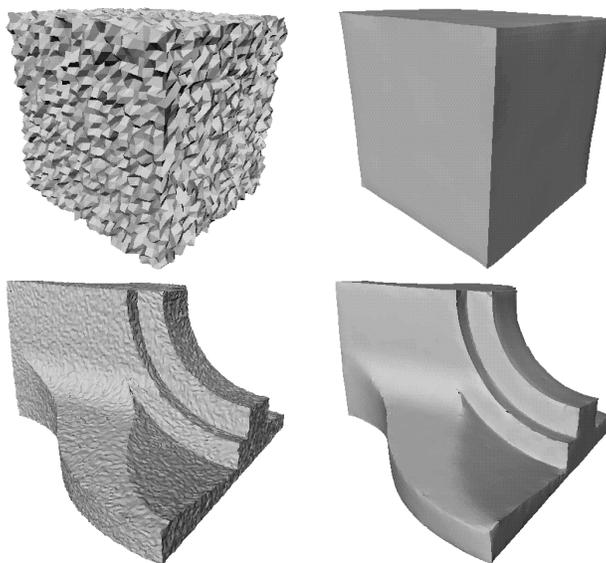


Fig. 9. Feature-preserving denoising using anisotropic smoothing [Desbrun et al. 2000].

Even though the mean curvature flow produces better results than Laplacian smoothing and is relatively independent of the mesh sampling rate, it increases the mesh irregularity and has no mechanism for slowing down the smoothing as time increases. In other words, it may lead to oversmoothing. Recently, Ohtake et al. [2000] have proposed new simple and effective mesh smoothing methods which combine the best properties of the Laplacian and mean curvature flows while reducing possible oversmoothing. The key idea underlying these new schemes is to use a normal speed component for smoothing and a tangent speed component to improve the mesh sampling rate. One possible local update rule is:

$$\mathbf{p}^{(j+1)} \leftarrow \mathbf{p}^{(j)} + \lambda \left( H(\mathbf{p}^{(j)}) \mathbf{n}(\mathbf{p}^{(j)}) + C \left[ \mathcal{U}_0(\mathbf{p}^{(j)}) - (\mathcal{U}_0(\mathbf{p}^{(j)}) \cdot \mathbf{n}(\mathbf{p}^{(j)})) \mathcal{U}_0(\mathbf{p}^{(j)}) \right] \right),$$

where  $C$  is either a positive constant or a function of surface curvatures to achieve a higher mesh sampling in curved surface regions. Note that the tangential component is a function of the projection  $\mathcal{U}_0 - (\mathcal{U}_0 \cdot \mathbf{n})\mathcal{U}_0$  of the umbrella vector on the tangent plane at  $\mathbf{p}$  defined by  $\mathbf{n}$ .

This scheme produces better results than the mean curvature flow and the method of Taubin. But even better results are obtained as follows. Define  $\mathbf{m} = \mathcal{U}_0 / \|\mathcal{U}_0\|$  and let  $\delta$  be the angle between the mean curvature vector  $H\mathbf{n}$  and  $\mathbf{m}$ . The idea is then to move the vertices in the direction of  $\mathbf{m}$  in such a way that the normal speed component is equal to the mean curvature. In other words, the local update rule is

$$\mathbf{p}^{(j+1)} \leftarrow \mathbf{p}^{(j)} + \lambda \mathcal{F}(\mathbf{p}^{(j)}),$$

where

$$\mathcal{F} = \begin{cases} \frac{|H|}{\cos \delta} \mathbf{m} & \text{if } \cos \delta > \epsilon, \\ 2H\mathbf{n} - \frac{|H|}{\cos \delta} \mathbf{m} & \text{if } \cos \delta < -\epsilon, \\ 0 & \text{if } |\cos \delta| \leq \epsilon. \end{cases}$$

$\epsilon = 0.1$  gives good results independently of the mesh sampling rate. A further refinement allows for the smoothing to slow down automatically, thus avoiding oversmoothing and the destruction of small-scale surface features.

Two examples showing the advantages of the new smoothing schemes over past methods are displayed in Figs. 10 and 11.

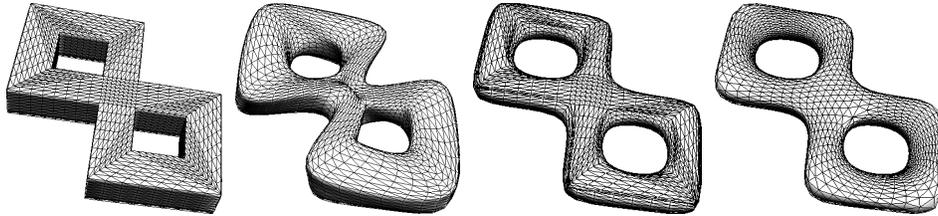


Fig. 10. From left to right: a pretzel-like shape consisting of parts with different sampling rates, and best smoothing with the Taubin algorithm (which substantially deforms the shape), the mean curvature flow (which produces an irregular mesh) the combined method of [Ohtake et al. 2000].

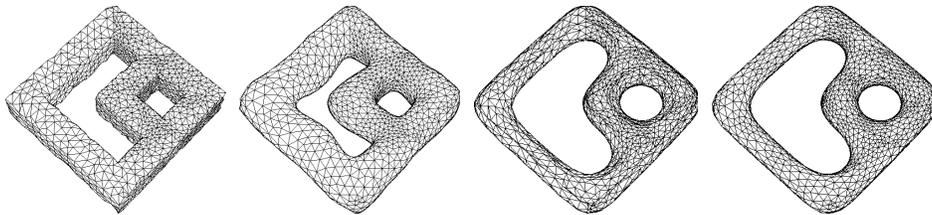


Fig. 11. From left to right: a two-holed polyhedral torus, and best smoothing with the Taubin scheme, the mean curvature flow and the new method of [Ohtake et al. 2001].

Ohtake et al. [2000] also introduce modifications of these schemes to better retain sharp edges. This idea is to smooth the face normals and then to move vertices based on the smoothed normals. Smoothing of normals is achieved by a weighted averaging of neighboring normals, with large weights if the normals are close and small weights if they are different. An example is shown on Figure 12.

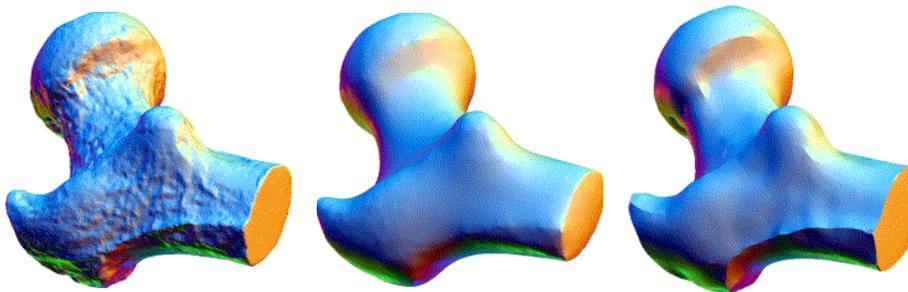


Fig. 12. From left to right: a mesh, a common smoothing and a feature-preserving smoothing [Ohtake et al. 2001].

## 4.2 Curvature consistency

Curvature consistency refers to a class of algorithms for improving differential parameters estimates originally developed in [Sander and Zucker 1990] for surface reconstruction in 3D voxel-based images and later applied to range images [Ferrie et al. 1993; Lagarde 1997]. It can be viewed as a second, more global, stage of processing applied to a set of augmented Darboux frames  $\Delta_{\mathbf{p}}$  estimated by local methods. The general idea is to smooth the surface while preserving the local structure described by the Darboux frames. This problem has a variational formulation: the objective is to minimize a functional form related to a minimum variation of curvature.

**4.2.1 Variational formulation.** In what follows, let  $\mathbf{p}$  be a vertex of  $T$  and  $\mathbf{p}_i$  the vertices of its 1-ring neighborhood. There are three main ingredients to the curvature consistency problem formulation [Ferrie et al. 1993; Ferrie et al. 1993]:

- A local surface estimation  $\Delta_{\mathbf{p}}$  at each vertex of  $T$ .
- A *transport model* which describes how  $\Delta_{\mathbf{p}_i}$  changes as it is transported to  $\mathbf{p}$  and vice-versa.
- *Update functionals* which describe how the Darboux frame at  $\mathbf{p}$  is updated so as to be consistent with the Darboux frames of its neighbors once they have been transported from  $\mathbf{p}_i$  to  $\mathbf{p}$  by the transport mechanism.

The transport model incorporates the constraint of minimum curvature variation, the update functionals enforce it.

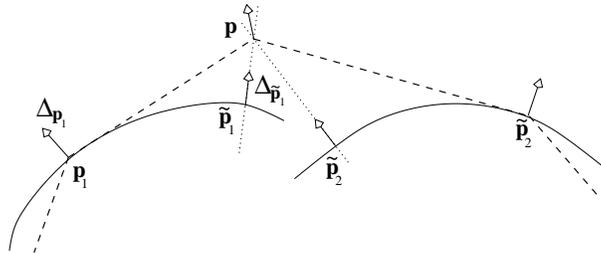


Fig. 13. Transporting  $\Delta_{\mathbf{p}_i}$  from  $\mathbf{p}_i$  to  $\mathbf{p}$ . The transport model used is a parabolic quadric.

Transporting the frame  $\Delta_{\mathbf{p}_i}$  from  $\mathbf{p}_i$  to  $\mathbf{p}$  involves extrapolation along a surface model  $S_{\mathbf{p}_i}$  that implements the desired constraint of locally constant curvature (Figure 13). The resulting frame at  $\mathbf{p}$  is denoted  $\Delta_{\tilde{\mathbf{p}}_i}$  and is an estimate of what the surface at  $\mathbf{p}$  should look like according to the description at  $\mathbf{p}_i$  under the transport constraint. There are many possibilities for a transport model, the only requirement being that the surface  $S_{\mathbf{p}_i}$  embeds the constant curvature constraint along an arc joining  $\mathbf{p}$  and  $\mathbf{p}_i$ . For relatively dense samplings, approximation by a parabolic quadric or even a plane is sufficient (see, e.g., [Krsek et al. 1997]).

The computational procedure required to transport each  $\Delta_{\mathbf{p}_i}$  to  $\mathbf{p}$  is as follows. Let  $\tilde{\mathbf{p}}_i$  be the projection of  $\mathbf{p}$  onto the transport surface  $S_{\mathbf{p}_i}$  associated to  $\mathbf{p}_i$ . (Projection may be either perpendicular or along the normal to  $\mathbf{p}$  [Sander and Zucker 1990].) Then an initial estimate of  $\Delta_{\mathbf{p}}$ ,  $\Delta_{\tilde{\mathbf{p}}_i}$ , is obtained from  $\Delta_{\mathbf{p}_i}$  by extrapolating along  $S_{\mathbf{p}_i}$ , i.e. by moving along  $S_{\mathbf{p}_i}$  in the direction of  $\mathbf{p}$ .

Let  $\Delta_{\mathbf{p}} = (\mathbf{p}, \mathbf{e}_1, \mathbf{e}_2, \mathbf{n}, \kappa_1, \kappa_2)$  and  $\Delta_{\tilde{\mathbf{p}}_i} = (\tilde{\mathbf{p}}_i, \tilde{\mathbf{e}}_{1_i}, \tilde{\mathbf{e}}_{2_i}, \tilde{\mathbf{n}}_i, \tilde{\kappa}_{1_i}, \tilde{\kappa}_{2_i})$ . Given the set of Darboux frames  $\Delta_{\tilde{\mathbf{p}}_i}$  determined by the transport model, the task is to compute a maximum likelihood estimate of  $\Delta_{\mathbf{p}}$  that minimizes variation subject to the constraint that  $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{n})$  is an orthonormal frame. As shown in [Sander and Zucker 1990], the minimization consists of two terms corresponding to

1. the surface normal  $\mathbf{n}$  and the principal curvatures  $\kappa_1$  and  $\kappa_2$ ,
2. the principal direction  $\mathbf{e}_1$ .

The remaining constraints are satisfied by computing  $\mathbf{e}_2$  as  $\mathbf{n} \times \mathbf{e}_1$ .

The first term of the minimization is:

$$E_1 = \min \left[ \sum_{i=1}^m (\|\mathbf{n} - \tilde{\mathbf{n}}_i\|^2 + (\kappa_1 - \tilde{\kappa}_{1_i})^2 + (\kappa_2 - \tilde{\kappa}_{2_i})^2 + \lambda(\mathbf{n} \cdot \mathbf{n} - 1)) \right],$$

where  $\lambda$  is a Lagrange multiplier needed to enforce the required condition on  $\mathbf{n}$ . This leads to the following updating functionals for the principal curvatures and the normal, where  $\tilde{\mathbf{n}}_i^{(j)} = (x_i^{(j)}, y_i^{(j)}, z_i^{(j)})$ :

$$\mathbf{n}^{(j+1)} = \frac{\left( \sum_i x_i^{(j)}, \sum_i y_i^{(j)}, \sum_i z_i^{(j)} \right)^T}{\sqrt{\left( \sum_i x_i^{(j)} \right)^2 + \left( \sum_i y_i^{(j)} \right)^2 + \left( \sum_i z_i^{(j)} \right)^2}}, \quad (11)$$

$$\kappa_1^{(j+1)} = \frac{1}{m} \sum_i \tilde{\kappa}_{1_i}^{(j)}, \quad \kappa_2^{(j+1)} = \frac{1}{m} \sum_i \tilde{\kappa}_{2_i}^{(j)},$$

where the superscript  $j$  refers to the current iteration step.

The second term minimizes the difference of principal directions in the tangent plane at  $\mathbf{p}$  [Ferrie et al. 1993]:

$$E_2 = \min_{\eta} \left[ \sum_{i=1}^n (1 - \mathbf{e}_1(\eta) \cdot \tilde{\mathbf{e}}_{1_i}) \right],$$

where  $\mathbf{e}_1$  is expressed in tangent plane coordinates:

$$\mathbf{e}_1(\eta) = \mathbf{a}_1 \cos \eta + \mathbf{a}_2 \sin \eta, \quad (\mathbf{a}_1, \mathbf{a}_2) \text{ orthonormal frame of tangent plane } \Gamma_{\mathbf{p}}. \quad (12)$$

As before, this leads to the following updating functional for  $\eta$ :

$$\eta^{(j+1)} = \tan^{-1} \left( \frac{A_{22}^{(j)} - A_{11}^{(j)} + \sqrt{\left( A_{11}^{(j)} - A_{22}^{(j)} \right)^2 + 4A_{12}^{(j)2}}}{2A_{12}^{(j)}} \right), \quad (13)$$

$$A_{pq}^{(j)} = \sum_{i=1}^m (\tilde{\mathbf{e}}_{1_i}^{(j)} \cdot \mathbf{a}_p)(\tilde{\mathbf{e}}_{1_i}^{(j)} \cdot \mathbf{a}_q).$$

$\tilde{\mathbf{e}}_{1_i}^{(j+1)}$  is then found by substituting the value of  $\eta^{(j+1)}$  of Eq. (13) back into Eq. (12).

The resulting curvature consistency algorithm works as follows. Compute initial estimates  $\Delta_{\mathbf{p}_j}^{(0)} = \Delta_{\mathbf{p}_j}$  for every vertex  $\mathbf{p}$  of the triangulation. With the updating functionals (11) and (13), compute  $\Delta_{\mathbf{p}^{(1)}}$  for every  $\mathbf{p}$ . Project  $\mathbf{p}$  onto the new transport surfaces of its neighbors  $\mathbf{q}_j$ , giving a new set of Darboux frames  $\Delta_{\mathbf{p}_j}^{(1)}$  and then repeat the operation.

The problem now is to determine at which point to terminate the iteration. Indeed, going too far will result in a surface with uniform curvature properties. Let  $E_{1\mathbf{p}}^{(i)}$  (resp.  $E_{2\mathbf{p}}^{(i)}$ ) be the value of the functional  $E_1$  (resp.  $E_2$ ) for point  $\mathbf{p}$  at iteration step  $i$ . Control over iteration is achieved by tracking the convergence over the following measure (see [Sander and Zucker 1990] for a discussion of convergence properties):

$$R_S^{(i)} = \sum_{\mathbf{p} \text{ vertex of } T} \left( E_{1\mathbf{p}}^{(i)} + E_{2\mathbf{p}}^{(i)} \right).$$

The algorithm is allowed to iterate until the difference  $|R_S^{(i)} - R_S^{(i-1)}|$  falls below a specified threshold. As reported in [Ferrie et al. 1993], stable results are generally realized within 5 iterations.

With this curvature consistency scheme, feature recovery is significantly enhanced for a minimal amount of smoothing. The characteristics of the  $H$ - $K$  map after refinement are consistent with the smooth nature of the surface. In addition, whereas the principal direction fields can be almost random before application of the curvature consistency algorithm, their structure are now correctly recovered.

**4.2.2 Correct localization of discontinuities.** The curvature consistency algorithm presented above corrects many of the deficiencies of the local methods for differential parameters estimation, but does not preserve the structure of surface discontinuities. Indeed, it tends to smooth the parameters in the vicinity of edge points. This is a major shortcoming with respect to feature localization and segmentation (Section 5).

Smoothing over discontinuities happens because there is no mechanism in the transport model for inhibiting propagation across  $\mathbf{p}$  and  $\mathbf{p}_i$  in the event of a discontinuity. What is needed is a modification of the updating functionals to weight contributions from the surrounding neighborhood according to how well these frames support a model of local continuity [Ferrie et al. 1993; Mathur and Ferrie 1997]. The idea is to embed a mechanism in the updating procedure that can characterize whether a point is part of a structural discontinuity or a noise point.

In the above algorithm, measurements from all neighbors of  $\mathbf{p}$  are averaged so as to give an estimate of the Darboux frame at  $\mathbf{p}$  at each iteration  $j$ :

$$\Delta_{\mathbf{p}}^{(j)} = \sum_{i=1}^m \lambda_i^{(j)} \Delta_{\mathbf{p}_i}^{(j)}, \quad \lambda_i^{(j)} = \frac{1}{m}. \quad (14)$$

How can measurements from the  $\mathbf{p}_i$  be combined in an optimal fashion in the sense of information fusion? In other words, what values of  $\lambda_i$  provide an optimal estimate of  $\Delta_{\mathbf{p}}$ ?

The idea, at each iteration  $j$ , is to predict what the estimation error  $\epsilon_i^{(j)}$  for each neighbor of  $\mathbf{p}$  would be if the basic averaging formula (14) was used and then compute the new estimate of  $\Delta_{\mathbf{p}}$  with weighting functions giving higher weights to neighbors with low error variances. In addition, these functions should provide similar weightings to neighbors which are correlated in their measurement and be bounded as the error variance approaches zero to prevent a single neighbor from taking over the process.

Weighting functions which encapsulate these desired properties are given by:

$$\lambda_i^{(j)} = \frac{W_i^{(j)}}{\sum_{i=1}^m W_i^{(j)}}, \quad W_i^{(j)} = e^{-\frac{\epsilon_i^{(j)2}}{\gamma^{(j)}}},$$

where  $\hat{\sigma}_j^{(i)^2}$  is an estimate of the error variance at iteration  $j$

$$\hat{\sigma}_i^{(j)^2} = \sum_{k=1}^j (\epsilon_i^{(k)})^2$$

and  $\gamma^{(j)}$  is a smoothing control parameter which can be set by taking the mean of the error variance values of all neighbors:

$$\gamma^{(j)} = \frac{2}{m} \sum_{i=1}^m \hat{\sigma}_i^{(j)^2}.$$

The estimation error  $\epsilon_i^{(j)}$ , which is a measure of the difference between  $\Delta_{\mathbf{p}}^{(j)}$  as given by Eq. (14) and  $\Delta_{\tilde{\mathbf{p}}_i}^{(j)}$ , can be computed in the spirit of the minimization terms  $E_1$  and  $E_2$ :

$$\epsilon_i^{(j)} = \|\mathbf{n}^{(j)} - \tilde{\mathbf{n}}_i^{(j)}\|^2 + \|\mathbf{e}_1^{(j)} - \tilde{\mathbf{e}}_{1_i}^{(j)}\|^2 + (\kappa_1^{(j)} - \tilde{\kappa}_{1_i}^{(j)})^2 + (\kappa_2^{(j)} - \tilde{\kappa}_{2_i}^{(j)})^2.$$

At the beginning of the curvature consistency iterations, the prediction error variance values are not available. Since there is no prior reason to believe that one neighbor would provide more accurate measurements of  $\Delta_{\mathbf{p}}$  than the others, each neighbor should be assigned equal weighting in the measurement update equation, i.e.  $\lambda_i^{(0)} = 1/m$ . At first, full variational relaxation takes place, smoothing out the random noise and starting to distort the discontinuities. But as the iterations progress, the real nature of the local surface is learned. After a few iterations the discontinuity localization process starts playing a major role. In subsequent iterations, the continuity constraint is applied only between neighbors believed to be on the same continuous surface and is inhibited between neighbors believed to be across discontinuities.

Edges formed by ramps and jump discontinuities are now correctly preserved.

## 5. SEGMENTATION

Segmentation is a vast and complex domain, both in terms of problem formulation and resolution techniques. For a human operator, it is fairly easy to identify regions of a surface that look like some simple geometric shape like a sphere, a cylinder or a cone. The task is much harder for a computer. Segmenting a range image or a triangulated mesh consists in formally translating the delicate visual notions of homogeneity and similarity, and defining criteria which allow their efficient implementation. Segmentation is a combinatorial problem which, in this sense, is closely related to parts decomposition. An example is displayed in Figure 14.

Originally, segmentation was applied to grey-level 2D images. The goal was to distinguish between the different parts of a digital image corresponding to different entities, in the physical and semantical sense of the application envisioned. Homogeneity was then typically defined as a function of the difference of intensity between neighboring pixels. With the introduction of new sensors, like the laser range finder, it became clear that segmentation extended way beyond ordinary 2D images. Similar principles apply to other kinds of data, with different interpretations of the notion of global homogeneity of a region [Besl and Jain 1988]. For range images, homogeneity is defined in geometrical and topological terms, often as a function of curvature measures.

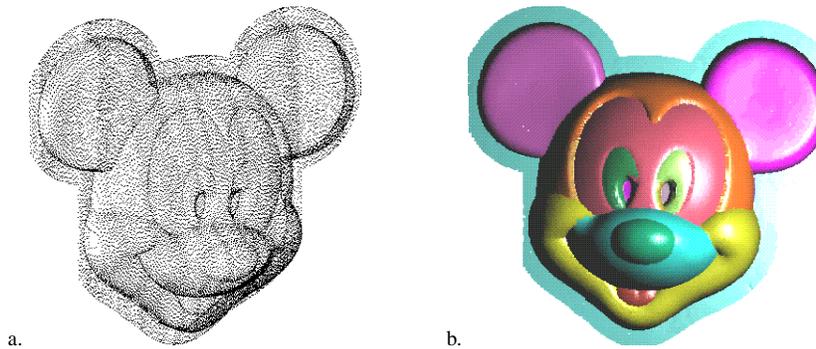


Fig. 14. Example of segmentation. a. An unstructured 3D point set. b. A segmentation of the point set [Chaine et al. 1999].

With few exceptions, almost all known algorithms for segmenting 3D data apply to partial range images. Some of them can however be extended to general triangular meshes or full range descriptions, with only minor adjustments.

This section reviews the segmentation literature applicable to the recovery of quadric surfaces in triangulated data. After an introduction on the different possible segmentation strategies (§ 5.1), we examine the several kinds of distinguished points and lines on a triangle mesh that can help guide its segmentation in primitive patches (§ 5.2). These features are defined in terms of the local curvature estimates obtained in Section 3 and refined in Section 4. We then describe some of the known methods for extracting specific types of quadrics (§ 5.3), focusing on the so-called *natural quadrics* (planes, spheres, cylinders and cones). Finally, we examine more general segmentation strategies (§ 5.4).

### 5.1 Basic segmentation strategies

Dividing a range image or a triangular mesh into regions according to shape-change detection has been a long-standing research problem. Past approaches fall into two main categories.

*Edge-based* approaches work by first identifying features corresponding to part boundaries and then interpolating to form smooth boundaries. Extracted boundaries are then used to guide the segmentation process. A homogeneity measure is needed to assert whether the regions cut out by the boundary lines are globally homogeneous or not and to split the non-homogeneous ones in smaller pieces.

*Region-based* approaches take the dual path and try to group vertices into regions corresponding to the same subjacent primitive surfaces, the boundaries being derived by computations on these surfaces. There are three popular approaches to region-based segmentation:

- *Split-and-merge* [Faugeras et al. 1983; Oshima and Shirai 1983; Medioni and Parvin 1986; Hoffman and Jain 1987] is a top-down method which recursively subdivides a range image until only globally homogeneous pieces remain. Since subdivision introduces artificial boundaries, adjacent regions are merged in a second stage of processing if their union satisfies the homogeneity criterion.
- *Region growing* [Besl and Jain 1988; Abdelmalek 1990; Sapidis and Besl 1995] is a bottom-up approach which involves starting with seed points (chosen either randomly

or using geometric criteria) and aggregating adjacent points to seed regions with similar local properties. When no more region can be grown, adjacent regions are merged if their union satisfies some homogeneity criterion.

- Algorithms based on *clustering* [Jain and Hoffman 1988; Lee et al. 1998; Köster and Spann 2000] estimate surface parameters on small patches and accumulate the parameters in a histogram, the large peaks of which correspond to instances of the surface in the data. Decision for region merging is typically based on the application of a statistical test to obtain a similarity measure.

Both classes of approaches have their strengths and weaknesses and usually need extensive post-processing phases. Edge-based methods tend to produce gaps in the boundaries of the regions. Indeed, the interpolation problem becomes very difficult when features are sparse. On the other hand, region-based methods can be easily perturbed in the presence of noise and generate boundaries which are connected but generally distorted. In addition, finding suitable initial seeds may be non-trivial.

In practical situations, each strategy taken separately is often insufficient for lack of robustness, non-uniqueness or complexity. Thus, a cooperation between edge-based and region-based techniques (hybrid approach) is often needed to ensure a reliable and robust segmentation [Yokoya and Levine 1989; Lejeune and Ferrie 1996].

## 5.2 Feature lines and initial segmentation

Several authors have noted that region-growing is an appropriate paradigm for segmenting general triangulated surfaces [Fisher et al. 1997; Sacchi et al. 1999; Robertson et al. 1999]. But even when region-based segmentation is used, some knowledge about patch boundaries is needed, to obtain a rough initial segmentation from which seed regions are grown. Ponce and Brady [1987] compute for instance a *surface primal sketch* from lines of discontinuity on height surfaces defined by range maps.

For triangulated surfaces, the first group of features to identify are folds, i.e. regions of high principal curvature, which are characteristic of part boundaries<sup>3</sup>. Among folds are discontinuities in the surface normal, i.e. sharp edges, which manifest themselves as high curvature regions due to the sampling process. Many methods are known for extracting folds. Yang and Lee [1999] propose for instance a two-step edge identification process. In the first step, the computation area is restricted by keeping as candidate edge points only those for which  $\max(|\kappa_1|, |\kappa_2|)$  is large. In the second step, edge points are extracted from the candidate points as follows. Let  $\mathbf{p}$  be the vertex considered. Then:

- If  $\kappa_1(\mathbf{p})$  is larger than  $\kappa_1(\mathbf{p} + \mathbf{e}_1)$  and  $\kappa_1(\mathbf{p} - \mathbf{e}_1)$ ,  $\mathbf{p}$  is marked as an edge point.
- If  $\kappa_2(\mathbf{p})$  is smaller than  $\kappa_2(\mathbf{p} + \mathbf{e}_2)$  and  $\kappa_2(\mathbf{p} - \mathbf{e}_2)$ ,  $\mathbf{p}$  is marked as an edge point.

The points thus identified have to be linked with neighboring edge points to create surface boundary curves. Another powerful method for extracting folds, using morphological operators adapted to triangle meshes, is described in [Rössl et al. 2000].

In what follows, we consider three other types of distinguished features (zero-crossings of the mean and Gaussian curvatures, ridges and umbilics) and discuss how these elements can be characterized and efficiently extracted from triangle meshes.

<sup>3</sup>For range images, there is another basic type of features, the occlusion boundaries (also called jump edges and depth discontinuities), where one surface hides another from the sensor's viewpoint.

5.2.1 *Zero-crossings of the mean and Gaussian curvatures.* Since segmentation is about dividing the data into regions that are homogeneous according to some criterion, it is natural to look for changes in concavity. Such changes occur at places where the mean curvature  $H$  vanishes. Krsek et al. [1997] note that the curves of inflection points may be the best distinguished features of a triangulation when the data has no sharp edges. In addition, it is important to detect where the surface is locally flat or at least flat in some direction. This information is given by the zero-crossings of  $K$ . This explains why many people have advocated segmenting according to the  $H$ - $K$  map as a preliminary step towards shape classification and reconstruction (see Figure 15 for the basic surface types according to the signs of  $H$  and  $K$ ).

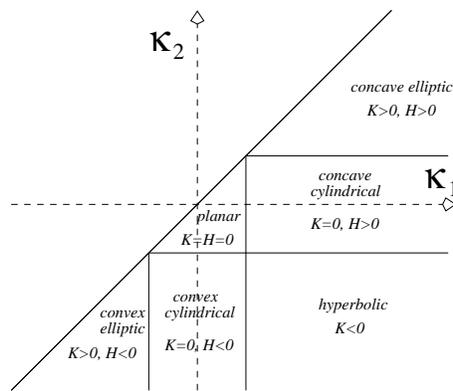


Fig. 15. The six basic surface types according to the signs of  $K$  and  $H$ .

Typical is the three-step approach of Besl and Jain [1988], which has generated much later research on region-based segmentation:

- *Rough segmentation:* The range image is divided into connected regions according to the  $H$ - $K$  map.
- *Seed placement:* Seeds are determined in regions having a sufficient number of points with a morphological operator.
- *Region growing:* Bivariate polynomials are used to produce an estimated surface fit to the largest seed region, starting with degree 1 up to degree 4. Then all pixels in all regions currently outside the seed region are tested for possible inclusion (based on difference in depth and comparison of normals). The largest connected region composed of pixels in the seed region and pixels that pass the compatibility tests is chosen as the new seed region. Expansion continues until there is almost zero change in region size. Finally, fit error is calculated. If it does not fall below a threshold, the region is rejected.

To recover quadrics in triangle meshes, the  $H$ - $K$  map is even more interesting. Indeed, none of them has concavity changes (i.e.  $H$  either vanishes or has constant sign everywhere) and each of them is made of a unique type of points, be them elliptic, hyperbolic, parabolic or planar (i.e.  $K$  either vanishes or has constant sign everywhere). It is thus of the utmost interest to segment the triangulation  $T$  along the regions of zero Gaussian curvature and zero mean curvature. Note however that the  $H$ - $K$  map is not sufficient to

classify all quadrics. For instance, both elliptic cylinders and elliptic cones are such that  $K = 0$  and  $H < 0$ . Planar and spherical points cannot be distinguished either with this map alone.

Extraction of regions of parabolic and inflection points can be made by thresholding the values of  $K$  and  $H$ . As Cai [1989] showed, thresholding must be made in a consistent manner, with the threshold  $\epsilon_K$  on  $K$  depending on the value of  $H$  (or conversely):

$$\epsilon_K \geq \epsilon_H^2 + 2|H|\epsilon_H.$$

After an extensive experimental assessment of range image segmentation algorithms based on estimates of the sign of the mean and Gaussian curvatures, Trucco and Fisher [1995] concluded that, because of quantization errors, noise smoothing is necessary before reliable extraction can take place. Unfortunately, denoising has an averaging effect. This has three consequences:

- The magnitudes of the curvatures are lower than the real ones. Thus,  $\epsilon_H$  should be a function of the number of smoothing cycles applied.
- Smoothing makes it difficult to distinguish accurately between planar patches and curved surfaces with low curvatures. It is thus advisable to first single out the planar patches (for instance using a specialized plane fitting technique).
- Sharp boundaries between regions may be distorted. Smoothing should thus be endowed with a mechanism to restrict the diffusion process to non-discontinuity points.

In view of the discrete nature of the data, consistent thresholding may miss zero-crossing lines. Consider two vertices  $\mathbf{p}_1$  and  $\mathbf{p}_2$  sharing a common edge of  $T$  and  $H_1, H_2$  the value of the mean curvature estimated at these points. If  $H_1 > \epsilon_H$  and  $H_2 < -\epsilon_H$ , there is a point  $\mathbf{p}_{12}$  between  $\mathbf{p}_1$  and  $\mathbf{p}_2$  with zero mean curvature that thresholding won't detect. Its location can be found as (see Figure 16):

$$\mathbf{p}_{12} = \mathbf{p}_1 + \frac{H_1}{H_2}(\mathbf{p}_2 - \mathbf{p}_1).$$

$\mathbf{p}_{12}$  must be linked with neighboring zero-crossings to detect where the features lines traverse the faces of the triangulations. Chain coding algorithms can then be used to form continuous boundary curves.

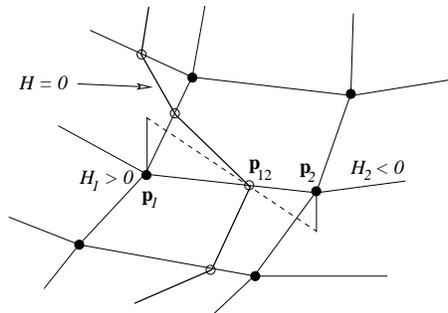


Fig. 16. Looking for points at which the mean curvature vanishes [Krsek et al. 1997].

Note that an alternative to the classification of surface points according to the signs of  $H$  and  $K$  is to use the *curvedness*  $R$ , which measures the size, and the *shape index*  $S$ , which measures the shape, of a local patch [Koenderink 1990]:

$$R = \frac{2}{\pi} \ln \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}}, \quad S = -\frac{2}{\pi} \arctan \left( \frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2} \right).$$

The Cartesian plane with the parameters  $(R, S)$  is then a conformal image of the  $\kappa_1 \geq \kappa_2$  part of the  $(\kappa_1, \kappa_2)$  plane (Figure 15).

**5.2.2 Ridges and valleys.** To segment a surface in meaningful parts and identify its most salient features, people have sought to generalize the notion of “edges” to smooth objects. In particular, a lot of focus has been put on geometric features known as ridges and valleys in the field of hydrology. *Valleys* are defined as the arrangement of ramified dry channels developed by the flow of water over the Earth’s surface and *ridges* are valleys of the inverted relief. Surface creases (ridges and valleys) have been studied in connection with research on structural geology, medical image analysis, face recognition and human perception.

There is considerable debate as to what constitutes a good mathematical characterization of ridges and valleys (see, e.g., [Lang et al. 1997; López et al. 1999]). One possible definition, which has proved to be meaningful in studies on the anatomy of the human brain [Pennec et al. 2000], is that of *crest lines*. Crest lines are the loci of the maxima of the maximal principal curvature (in absolute value) in the direction of the associated principal direction. In other words, if  $\nabla_{\mathbf{v}} f$  denotes the directional derivative of the scalar function  $f$  in the direction  $\mathbf{v}$ , i.e.

$$(\nabla_{\mathbf{v}} f)(\mathbf{p}) = \lim_{\tau \rightarrow 0} \frac{f(\mathbf{p} + \tau \mathbf{v}) - f(\mathbf{p})}{\tau} = \mathbf{v} \cdot \nabla f(\mathbf{p}),$$

then crest points are to be found as zero-crossings of  $\zeta = \nabla_{\mathbf{e}_1} \kappa_1$ . More precisely, crest points are of two types:  $\nabla_{\mathbf{e}_1} \zeta < 0$  and  $\kappa_1 > 0$  (positive largest curvature maxima, ridge-like) and  $\nabla_{\mathbf{e}_1} \zeta > 0$  and  $\kappa_1 < 0$  (negative largest curvature minima, valley-like).

Crest line extraction has been carried out especially on medical 3D data. For the case of iso-intensity surfaces, Thirion and Gourdon [1995] propose to compute these features by marching on the intersection of two implicit surfaces  $f = I$  and  $\zeta = 0$ . An example of computation is shown on Figure 17. This marching algorithm can be adapted to piecewise-linear surfaces [Lengagne et al. 1996]. For this, apply the following procedure to each face  $F$  of the mesh:

- For each vertex  $\mathbf{p}$  of  $F$ , estimate  $\zeta = \nabla_{\mathbf{e}_1} \kappa_1$  as follows. Choose the 1-ring neighborhood vertex  $\mathbf{p}_1$  of  $\mathbf{p}$  that maximizes  $(\mathbf{p}_1 - \mathbf{p}) \cdot \mathbf{e}_1$ , where  $\mathbf{e}_1$  is the estimated first principal direction at  $\mathbf{p}$ . Then take as estimate of the directional derivative of  $\kappa_1$  at  $\mathbf{p}$ :

$$\zeta(\mathbf{p}) = \kappa_1(\mathbf{p}_1) - \kappa_1(\mathbf{p}).$$

- If, for two 1-ring neighbors  $\mathbf{p}$  and  $\mathbf{q}$ ,  $\zeta(\mathbf{p})\zeta(\mathbf{q}) < 0$ , there is a crest point on the edge  $\mathbf{p}\mathbf{q}$ . Interpolate linearly  $\zeta$  along the edge to find the location of the zero-crossing of  $\nabla_{\mathbf{e}_1} \kappa_1$ .
- Another zero-crossing must appear on one of the other edges of  $F$ . Locate it and draw a segment between the two crest points found across the face.

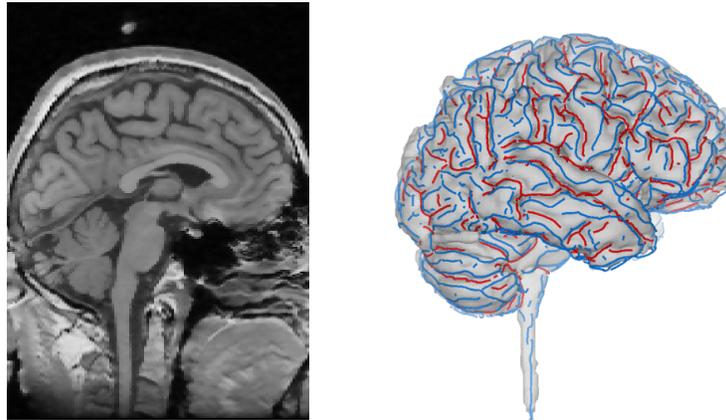


Fig. 17. A 3D magnetic resonance image (left) and a 3D view of the crest lines extracted superimposed on the surface of the brain (right). Red curves correspond to ridge-like crests, blue curves to valley-like crests [Pennec et al. 2000].

A simple thresholding on the interpolated value of  $\kappa_1$  at each crest point, compared to the maximum value of this curvature on the whole surface, allows the elimination of most of the spurious points.

Other authors have advocated defining ridges as the locus of points at which the maximal principal curvature  $\kappa_1$  attains a positive maximum along its line of curvature<sup>4</sup> and the valleys as the locus of points at which the minimal principal curvature  $\kappa_2$  attains a negative minimum along its line of curvature. As noted in [Belyaev and Ohtake 2000], this definition has a lot in common with a widely used definition of edges in classical image processing: edges are made of pixels at which the magnitude of the gradient of the image intensity has a local maximum in the direction of the gradient.

Algorithms have been proposed to detect the crease points so defined on surfaces approximated by triangle meshes. Define the principal centers of curvature of a point  $\mathbf{p}$  on a piecewise-smooth surface  $S$  as the points situated at distances  $1/\kappa_1$  and  $1/\kappa_2$  of  $\mathbf{p}$  on the line defined by  $\mathbf{p}$  and the surface normal  $\mathbf{n}$  at  $\mathbf{p}$ :

$$\mathbf{p} + \frac{1}{\kappa_1} \mathbf{n}, \quad \mathbf{p} + \frac{1}{\kappa_2} \mathbf{n}.$$

Call *focal surfaces* the loci of the principal centers. Then it turns out that the singularities of the focal surfaces are space curves corresponding either to crease curves of  $S$  (as defined above) or to points of  $S$  where the principal curvatures are equal (umbilics). The method proposed in [Lukács and Andor 1998] for extracting ridges and valleys is based on the following observation. Let  $\mathbf{pqr}$  be a small triangle on  $S$  and consider the associated triangle  $\mathbf{abc}$  on a focal surface,

$$\mathbf{a} = \mathbf{p} + \frac{1}{\kappa(\mathbf{p})} \mathbf{n}, \quad \mathbf{b} = \mathbf{q} + \frac{1}{\kappa(\mathbf{q})} \mathbf{n}, \quad \mathbf{c} = \mathbf{r} + \frac{1}{\kappa(\mathbf{r})} \mathbf{n},$$

<sup>4</sup>The lines of curvature are the curves whose tangents are in the direction of the principal curvatures. They form a natural parameterization of a smooth surface [Brady et al. 1985] and come in two colors (flavors): one for the maximal principal curvature and one for the minimal principal curvature.

where  $\kappa = \kappa_1$  or  $\kappa_2$ . Assume that  $\mathbf{o}$  is a surface point. Then  $\mathbf{o}$  is a crease point if and only if

$$\frac{\text{area}(\mathbf{abc})}{\text{area}(\mathbf{pqr})} \rightarrow 0 \quad \text{when } \mathbf{p} \rightarrow \mathbf{o}, \mathbf{q} \rightarrow \mathbf{o}, \mathbf{r} \rightarrow \mathbf{o}.$$

According to [Belyaev and Ohtake 2000], detection based on this observation has several drawbacks: it does not locate crease points well and is unable to separate the curvature extrema into maxima and minima.

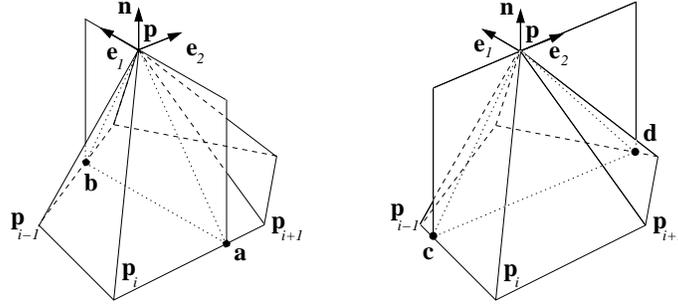


Fig. 18. Evaluating if  $\mathbf{p}$  is a ridge point [Belyaev and Ohtake 2000].

A different procedure was recently proposed [Belyaev and Ohtake 2000]. Suppose that we want to recover ridge points (this is no restriction: according to the definition, ridges and valleys are dual and changing the surface orientation turns ridges into valleys and vice versa). Consider the setup of Figure 18. To decide whether  $\kappa_1$  attains a maximum along its line of curvature, do the following:

- Find the intersection between the normal section plane generated by  $\mathbf{e}_1$  and  $\mathbf{n}$  and the 1-ring neighborhood of  $\mathbf{p}$ . Let the intersection consist of two points  $\mathbf{a}$  and  $\mathbf{b}$ . Suppose that  $\mathbf{a}$  belongs to  $\mathbf{p}_i\mathbf{p}_{i+1}$ . Compute the maximal principal curvature at  $\mathbf{a}$  and  $\mathbf{b}$  by interpolation. For instance,  $\kappa_1(\mathbf{a})$  can be estimated by linear interpolation between  $\kappa_1(\mathbf{p}_i)$  and  $\kappa_1(\mathbf{p}_{i+1})$ .
- Find the intersection between the normal section plane generated by  $\mathbf{e}_2$  and  $\mathbf{n}$  and the 1-ring neighborhood of  $\mathbf{p}$ . Let the intersection consist of two points  $\mathbf{c}$  and  $\mathbf{d}$ . Estimate  $\kappa_1(\mathbf{c})$  and  $\kappa_1(\mathbf{d})$ . Estimate  $\nabla_{\mathbf{e}_2}\kappa_1$  at  $\mathbf{p}$  as a function of  $\kappa_1$ ,  $\kappa_1(\mathbf{c})$  and  $\kappa_1(\mathbf{d})$ .
- Compute

$$\alpha_{\mathbf{a}} = \kappa_1(\mathbf{a}) + \frac{(\nabla_{\mathbf{e}_2}\kappa_1)^2}{2(\kappa_1 - \kappa_2)} \|\mathbf{pa}\|^2, \quad \alpha_{\mathbf{b}} = \kappa_1(\mathbf{b}) + \frac{(\nabla_{\mathbf{e}_2}\kappa_1)^2}{2(\kappa_1 - \kappa_2)} \|\mathbf{pb}\|^2.$$

Mark  $\mathbf{p}$  as a ridge point if  $\kappa_1$  is positive and larger than  $\alpha_{\mathbf{a}}$  and  $\alpha_{\mathbf{b}}$  simultaneously.

The above procedure produces many insignificant ridge vertices because of defects in the triangulated surface. One way to reduce the number of undesirable vertices is to apply one of the smoothing algorithms seen in § 4.1 to the triangulation beforehand. Thresholding the results can also help. Belyaev and Ohtake [2000] advocate taking two thresholds  $\kappa_{\text{low}}$  and  $\kappa_{\text{high}}$ , chosen so that 30 % (resp. 60 %) of all surface vertices are such that  $\kappa_1 < \kappa_{\text{low}}$  (resp.  $\kappa_1 < \kappa_{\text{high}}$ ), and retaining a chain of connected ridge vertices with curvature  $\kappa_1 > \kappa_{\text{low}}$  if

the chain contains at least one ridge vertex with  $\kappa_1 > \kappa_{\text{high}}$ . Even after this filtering, the crease patterns may be highly fragmented and simple morphological operators are used to reduce that fragmentation.

Figure 19 shows the ridges computed with this procedure on two different models. Ridge vertices are indicated by red strokes along the direction  $\mathbf{e}_2$  and valley vertices by blue strokes along  $\mathbf{e}_1$ .

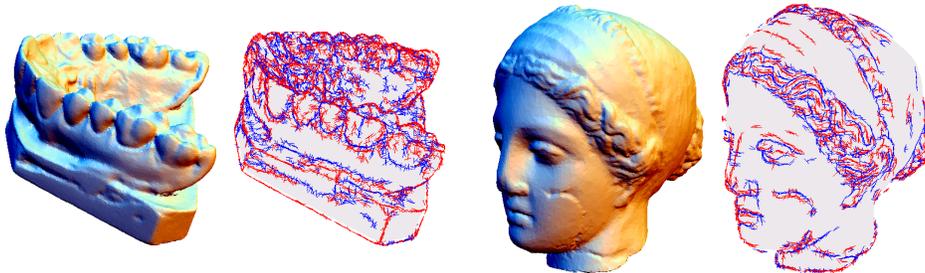


Fig. 19. From left to right: the teeth casting model and the ridges detected, the Venus model and the ridges detected [Belyaev and Ohtake 2000].

Related to ridges are other features whose significance is increasing, the sub-parabolic lines [Bruce et al. 1996]. They appear as the locus of geodesic inflections of the lines of curvature. As ridges and parabolic lines (which separate elliptic and hyperbolic regions), but contrary to lines of curvature, they are robust features, i.e. they deform if the surface is slightly deformed. An alternate characterization of sub-parabolic lines is as “counter-ridges”: they are the loci of points at which the principal curvature of one color has an extremal value when moving along a line of curvature of the other color.

**5.2.3 Umbilics.** Umbilics, i.e. points at which the principal curvatures are equal, are other important geometric features which may help guide the segmentation. Obviously, every point of a planar or spherical patch is an umbilic. But regions may also contain isolated umbilics. The ellipsoid, for instance, has four umbilics of a stable type, i.e. they do not vanish if the function representing the surface undergoes a small perturbation. If they can be identified, isolated umbilics may serve as seed points of a region-growing algorithm or help in the classification of local shapes.

Localizing umbilic points in sampled surfaces can be done by thresholding the “normalized” absolute value of the difference between the principal curvatures, i.e.

$$\frac{|\kappa_1 - \kappa_2|}{\max(|\kappa_1|, |\kappa_2|)} < \epsilon.$$

This is the approach taken in [Brady et al. 1985] to identify regions of umbilics in range images. However, as shown by Sander and Zucker [1992], the straightforward mathematical definition of umbilics as points with equal principal curvatures does not translate well into a computational procedure, especially for isolated umbilics.

Sander and Zucker advocate the use of the following alternate characterization to identify umbilics in a more robust fashion. Given a direction field at point  $\mathbf{p}$ , the *index* of  $\mathbf{p}$

is

$$\frac{1}{2\pi} \int_0^{2\pi} \psi(\mathbf{r}) \, d\mathbf{r},$$

where  $\psi(\mathbf{r})$  is the angle between the direction of the field and some fixed direction, and the integral is taken over a small counterclockwise circuit about  $\mathbf{p}$ . At all points except at umbilics, the index is zero. Implementation of this definition thus provides a conclusive test for umbilics: if the index over a local neighborhood is zero, no umbilic is present.

Let  $\mathbf{p}$  be a vertex of a triangulation  $T$ . Take the first principal direction as direction field. Locally pull back this field onto the “tangent plane” at  $\mathbf{p}$  and compute the circuit there (this does not change the qualitative nature of the field). Consider the smallest circle centered at  $\mathbf{p}$  enclosing the 1-ring neighborhood of  $\mathbf{p}$ , as in Figure 20. At each point  $\mathbf{x}_i$  of this circle, we can estimate the first principal direction  $\mathbf{e}_1$  by interpolation from the values computed at the vertices of  $T$ . Assume the counterclockwise circuit is divided in  $w$  incremental steps of equal lengths and let  $\Delta\xi_i$  be the change in angle that  $\mathbf{e}_1$  makes between  $\mathbf{x}_i$  and  $\mathbf{x}_{i+1}$ . Then the index of  $\mathbf{p}$  is estimated as:

$$\frac{1}{2\pi} \sum_{i=0}^w \Delta\xi_i.$$

Work remains to be done to assert how accurately this method can identify isolated umbilics in realistic noisy data and how such features can be best used for segmentation purposes.

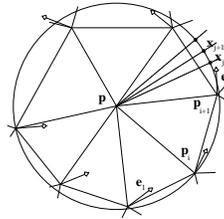


Fig. 20. Computing the index at  $\mathbf{p}$ . Here  $\mathbf{p}$  is an umbilic, with index  $\frac{1}{2}$ .

### 5.3 Recovering specific types of quadrics

Once Darboux frames have been estimated at each vertex of the triangulation, once surface folds have been detected, once an initial segmentation has been made using for instance parabolic and ridge lines, we are left with a division of the data into sets which correspond to smooth surface patches. Focus is then on the identification of sub-patches of these that correspond to patches from quadric surfaces.

In the past, many researchers have sought to extract quadrics (mostly natural quadrics) from range imagery: [Oshima and Shirai 1983; Taylor et al. 1989; Bock and Guerra 1999] detect planes, [Bolles and Fischler 1981; Grimson et al. 1993] look for cylinders, [Hebert and Ponce 1982; Jones and Illingworth 1994] identify planes, cylinders and cones, [Han et al. 1987; Yokoya and Levine 1989; Flynn and Jain 1991] extract planes, cylinders and spheres, [Boulanger and Rioux 1987] consider planes, spheres, ellipsoids and other simple quadrics, [Flynn and Jain 1988; Newman et al. 1993] deal with planes, spheres, cylinders

and cones and [Faugeras et al. 1983; Faugeras and Hebert 1986; Fan et al. 1987; 1989] consider general quadrics.

Among these works, many have used differential parameter estimates for quadric surface classification and parameter estimation. Fan et al. [1987; 1989] start by segmenting the surface at “discontinuities” (jump boundaries, folds and parabolic lines) which are detected by examining zero-crossings and extremal values of surface curvature measures. The sub-patches of the resulting initial segmentation are approximated by quadrics, whose coefficients are computed by a least-squares method. Yokoya and Levine [1989] use a hybrid approach to extract quadrics of revolution from range data. Principal curvatures are used to produce three initial segmentations, based on detection of homogeneous regions, occluding contours and sharp edges. A combination of these initial maps is used to compute the final segmentation.

For quadric parameter estimation, some researchers have used accumulation techniques based on the Hough transform<sup>5</sup> instead. Hebert and Ponce [1982] use Hough techniques to classify surfaces as planar, cylindrical and conical by clustering projected points along surface normals onto the Gaussian sphere (the unit sphere of directions). Muller and Mohr [1984] present a general framework for surface parameter computation using a divide-and-conquer Hough approach. Yokoya and Levine [1989] use the Hough transform to find the parameters of surface of revolution. Newman et al. [1993] use accumulation techniques for spherical and cylindrical parameter estimation.

#### 5.4 Segmentation for quadric surface recovery

Many authors have noted that the region-growing paradigm to segmentation is the most appropriate for general triangulated surfaces. This section presents three region-growing segmentation algorithms that have been specifically designed for extracting quadric surface patches.

Note that there are many, more general, segmentation methods which could certainly be applied to the recovery of quadric surfaces. Among them, let us mention the anisotropic diffusion scheme of [Chaine et al. 1999], the region-growing algorithm driven by differential features of [Bricault and Monga 1997], the recover-and-select segmentation strategy of [Leonardis et al. 1995] (used in [Lukács et al. 1998]), the watershed segmentation of [Mangan and Whitaker 1999], the tensor voting approach of [Tang and Medioni 1998; 1999], the hybrid approach of [Lejeune and Ferrie 1996] and the construction of polynomial surfaces through region growing of [Sapidis and Besl 1995].

*5.4.1 Update statistics for region growing.* McIvor and Waltenberg [1998] describe techniques for the robust identification of patches from planes, spheres and cylinders in range images. The authors propose two segmentation algorithms, one based on region growing and the other on unsupervised classification. The first stage of both algorithms

<sup>5</sup>The Hough transform is a technique used to isolate features of a particular kind within the data. It requires that the desired features be specified in some parametric form, so is most commonly used for the detection of regular features such as lines, circles, ellipses, etc.

The standard formulation of the Hough transform for line detection is as follows. Suppose given a set of image points and the goal is to determine subsets of them lying on straight lines. The Hough transform associates an image point with a line in the parameter space. Since collinear points are transformed into lines intersecting at the same point, the problem is solved as an intersection problem in parameter space.

The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions. It is also relatively insensitive to image noise.

is the division of the data into smooth surface patches separated by occlusion boundaries and surface folds. Then the emphasis of this work is on a boundary representation of 3D data that makes explicit the underlying class of surface of the component patches that each smooth patch is divided into, i.e. the identification of sub-patches that correspond to pieces of planar, spherical and cylindrical surfaces. The common framework of both algorithms is as follows:

1. Find all planar points in the data and extract connected planar regions within them.
2. Find all umbilics within the data which are not assigned to a planar region and extract connected spherical regions from them.
3. Find all parabolic points within the data not yet assigned to a planar or spherical region and extract connected cylindrical regions from them.

Let us start with the region-growing approach. Given a set of points from a smooth region, the curvature properties of a particular class of surface are used to identify and reject points that could not be from such a surface. The structured surface patches are then identified within these regions by further classification on the basis of position and principal quadric. In other words, no costly least-squares fitting takes place.

Planar surface patches are simple to characterize in terms of point properties of the principal curvatures. Namely, both principal curvatures must be zero or, equivalently, the Gaussian and mean curvatures must vanish. Thus planar patches can be extracted in two simple steps. First, threshold the principal curvatures about 0 to detect planar points. Then, group detected points into patches by a region labeling algorithm.

Points on a spherical patch are umbilic, i.e. the directional curvatures are equal in all directions. Thus spherical surface patches can be segmented by first identifying points which satisfy  $|\kappa_1 - \kappa_2| < \epsilon$  (or  $|H^2 - K| < \epsilon^2/4$ ) and then computing the regions of spatially constant principal curvature. The authors propose two methods to achieve this.

The first is based on region growing. Point aggregation is started from a digital center of the mask generated by identifying umbilic points. What is needed is:

- A *similarity measure*, which says when a data point on the boundary of the already extracted spherical region  $R$  is “similar” enough to the current estimate of the parameters of the sphere underlying  $R$  that it should be considered part of  $R$ .
- A method for *updating* the estimate of the sphere’s parameters given the new point added to  $R$ .

First, a point  $\mathbf{p}$  on the boundary of  $R$  is mapped to the sphere on which it lies:

$$r_{\mathbf{p}} = \frac{1}{|H_{\mathbf{p}}|}, \quad \mathbf{c}_{\mathbf{p}} = \mathbf{p} + \frac{1}{H_{\mathbf{p}}} \mathbf{n}_{\mathbf{p}},$$

where  $H_{\mathbf{p}}$  and  $\mathbf{n}_{\mathbf{p}}$  are estimates respectively of the mean curvature and the normal at  $\mathbf{p}$ , and  $r_{\mathbf{p}}$ ,  $\mathbf{c}_{\mathbf{p}}$  the radius and center of the sphere on which  $\mathbf{p}$  lies. Assume that there are  $j$  points in the already extracted spherical region  $R$ . The point  $\mathbf{p}$  is accepted as part of  $R$  if:

$$(r_{\mathbf{p}} - \bar{r}_j)^2 \leq \alpha^2(\check{r}_j + \acute{r}_{\mathbf{p}}) \quad \text{and} \quad \|\mathbf{c}_{\mathbf{p}} - \bar{\mathbf{c}}_j\|^2 \leq \alpha^2(\check{\mathbf{c}}_j + \acute{\mathbf{c}}_{\mathbf{p}}),$$

where  $\bar{r}_j$  is the mean and  $\check{r}_j$  the variance of the extracted surface’s radius estimate,  $\acute{r}_{\mathbf{p}}$  the radius measurement error variance (a function of the error in measuring  $\mathbf{p}$ ),  $\bar{\mathbf{c}}_j$ ,  $\check{\mathbf{c}}_j$ ,  $\acute{\mathbf{c}}_{\mathbf{p}}$  have similar meanings for the sphere’s center, and  $\alpha$  is a chosen significance level. These

validation gates assume that current estimates and measurement errors are independent and Gaussian and that the distribution of the sphere's center is symmetric in space. When a new point is added to  $R$ , the statistics, which are simple, uniformly weighted, estimators, are updated as follows:

$$\begin{aligned}\bar{r}_{j+1} &= \frac{j}{j+1}\bar{r}_j + \frac{1}{j+1}r_{j+1}, & \check{r}_{j+1} &= \frac{j}{j+1}\check{r}_j + \frac{1}{j+1}(r_{j+1} - \bar{r}_{j+1})^2, \\ \bar{\mathbf{c}}_{j+1} &= \frac{j}{j+1}\bar{\mathbf{c}}_j + \frac{1}{j+1}\mathbf{c}_{j+1}, & \check{\mathbf{c}}_{j+1} &= \frac{j}{j+1}\check{\mathbf{c}}_j + \frac{1}{j+1}\|\mathbf{c}_{j+1} - \bar{\mathbf{c}}_j\|^2.\end{aligned}$$

An alternative to region growing to aggregate points is clustering, i.e. unsupervised learning. The number of clusters  $c$ , which is needed by the clustering algorithm, is not known *a priori*. It is computed by repeating the clustering procedure for  $c = 1, 2, 3, \dots$ , and looking at how the sum-of-squared error  $J_e(c)$ , used by the algorithm to determine the optimal partitioning, varies with  $c$ . A suitable value is the smallest  $c$  such that

$$\frac{J_e(c+1)}{J_e(c)} > 1 - \epsilon,$$

where a good value for  $\epsilon$  is 0.01.

Similar procedures are given for growing cylindrical regions. Given a point  $\mathbf{p}$ , with estimated normal  $\mathbf{n}_p$ , mean curvature  $H_p$  and principal direction  $\mathbf{e}_p$  in which the principal curvature is 0, the parameters of the cylinder on which  $\mathbf{p}$  must lie are its radius  $r_p$ , the direction of its axis  $\mathbf{d}_p$  and a unique point on axis  $\mathbf{x}_p$ , with:

$$r_p = \frac{1}{2|H_p|}, \quad \mathbf{d}_p = \pm \mathbf{e}_p, \quad \mathbf{x}_p = \mathbf{x}_0 + \left(\mathcal{I} - \bar{\mathbf{d}}_j \bar{\mathbf{d}}_j^T\right) \left(\mathbf{p} + \frac{1}{2H_p} \mathbf{n}_p\right),$$

where  $\mathbf{x}_0$  is an arbitrary fixed point and  $\bar{\mathbf{d}}_j$  is the current estimate of the axis direction.

The new point  $\mathbf{p}$  is added to the already extracted cylindrical region if:

$$(r_p - \bar{r}_j)^2 \leq \alpha^2(\check{r}_j + \check{r}_p), \quad \|\mathbf{x}_p - \bar{\mathbf{x}}_j\|^2 \leq \alpha^2(\check{\mathbf{x}}_j + \check{\mathbf{x}}_p),$$

with meanings as before, and if the angle between the axis direction  $\mathbf{d}_p$  and the current axis direction estimate mean is outside a validation gate. As in the sphere case, statistics are updated with recursive implementations of standard uniformly weighted estimators [McIvor and Waltenberg 1998].

Cylinders can also be recovered using a clustering algorithm. In this case, the authors advocate splitting clustering into three sequential stages: clustering by radius, clustering by axis direction and clustering by axis intersection point.

**5.4.2 Reverse engineering regular objects.** In CAGD, a question that naturally arises is how a manufactured part compares with the way it was originally designed. To make this comparison, a solid model of the existing part has to be built. Generating a solid model from CMM (coordinate-measuring machine) point data is the goal of reverse engineering. The created model should be made of continuous surface elements forming the boundaries of the existing object and possibly reflect its design structure as accurately and concisely as possible [Várady et al. 1997].

Since low-degree surfaces encompass most of the conventional parts in industrial environments, many works in reverse engineering have dealt with the recovery of quadric surfaces (see, e.g., [Chivate and Jablokow 1993; Várady et al. 1998; Yang and Lee 1999;

Thompson et al. 1999]). We here give a detailed presentation of the algorithm of [Várady et al. 1998] which is well adapted to the reverse engineering of *regular* objects, i.e. objects bounded by relatively large primary surfaces (planes, cylinders, cones and spheres – we leave out tori that the authors also consider) smoothly connected by relatively small transition surfaces (blending surfaces or healing strips). An example of such a regular object is given in Figure 21.

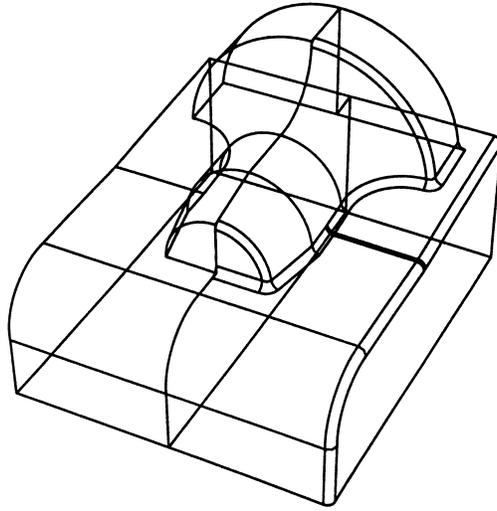


Fig. 21. A simple example of regular object [Várady et al. 1998].

Arguing that simpler surfaces occur more frequently in mechanical engineering and can be detected in a more reliable and efficient manner, the authors advocate a direct, non-iterative segmentation method which gradually recovers surfaces of increasing complexity. The algorithm consists of the following elementary steps<sup>6</sup>:

1. *Compute an initial segmentation* by identifying highly curved point regions, i.e. points in the vicinity of sharp edges or lying on blended edges. Call any region thus outlined simple if it is well approximated by a single quadric surface, and multiple otherwise.
2. *Identify simple regions* by testing a sequence of simple hypotheses, in increasing order of complexity: test whether the surface is a plane, a cylinder, a cone or a sphere. If none of these tests is successful, the region is classified as multiple.
3. *Segment multiple regions* by dimensionality filtering on the Gaussian sphere.
4. *Build a B-rep* with the detected primary surfaces and compute transition surfaces.

The initial segmentation is made by filtering highly curved regions. At each vertex  $\mathbf{p}$  of the triangle mesh  $T$ , compute the following value:

$$h = \frac{1}{m} \sum_i \mathbf{n} \cdot (\mathbf{p}_i - \mathbf{p}),$$

<sup>6</sup>Várady et al. [1998] start with a point cloud and estimate normals without reconstructing a piecewise-linear interpolating surface. We assume we have a triangle mesh for which vertex normals have been estimated.

where  $\mathbf{n}$  is the unit normal at  $\mathbf{p}$  and the  $\mathbf{p}_i$  are the  $m$  1-ring neighbors of  $\mathbf{p}$ . If  $h$  is larger than some predefined threshold,  $\mathbf{p}$  is likely to be in the vicinity of a sharp edge or to belong to a transition surface or to be a very noisy data point, and is temporarily removed.

After the initial segmentation step, simple surfaces are identified by increasing order of complexity, starting with planes, then cylinders, cones and spheres. Planes are found by least-squares fitting (see Section 6). If the error-of-fit is too large, look for cylinders. Cylinders can also be detected by a least-squares technique but a different approach based on the Gaussian sphere is used. The idea is that on an ideal cylinder the normal vectors are orthogonal to its axis. On the Gaussian sphere, they determine a great circle. Thus, to test whether the region under consideration is well described by a cylinder, the vertex normals are projected onto the Gaussian sphere and a plane is fitted to the projected points. If the fit is good, the normal to the plane gives the axis of the cylinder. Projecting the data points onto a plane orthogonal to this axis and fitting a circle gives the radius of the cylinder (again if the region is indeed cylindrical).

A similar procedure is used to detect cones. The normal vectors to an ideal cone determine a small circle on the Gaussian sphere. Again fit a plane to the projected points, which gives the direction of the cone axis. The radius of this small circle is equal to  $\sin \alpha$ , where  $\alpha$  is the semi-vertical angle. One can find rulers of the cone – and thus the apex – by projecting data points onto a plane orthogonal to the axis direction. Finally, spheres are detected by least-squares fitting.

To segment multiple regions, Várady et al. [1998] use a tool called dimensionality filtering. The idea is to label each point according to the density of points in the vicinity of its projection on the Gaussian sphere. Take two concentric balls around the projection of a point  $\mathbf{p}$  on the Gaussian sphere (radii  $r_1$  and  $r_2$ ) and count the number of points within the balls ( $n_1$  and  $n_2$ ). The dimensionality of the surface underlying  $\mathbf{p}$  is estimated as:

$$D = \frac{\log \frac{n_2}{n_1}}{\log \frac{r_2}{r_1}}.$$

The ratio of the radii is set to 1:2. If  $\mathbf{p}$  belongs to a planar region, then all the normals around it project (ideally) to a single point on the Gaussian sphere. Thus  $D \approx 0$ . If it stands in a cylindrical or conical area, then  $D \approx 1$ . Otherwise  $D \approx 2$  and  $\mathbf{p}$  is on a spherical patch.

The idea then is to first keep only those points with  $D \approx 0$  and identify the corresponding planar subregions (each cluster of points corresponds to such a subregion). Those points are then removed. Then points with  $D \approx 1$  are considered. Translational axes (great circles) can be identified with a Hough transform. The parameters of the associated cylinders are computed as before. A similar approach allows to identify cones. After removal of planar, cylindrical and conical points, the remaining subregions should be of rotational type and spherical patches can be extracted. A final trick allows to identify if a subregion is a smoothly connected composition of several rotational surfaces.

After primary surfaces have been recovered, an initial B-rep model is constructed. If two adjacent surfaces intersect enough transversally (the angle between the two normals at the points of intersection is larger than a specified value), the actual sharp edge is computed by surface-surface intersection and then replaced by a blending surface. Otherwise, the edge is considered quasi-smooth. In that case, a rough, approximate polygonal edge is computed and substituted by a healing strip [Kós et al. 2000].

5.4.3 *The UE segmenter.* The first global effort at setting up a framework for comparing experimentally and assessing the performance of range image segmenters was recently reported [Hoover et al. 1996]. Four planar segmentation algorithms have been tested on 80 real images with ground truth and objective performance measures<sup>7</sup>. One conclusion is that the planar segmenter developed over the years at the University of Edinburgh has the best performance among current algorithms. In fact, even though this study was restricted to planar segmenters, the UE algorithm actually extracts quadric surface patches. Since it appears to perform nicely on both plane and quadric surface patches, we think it worth describing at length the underlying segmentation strategy.

The UE algorithm for segmenting complete range images has three main ingredients. Initially, the data undergoes a *rough segmentation* from which seed patches are determined. Starting from these patches, a *region growing* is performed, during which a special constraint allows for patch edge adjustment. Finally, topological information is extracted from the result of the segmentation and a *conversion* to a B-rep representation is performed.

In early version of their segmentation algorithm, researchers at UE used an initial segmentation based on the  $H$ - $K$  map, labeling vertices as belonging to particular surface types based on the combined signs of the mean curvature  $H$  and the Gaussian curvature  $K$  [Trucco and Fisher 1992; 1995]. Each curvature value is classified as Negative, Zero, Positive or Unknown based on two thresholds, called “inner” and “outer”. The “inner” threshold determines the range of values called Zero. The “outer” threshold determines the limit of the ranges of the Negative and Positive values. Between these values, vertices of  $T$  are labeled as Unknown. Once this basic classification is made, vertices of similar labeling are grouped to form initial regions. This first segmentation map is morphologically dilated and eroded in a specific manner to fill small Unknown areas, remove small regions, separate thinly connected components and provide seed patches for the region growing stage.

Experimenting with this rough segmentation based on the  $H$ - $K$  map, the authors noticed that later stages of the algorithm had a more significant impact on performance than the initial curvature classification. Accordingly, they chose in recent versions of their segmenter to implement a simpler strategy for initially segmenting the data [Fisher et al. 1997; Robertson et al. 1999]. This shape classification process estimates the local curvedness by finding the maximum angle  $\theta_{max}$  that adjacent surface polygons turn away from the current polygon  $P$ . This angle is found by examining the angle between polygon normals in a neighborhood about  $P$ . It gives an indication of how curved the surface is locally and forms the basis for the initial labeling of polygons into different surface shape classes:

- If  $\theta_{max} < \tau_{plane}$  (= 5 deg),  $P$  is labeled “planar”.
- If  $\theta_{max} > \tau_{edge}$  (= 10 deg),  $P$  is labeled “edge”.
- Otherwise,  $P$  is labeled “quadric”.

(Note that only the surface orientation discontinuities are detected by this classification scheme, since the authors consider complete range descriptions and not single view images.) Surface patches are then formed by grouping polygons with a similar label to form the initial seed patches, which are cleaned up using a morphological operator.

<sup>7</sup>Comparison of curve-surface segmenters is only at an early stage – see [Powell et al. 1998].

Starting from this rough segmentation, an initial quadric surface is fitted to each region above a minimal size. Region growing is then performed through an iterative expand/fit/contract cycle:

- First, the boundary of the current region is *expanded* by adding to it adjacent points whose position and normal are within a minimal agreement with the position and normal of the closest surface point. The region is extended in this manner as far as possible.
- Then, a quadric surface is *fitted* to each new region formed using Taubin’s generalized eigenvalue fit (see [Taubin 1991] and Section 6). Note that the surface model used is the general quadric patch form, not Monge’s form as in [Besl and Jain 1988]. In early versions of the algorithm [Fitzgibbon et al. 1997], the decision about which type of surface to fit was made by examining the covariance matrix of the region points (Section 3.3). If the ratio of its two smallest eigenvalues exceeded some threshold, a plane was fitted. Otherwise, a general quadric was fitted. In newer versions [Robertson et al. 1999], the selection between the different shape classes is based on minimizing the surface fit error. If this error is comparable for several classes, the selection is biased in favor of simpler models. For instance, if a plane, a cylinder and a more general quadric are in competition, the following quantities are computed:

$$e_{\text{plane}} = \frac{1}{\sigma^2} \sum \epsilon_i^2 + 3\alpha, \quad e_{\text{cylinder}} = \frac{1}{\sigma^2} \sum \epsilon_i^2 + 5\alpha, \quad e_{\text{quadric}} = \frac{1}{\sigma^2} \sum \epsilon_i^2 + 9\alpha,$$

with  $\epsilon_i$  the error distance between the model and the polygon (estimated using Taubin’s fitting algorithm),  $\sigma$  is the estimated standard deviation of the polygon patch center from the true surface and  $\alpha$  is a heuristically chosen weight factor. The class with the smallest  $e_i$  is chosen.

- Finally, a *contraction* of the boundary of each region is performed. Each of the points added in the expansion step is tested against the new surface estimate. Those that are not best accounted for by the new surface are returned to the pool of unfitted polygons.

This expand/fit/contract cycle goes until the region boundary stabilizes. In later iterations, an additional constraint is added to adjust patch boundaries, which may otherwise be very ragged due to the effects of noise in the data. This constraint incorporates the idea that ambiguous points (those that are within the distance threshold of more than one region) should be on the “right” side of the boundary between adjacent regions. A point is thus likely to switch sides if it turns out that it is on the wrong side of a theoretical boundary given by some decision surface. This method gives very good results in terms of regularity of boundaries. An example of segmentation is shown on Figure 22.

Finally, once the segmentation is over, a post-processing stage converts the segmentation result to a B-rep model and recovers the topological information. The output is in the form of shape parameters (surface equation in a canonical position), surface extent (collection of space curves lying on the surface and a point defining the interior) and position.

## 6. QUADRIC SURFACE FITTING

We now assume that we are given a set of 3D points. We want to find the quadric surface best fitting this point set and have a measure of the goodness of fit, i.e. of the “minimum distance” between the point set and a quadric, so as to decide if the surface fitted does indeed adequately represent the entire point set or if it needs to be broken into smaller

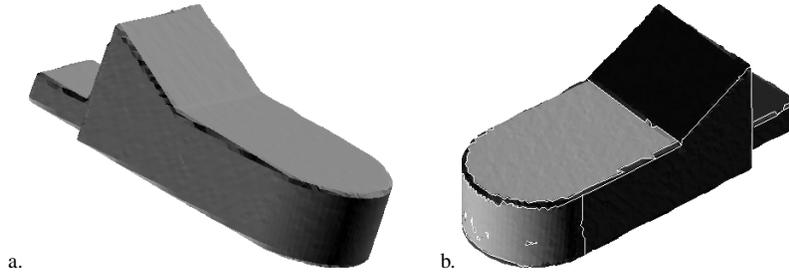


Fig. 22. Segmentation of a machined part [Robertson et al. 1999]. a. Full range description of a part with cylindrical and planar surfaces. b. Segmented description of the part. Patch boundaries are indicated by white lines.

pieces. Unfortunately, while plane fitting is well understood, the least-squares fitting of (even low-degree) curved surfaces has received much less attention.

This section reviews the literature applicable to the fitting of second-order surfaces. We start by giving general considerations on linear and non-linear algebraic surface fitting (§ 6.1) and then look at methods for the specific reconstruction of quadrics (§ 6.2).

### 6.1 General considerations

Let  $\Omega$  be a set of  $n$  3D points  $\mathbf{p}_i = (x_i, y_i, z_i), i = 1, \dots, n$ . The goal is to find the surface  $S$ , in a family of surfaces  $\chi$  parameterized by  $G \subseteq \mathbb{R}^s$ , which best describes the point set  $\Omega$ . Assume that elements of  $\chi$  are defined in implicit polynomial form:

$$f(\mathbf{p}, \mathbf{s}) = 0, \quad \mathbf{p} \in \mathbb{R}^3, \mathbf{s} \in G,$$

where  $f$  has degree  $p$ ,  $s = \binom{p+1}{3} - 1$ . A surface  $Z(f)$  (zero-set of  $f$ ) which goes through all the points of  $\Omega$  is the member of  $\chi$  which corresponds to the solution of the following system of linear equations:

$$f(\mathbf{p}_i, \mathbf{s}) = 0, \quad i = 1, \dots, n. \quad (15)$$

Sometimes, there are additional constraints (which for instance restrict the surface to be of a specific type)

$$H_1(\mathbf{s}) = 0, \dots, H_k(\mathbf{s}) = 0 \quad (16)$$

forming a subspace of dimension  $t$  of  $G$ , for some integer  $t < s$ . In that case, the constraints (16) are used to eliminate  $t$  unknowns from (15) and reduce the problem to an unconstrained optimization problem in a lower dimensional space.

Usually,  $n$  is much larger than the number of degrees of freedom  $s$  and the system of equations (15) is overdetermined. Thus, in general, it cannot be solved, except in the least-squares sense: find the surface  $Z(f)$  which best fits the point set “on the average”, i.e. which minimizes

$$E(\mathbf{s}) = \frac{1}{n} \sum_{i=1}^n f(\mathbf{p}_i, \mathbf{s})^2.$$

$f(\mathbf{p}_i, \mathbf{s})$  is called the *algebraic distance* from  $\mathbf{p}_i$  to  $Z(f)$ .

Assume first that the constraints (16) are linear, so the problem turns into a linear least-squares minimization with  $s' \leq s$  unknown parameters. For the sake of illustration, we

assume below that  $s' = s$ . Eq. (15) can be written in matrix form:

$$\mathcal{A}\mathbf{s} = \mathbf{b},$$

where  $\mathcal{A}$  is a  $n \times s$  matrix

$$\mathcal{A} = \begin{pmatrix} x_1^p & x_1^{p-1}y_1 & \cdots & y_1^p & y_1^{p-1}z_1 & \cdots & z_1^p & \cdots & z_1x_1^{p-1} & x_1^{p-1} & \cdots & x_1 & y_1 & z_1 \\ x_2^p & x_2^{p-1}y_2 & \cdots & y_2^p & y_2^{p-1}z_2 & \cdots & z_2^p & \cdots & z_2x_2^{p-1} & x_2^{p-1} & \cdots & x_2 & y_2 & z_2 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & & \vdots & \vdots & & \vdots & \vdots & \vdots \\ x_n^p & x_n^{p-1}y_n & \cdots & y_n^p & y_n^{p-1}z_n & \cdots & z_n^p & \cdots & z_nx_n^{p-1} & x_n^{p-1} & \cdots & x_n & y_n & z_n \end{pmatrix}$$

and  $\mathbf{b} = (1, \dots, 1)^T$  is a vector of length  $n$ . Solving this system in the least-squares sense means finding the solution  $\mathbf{s}$  which minimizes the norm of the residual error

$$\mathbf{r} = \mathcal{A}\mathbf{s} - \mathbf{b}.$$

Assume that the surface  $Z(f)$  does not go through the origin. If  $\mathcal{A}$  is not singular, then  $\mathbf{s}$  is unique and given by the normal equation:

$$\mathbf{s} = (\mathcal{A}^T\mathcal{A})^{-1}\mathcal{A}^T\mathbf{b}.$$

If  $\mathcal{A}$  is singular, which may happen if several of the  $\mathbf{p}_i$ 's are concentrated at a single point or on a straight line, a singular value decomposition can be used. This consists in finding a decomposition of  $\mathcal{A}$  into

$$\mathcal{A} = \mathcal{U}\mathcal{W}\mathcal{V}^T,$$

where  $\mathcal{W}$  is a diagonal matrix with elements  $w_1 \geq \dots \geq w_r > w_{r+1} = \dots = 0$  and  $\mathcal{U}, \mathcal{V}$  are orthogonal matrices. If we let  $\delta_j = 1/w_j$  if  $w_j \neq 0$  and  $\delta_j = 0$  otherwise, then the pseudoinverse of  $\mathcal{A}$  is

$$\mathcal{A}^{-1} = \mathcal{V}[\text{diag}(\delta_j)]\mathcal{U}^T$$

and the solution  $\mathbf{s}$  is given by

$$\mathbf{s} = \mathcal{A}^{-1}\mathbf{b}.$$

If the surface goes through the origin, i.e.  $c = f((0, 0, 0), \mathbf{s}) \neq 0$ , both of the above two methods will result in the trivial solution  $\mathbf{s} = 0$ . A way of solving this problem is to treat  $c$  as a variable [Cao and Shrikhande 1991]. Let  $\mathcal{A}_1$  be the matrix  $\mathcal{A}$  augmented by the vector  $\mathbf{b} = (c, \dots, c)^T$ . Then the following linear system of equations can be established:

$$\mathcal{A}_1\mathbf{s} = 0.$$

Multiplying both sides by  $\mathcal{A}_1^T$  yields

$$\mathcal{A}_1^T\mathcal{A}_1\mathbf{s} = 0.$$

Denote by  $\lambda_j$  the eigenvalues of  $\mathcal{A}_1^T\mathcal{A}_1$ . If one of the  $\lambda_j$ 's vanishes, then the solution  $\mathbf{s}$  is the corresponding eigenvector. If none of the  $\lambda_j$ 's vanishes, the eigenvector associated to the minimum eigenvalue of the matrix is taken as solution  $\mathbf{s}$  of the least-squares fit.

Minimizing the sum of squared residual errors (we drop the reference to  $\mathbf{s}$ )

$$E(\mathbf{s}) = \frac{1}{n} \sum_{i=1}^n f(x_i, y_i, z_i)^2$$

as above works well if the data points satisfy the condition that the independent variables (say  $x, y$ ) are measured without error and the dependent variable ( $z$ ) has Gaussian noise [Cao and Shrikhande 1991]. Unfortunately, this condition may not be satisfied for real-world data. And the residual error measure may be very large even when a data point  $\mathbf{p}_i$  is very close to the surface  $f$ . The reverse can also happen: the error measure can be small even when several of the data points are far from the surface [Bookstein 1979].

An alternative is to minimize the sum of squared orthogonal distances to the zero-set of  $f$ :

$$E(\mathbf{s}) = \frac{1}{n} \sum_{i=1}^n d(\mathbf{p}_i, Z(f))^2. \quad (17)$$

Even though  $f$  and  $d$  have the same roots in space, they behave a lot differently for points which do not lie on  $Z(f)$ . By contrast to the linear least-squares method above, fitting based on distance  $d$  is not biased. If the distances are normally distributed with mean zero, the solution is a maximum likelihood estimate of the parameters. This approach allows for noise in all three variables.

Unfortunately, minimizing (17) is computationally impractical – except when both  $f$  and the constraints (16) are of simple form, e.g. linear with respect to the parameters of  $\mathbf{s}$  – because there is no closed form expression for the true distance  $d$  from a point to an algebraic surface. In practice, fitting methods generally rely on approximations to this distance. Taubin [1991] proposed to use a first-order approximation

$$d(\mathbf{p}, Z(f))^2 \approx \frac{f(\mathbf{p})^2}{\|\nabla f(\mathbf{p})\|^2}.$$

For surfaces having the value of  $\|\nabla f(\mathbf{p})\|$  constant on  $Z(f)$ , it turns out that

$$E(\mathbf{s}) = \frac{1}{n} \sum_{i=1}^n \frac{f(\mathbf{p}_i)^2}{\|\nabla f(\mathbf{p}_i)\|^2} \approx \frac{\frac{1}{n} \sum_{i=1}^n f(\mathbf{p}_i)^2}{\frac{1}{n} \sum_{i=1}^n \|\nabla f(\mathbf{p}_i)\|^2} = \frac{\mathbf{s}\mathcal{M}\mathbf{s}^T}{\mathbf{s}\mathcal{N}\mathbf{s}^T}, \quad (18)$$

where  $\mathcal{M}$  and  $\mathcal{N}$  are symmetric matrices:

$$\mathcal{M} = \frac{1}{n} \sum_{i=1}^n \mathbf{h}_i \mathbf{h}_i^T, \quad \mathcal{N} = \frac{1}{n} \sum_{i=1}^n d\mathbf{h}_i d\mathbf{h}_i^T, \quad \mathbf{h}_i = (x_i^p, x_i^{p-1}y_i, \dots, x_i, y_i, z_i)^T,$$

and  $d\mathbf{h}_i$  is the Jacobian matrix of  $\mathbf{h}_i$  with respect to  $(x_i, y_i, z_i)$ . Minimizing (18) then amounts to finding the eigenvector corresponding to the minimum eigenvalue of the pencil  $\mathcal{M} - \lambda\mathcal{N}$ .

While it is popular, this generalized eigenvalue fit is still biased. Indeed, if a data point  $\mathbf{p}_i$  is close to a critical point of the polynomial  $f$ , i.e.  $\nabla f(\mathbf{p}_i) \approx 0$ , but such that  $f(\mathbf{p}_i) \neq 0$ , then the ratio  $f(\mathbf{p}_i)^2 / \|\nabla f(\mathbf{p}_i)\|^2$  becomes large. In addition, much better fitting results are obtained with the exact Euclidean distance (at the cost of larger computation times). This led Taubin [1993] to introduce higher-order approximations  $\delta_i$  to the true distance function which have the nice property of being upper bounded by the true distance in the vicinity of a regular point of  $Z(f)$ , i.e.

$$0 \leq \delta_i(\mathbf{p}, Z(f)) \leq d(\mathbf{p}, Z(f)).$$

Non-linear least-squares problems can also be solved using iterative optimization techniques such as the Gauss-Newton or Levenberg-Marquardt methods [Press et al. 1988;

Björk 1996]. Suppose that the goal is to minimize

$$E(\mathbf{s}) = \frac{1}{n} \sum_{i=1}^n f(\mathbf{p}_i, \mathbf{s})^2,$$

with  $\mathbf{s} = (s_1, \dots, s_v)$  the unknown parameter vector. Iterative techniques try to find a sequence  $\mathbf{s}^{(j)}$  of parameter vectors such that

$$E(\mathbf{s}^{(j+1)}) < E(\mathbf{s}^{(j)}) \quad \text{for all iteration index } j.$$

The general form of the sequence is:

$$\mathbf{s}^{(j+1)} = \mathbf{s}^{(j)} + \lambda_j \mathbf{d}_j, \quad (19)$$

where  $\lambda_j$  is a step size in a direction called the *displacement direction*  $\mathbf{d}_j$ .

The Gauss-Newton's method uses the Hessian matrix  $\mathcal{H}$  of  $E$  multiplied by the negative gradient  $-\nabla E$  as the displacement direction, i.e.  $\mathbf{d}_j = -\mathcal{H}^{(j)} \nabla E^{(j)}$  and size  $\lambda_j = 1$ . The first-order derivatives of  $E$  are:

$$\frac{\partial E}{\partial s_k}(\mathbf{s}) = \frac{2}{n} \sum_{i=1}^n f(\mathbf{p}_i, \mathbf{s}) \frac{\partial f}{\partial s_k}(\mathbf{p}_i, \mathbf{s}), \quad k = 1, \dots, v. \quad (20)$$

Define the matrix  $\mathcal{F}$  as follows:

$$\mathcal{F} = \begin{pmatrix} \frac{\partial f}{\partial s_1}(\mathbf{p}_1, \mathbf{s}) & \cdots & \frac{\partial f}{\partial s_v}(\mathbf{p}_1, \mathbf{s}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial s_1}(\mathbf{p}_n, \mathbf{s}) & \cdots & \frac{\partial f}{\partial s_v}(\mathbf{p}_n, \mathbf{s}) \end{pmatrix}.$$

Usually,  $n$  is much larger than  $v$  and  $\mathcal{F}$  is not a square matrix. Eq. (20) can be rewritten as:

$$\nabla E = \frac{2}{n} \mathcal{F}^T \mathbf{f},$$

where  $\nabla E$  is the gradient of  $E$  and  $\mathbf{f} = (f(\mathbf{p}_1, \mathbf{s}), \dots, f(\mathbf{p}_n, \mathbf{s}))^T$ . The second-order derivatives of  $E$  are:

$$\frac{\partial^2 E}{\partial s_k \partial s_l}(\mathbf{s}) = \frac{2}{n} \sum_{i=1}^n \left( \frac{\partial f}{\partial s_k}(\mathbf{p}_i, \mathbf{s}) \frac{\partial f}{\partial s_l}(\mathbf{p}_i, \mathbf{s}) + f(\mathbf{p}_i, \mathbf{s}) \frac{\partial^2 f}{\partial s_k \partial s_l}(\mathbf{p}_i, \mathbf{s}) \right), \quad \begin{matrix} k = 1, \dots, v \\ l = 1, \dots, v \end{matrix} \quad (21)$$

In this equation,  $f(\mathbf{p}_i, \mathbf{s})$  is a residual term which can be assumed to be small when  $\mathbf{s}$  is not too far from the optimum value. Ignoring the rightmost term, Eq. (21) becomes

$$\frac{\partial^2 E}{\partial s_k \partial s_l}(\mathbf{s}) \approx 2 \sum_{i=1}^n \frac{\partial f}{\partial s_k}(\mathbf{p}_i, \mathbf{s}) \frac{\partial f}{\partial s_l}(\mathbf{p}_i, \mathbf{s}),$$

which can be rewritten as:

$$\mathcal{H} \approx \frac{2}{n} \mathcal{F}^T \mathcal{F},$$

where  $\mathcal{H}$  is the Hessian matrix of  $E$ . Thus, Eq. (19) for Gauss-Newton becomes

$$\mathbf{s}^{(j+1)} - \mathbf{s}^{(j)} = \left( \mathcal{F}^{(j)T} \mathcal{F}^{(j)} \right)^{-1} \mathcal{F}^{(j)} \mathbf{f}^{(j)},$$

where  $\mathcal{F}^{(j)}$  means that  $\mathcal{F}$  is evaluated at  $\mathbf{s}^{(j)}$ .

There are two problems with this basic optimization scheme. First, the search sequence may not converge, i.e.  $E(\mathbf{s}^{(j+1)}) > E(\mathbf{s}^{(j)})$  for some  $j$ . Second, the matrix  $\mathcal{F}^{(j)T} \mathcal{F}^{(j)}$  may be nearly singular. One way of avoiding these difficulties is to consider the following modified scheme:

$$\mathbf{s}^{(j+1)} - \mathbf{s}^{(j)} = \left( \mathcal{F}^{(j)T} \mathcal{F}^{(j)} + l_j \mathcal{I} \right)^{-1} \mathcal{F}^{(j)} \mathbf{f}^{(j)},$$

where  $l_j$  is some positive scalar and  $\mathcal{I}$  the unit matrix of size  $v$ . The idea then is to start with a low value for  $l_j$ , say  $l_j = 0.001$ , and to adjust it at each iteration step depending on the outcome, i.e. increase it if  $E(\mathbf{s}^{(j+1)}) > E(\mathbf{s}^{(j)})$  and decrease it otherwise. This modified Gauss-Newton algorithm is known as the Levenberg-Marquardt algorithm.

## 6.2 Quadric surface fitting

Surface fitting has been used for the recovery of quadric surfaces for a number of years. Most past research has focused on standard (linear or non-linear) least-squares techniques.

**6.2.1 General quadric fitting.** Bolles and Fischler [1981] describe the RANSAC (Random Sample Consensus) technique for fitting surfaces to noisy data. To deal with the noise inherent in light-stripe range data, an initial filtering eliminates gross errors, after which a fit is computed by way of a standard least-squares technique. RANSAC is applied to finding cylinders. Hall et al. [1982] discuss the application of least-squares fitting for the extraction of quadric surface parameters from depth maps. Faugeras and Hebert [1986] use the algebraic distance for fitting planes and quadrics. Bolle and Cooper [1986] and Flynn and Jain [1988] apply iterative non-linear optimization with the true Euclidean distance on specific representations of quadrics. Local curvature estimates are used to form initial estimates of the geometric parameters of the surface. Boggs et al. [1987] describe a strategy for non-linear orthogonal regression based on the Levenberg-Marquardt technique. The algorithm uses a specific implicit quadric equation for each surface type. Dai and Newman [1998; 1999] present two methods for the fitting of hyperboloids and paraboloids, which frequently appear in industrial parts at the junction of two surface elements. The first method is based on the least-squares technique. The second, called parameter optimization, is type-specific and more geometric. It is an iterative technique which directly tries to estimate the parameters of the quadric (i.e., the lengths of the two real axes and of the imaginary axis if the quadric is an hyperboloid) starting from an initial estimate given by the user.

Various constraints can be used to restrict the search space. For example, one can require that the vector of parameters of the surface have unit norm. But this construction is not invariant under rigid transformation. An alternative is to impose the constraint that the sum of the squares of the degree 2 coefficients be equal to 1, which is invariant. This is discussed in [Faugeras and Hebert 1986].

Few of the above techniques perform correctly if the level of noise is high or if the data points are sampled from a small area. Cao et al. [1994] advocate using an approximate orthogonal distance regression. To avoid directly minimizing a non-linear function, the distance is approximated by testing along several discrete directions in 3D space. Then the minimum of these distances is used as an approximation of the true distance. The sum of squares of approximate distances can be thought of as a function of the surface parameters. It is minimized using an iterative method by adjusting the parameter estimates. With this algorithm, the authors report improvements over conventional least-squares techniques on

the goodness of fit, on the effects of noise and the distribution of the data and in terms of efficiency.

There is currently little consensus on what is the best way of fitting general quadrics. Clearly, much work remains to be done. We have however interesting insights from people who have worked on conic fitting (conics are in 2D what quadrics are in 3D). As a result of comprehensive tests on several 2D conic fitting algorithms, Fitzgibbon and Fisher [1995] found the generalized eigenvalue fit to provide the best tradeoff between speed and accuracy. A similar evaluation has not been performed for 3D fitting algorithms, but because of the simple analogy between conics and quadrics, there are good chances that this observed result will extend to higher dimensions.

**6.2.2 Type-specific fitting.** Clearly, there are situations where the type of the curved surface underlying a point set is not at all obvious. For instance, Karras [1992] reports that the surface best fitting the abdominal area during pregnancy consistently turns out to be a two-sheeted hyperboloid, something which could not be foreseen. In such cases, no constraint can be imposed on the quadric and a general fitting method must be used.

In other situations, it may be important to do type-specific fitting, i.e. to look directly for surfaces of a certain kind. Indeed, if a scene is known to contain cylinder or cone patches for instance, then it is probably not a good idea to use a general linear least-squares method to recover them. The reason is that the subspaces of  $\mathbb{R}^9$  corresponding to the different quadric types do not all have the same dimension: hyperboloids of one and two sheets and ellipsoids make up subspaces of dimension 9, cones and paraboloids (elliptic and hyperbolic) make up subspaces of dimension 8, hyperbolic and elliptic cylinders make up subspaces of dimension 7 and parabolic cylinders make up a subspace of dimension 6. For even more specific primitives (of the type people want to recover in reverse engineering), the dimension drops further: right circular cones (dimension 6), right circular cylinders (dimension 5) and spheres (dimension 4). Thus, if a general least-squares method is used, the “chance” that it will correctly recover, say, cylinders are low and the solution may be very different from the optimal surface. Also, according to [Fitzgibbon et al. 1999], type-specific fitting has many advantages in terms of occlusion and noise sensitivity: it avoids undue oscillations of the fitted surface owing to the presence of data noise. In addition, the increased stability of the algorithm widens its scope of application to cases where the data is not strictly, say, elliptical but needs to be minimally represented by an elliptical “blob” (see, e.g., [Banegas et al. 1999]).

To detect spheres, the simplest solution is to define  $f$  as

$$f(\mathbf{p}) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2,$$

where  $\mathbf{p} = (x, y, z)$ ,  $(x_0, y_0, z_0)$  is the center of the sphere and  $r$  is its radius. Setting  $\eta = x_0^2 + y_0^2 + z_0^2 - r^2$  turns the minimization of

$$\sum_i f(\mathbf{p}_i)^2$$

into a linear least-squares problem. Pratt [1987] proposes a different representation which is better behaved when the data points are unevenly distributed and in small number. He minimizes the expression

$$\sum_i (A(x_i^2 + y_i^2 + z_i^2) + Dx_i + Ey_i + Fz_i + G)^2$$

subject to the condition  $D^2 + E^2 + F^2 - 4AG = 1$ . This leads to an eigenvalue problem. For more on the influence of data point distribution and representation for sphere fitting, see [Bourdet et al. 1993].

For circular cylinders and cones, no linear least-squares method is known, the reason being that the equations expressing the conditions for a quadric to be a circular cylinder or cone are quadratic [Bolle and Cooper 1986]. To recover circular cylinders, a possible representation is:

$$f(\mathbf{p}) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - (a_x(x - x_0) + a_y(y - y_0) + a_z(z - z_0))^2 - r^2,$$

where  $(x_0, y_0, z_0)$  is an arbitrary point on the axis of the cylinder,  $(a_x, a_y, a_z)$  is a unit vector along the axis and  $r$  is the radius. For circular cones, take

$$f(\mathbf{p}) = [(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2] \cos^2 \alpha - (a_x(x - x_0) + a_y(y - y_0) + a_z(z - z_0))^2,$$

where  $(x_0, y_0, z_0)$  is the apex of the cone,  $(a_x, a_y, a_z)$  is a unit vector defining the orientation of the cone axis and  $\alpha$  is the semi-vertical angle. Other representation functions are possible (see [Werghi et al. 1998]).

Several authors have reported that the choice of representation function may severely influence the behavior of a non-linear fitting algorithm. For instance, Rosin [1996], in the context of ellipse fitting, shows that choosing  $f$  carelessly can lead to biased estimates for  $\mathbf{s}$ . More generally, the choice of distance function has a deep impact on the outcome of least-squares methods. This led Lukács et al. [1998] to consider approximations of the Euclidean distance  $d$  which are specific to each quadric type and have none of the singularities of  $d$  (i.e., points at which it is not differentiable). Call an approximation  $\tilde{d}$  of  $d$  *faithful* if  $\tilde{d}$  is zero wherever  $d$  is zero and the derivatives of  $\tilde{d}$  are equal to those of  $d$  at these points. If  $d$  has the form

$$d = \sqrt{g} - h,$$

then the following approximation is faithful:

$$\tilde{d} = \frac{g - h^2}{2h}.$$

Lukács et al. [1998] introduce faithful distances for spheres, circular cylinders and cones and tori and parameterizations of these surfaces which allow faithful distances to be used.

Consider the case of the right circular cylinder. Let  $\rho\mathbf{b}$  be the closest point of the cylinder to the origin, with  $\|\mathbf{b}\| = 1$ . Assume that  $\mathbf{a}$  is a unit vector along the axis of the cylinder and  $1/k$  the radius of the cylinder. The vector of parameters is  $\mathbf{s} = (\mathbf{a}, \mathbf{b}, k, \rho)$ . Note that it is only 5-dimensional, since  $\mathbf{a} \cdot \mathbf{b} = 0$ . The Euclidean distance from an arbitrary point  $\mathbf{p}$  in space to the cylinder is:

$$d(\mathbf{p}, \mathbf{s}) = \left\| \left( \mathbf{p} - \left( \rho + \frac{1}{k} \right) \mathbf{b} \right) \times \mathbf{a} \right\| - \frac{1}{k}. \quad (22)$$

This expression is of the form  $\sqrt{g} - h$ . This leads to the following faithful approximation of  $d$ :

$$\tilde{d}(\mathbf{p}, \mathbf{s}) = \frac{k}{2} \left\| (\mathbf{p} - \rho\mathbf{b}) \times \mathbf{a} \right\|^2 - (\mathbf{p} - \rho\mathbf{b}) \cdot \mathbf{b}.$$

This expression is better behaved than the true distance when  $k$  is small: (22) involves the subtraction of two large quantities, which may be numerically unstable. In the limit case  $k \rightarrow 0$ ,  $\tilde{d}$  becomes  $\rho - \mathbf{p} \cdot \mathbf{b}$  and the problem reduces to a linear least-squares fitting of planes. The non-linear fitting problem is then solved with some iterative technique like the Levenberg-Marquardt algorithm. A good initial estimate is required. Lukács et al. [1998] also show how to compute the derivatives of  $\tilde{d}$  with respect to  $\rho, k$  and the angles parameterizing  $\mathbf{a}$  and  $\mathbf{b}$ , which are needed by the optimization algorithm.

A common characteristic of all the above works is that they treat each surface to be fitted separately. When the points to be interpolated cover a small area and the data is corrupted by measurement noise, the fitting technique fails to perform reasonably, resulting in a highly biased estimate and a quadric which does not reflect the actual type of the underlying surface. Werghi et al. [1998; 1999; 2000] advocate compensating the poorness of information embodied in the data by extra knowledge about the surfaces such as their type and relationships with neighboring quadrics. The overall idea is to incorporate specific constraints into the reconstruction process and then use least-squares techniques to solve the semi-global fitting. The additional information, which may take the form of geometric or topological constraints (surfaces are parallel, they intersect in a right angle, ...), can be generated by a computer program based on statistical tests and either accepted or rejected by the user. Looking for such constraints makes a lot of sense in reverse engineering and manufactured object modeling where the majority of parts are designed with intended feature relationships.

Consider a set of  $l$  surface patches assumed to be quadratic. The minimization criterion for surface  $j$ , considering the algebraic distance, has the form:

$$E_j = \sum_{i=1}^{n_j} f_j(\mathbf{p}_i^j, \mathbf{s}_j)^2.$$

Globally, one wants to minimize

$$E = E_1 + \dots + E_l = \mathbf{s}^T \mathcal{H} \mathbf{s},$$

where  $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_l)$ , subject to a set of constraints incorporating the shape characteristics of the surfaces and their relationships:  $H_j(\mathbf{s}) = 0, j = 1, \dots, k$ . The problem can be seen as a constrained optimization problem which is well behaved if the constraint functions are continuous, differentiable and convex [Fletcher 1987]. The estimation of the parameter vector  $\mathbf{s}$  is achieved with a sequential unconstrained technique. Consider the following optimization function:

$$J(\mathbf{s}) = \mathbf{s}^T \mathcal{H} \mathbf{s} + \sum_{j=1}^k \lambda_j H_j(\mathbf{s}), \quad (23)$$

where the second term is a penalty function. The algorithm increments sequentially the set of weights  $\lambda_j$  and at each step (23) is minimized with the Levenberg-Marquardt technique, giving an updated  $\mathbf{s}$ . The initial value of  $\mathbf{s}$  is determined by estimating each  $\mathbf{s}_j$  individually with a generalized eigenvalue fit and then concatenating all the vectors into one. The algorithm stops when the constraints are satisfied to the desired degree or when the parameter vector remains stable for a number of iterations.

## 7. CONCLUSION

This paper has surveyed the methods for recovering quadrics in triangle meshes. We have successively delved into local geometry estimation, mesh denoising, segmentation and reconstruction, which constitute the four major steps in shape recovery algorithms.

One of the major observations of this guided tour is that there is still a lot of room for improvement in quadric surface recovery methods. Some subtasks seem more mature than others, but more work needs to be done in all of them before a reliable and automatic quadric extraction can be achieved:

—*Estimation and denoising*: Many different methods have been proposed for estimating the local surface geometry at points of a triangle mesh. Several of them, especially those which are natural equivalents in the discrete setting of formulas in the continuous case, seem to give meaningful results. What is needed now is a global comparison of those methods, to understand how they behave under varying noise levels, to assert the influence of mesh sampling regularity, mesh connectivity, scaling. . .

For images acquired with laser range devices or coordinate-measuring machines, fairing is necessary before estimating local surface differential parameters. Future work should be devoted to deciding what is the best and most efficient way of denoising meshes while preserving salient features (e.g., sharp edges).

—*Segmentation and classification*: Segmentation is probably the subtask of shape recovery methods with the smallest level of consensus and the largest number of open research directions. Future work should single out a number of segmentation strategies applicable to general piecewise-linear surfaces (and not just to range images), continue curve-surface segmenters comparison as initiated by [Powell et al. 1998], identify the most interesting features (and those that can be most reliably extracted) for the computation of the initial segmentation of region-growing methods and incorporate more domain knowledge in the segmentation process (for instance, since natural quadrics are the most present in mechanical pieces, a sequential type-specific fitting of cones, cylinders and spheres is a better idea for the reverse engineering of such pieces than an unguided, unconstrained segmentation).

—*Reconstruction*: Most known methods for fitting quadrics to segmented 3D data are based on the least-squares technique. However, least-squares estimation is not robust to noise or outliers since the larger the residual the more important is its influence on the estimate. Robust fitting should thus be a major concern of the future [Miller and Stewart 1996]. Already, people have drawn conclusions on the use of least-median-of-squares fitting, which performs better if outliers are present in the data [Chivate et al. 1994]. Experiments should also be made with M-estimators. Let  $r_i$  be the residual of the  $i$ -th datum. Instead of minimizing  $\sum_i r_i^2$ , the M-estimators try to reduce the effect of outliers by replacing the squared residuals  $r_i^2$  by another function, yielding

$$\min \sum_i \rho(r_i),$$

where  $\rho$  is a continuous, symmetric function with minimum value at zero. The above is equivalent to solving

$$\sum_i \rho'(r_i) \frac{\partial r_i}{\partial s_k}, \quad k = 1, \dots, v.$$

In other words, the derivative of  $\rho$  may be seen as a weighting or influence function. For traditional least-squares fitting,  $\rho'$  is a linear function, so the influence of a datum on the estimate increases linearly with the size of its error, confirming the non-robustness of the technique. A better estimation is obtained for instance with the Cauchy M-estimator:

$$\rho(x) = \frac{c^2}{2} \log \left( 1 + \left( \frac{x}{c} \right)^2 \right).$$

With this function, the more the points deviate from the model, the less importance they have in the calculation (i.e.,  $\rho'$  decreases as  $x$  increases).

The development of computationally efficient algorithms for the type-specific fitting of quadrics other than the natural quadrics is also an important research direction.

#### ACKNOWLEDGMENT

Thanks are due to Eric Wies for his assistance in preparing this paper and his help in identifying all the relevant literature.

#### REFERENCES

- ABDELMALEK, N. 1990. Surface curvatures and 3D range images segmentation. *Pattern Recognition* 23, 8, 807–817.
- ALBOUL, L. AND VAN DAMME, R. 1996. Polyhedral metrics in surface reconstruction. In *The Mathematics of Surfaces VI*, G. Mullineux, Ed. Oxford University Press, 171–200.
- ALONSO, L., CUNY, F., PETITJEAN, S., PAUL, J.-C., LAZARD, S., AND WIES, E. 2001. The virtual mesh: A geometric abstraction for efficiently computing radiosity. *ACM Trans. Graphics* 20, 3, 169–201.
- ANGELOPOULOU, E. AND WÖLFF, L. 1998. Sign of Gaussian curvature from curve orientation in photometric space. *IEEE Trans. Pattern Analysis and Machine Intelligence* 20, 10, 1056–1066.
- ARMAN, F. AND AGGARWAL, J. 1993. Model-based object recognition in dense-range images – A review. *ACM Comput. Surv.* 25, 1, 5–43.
- BAJAJ, C., IHM, I., AND WARREN, J. 1993. Higher-order interpolation and least-squares approximation using implicit algebraic surfaces. *ACM Trans. Graphics* 12, 4, 327–347.
- BANEGAS, F., MICHELUCCI, D., ROELEN, M., AND JAEGER, M. 1999. Automatic extraction of significant features from 3D point clouds by ellipsoidal skeleton. In *Proc. of International Conference on Visual Computing, Goa, India*. 58–67.
- BELYAEV, A. AND OHTAKE, Y. 2000. An image processing approach to detection of ridges and ravines on polygonal surfaces. In *Proc. of Eurographics*. Short presentation.
- BERKMANN, J. AND CAELLI, T. 1994. Computation of surface geometry and segmentation using covariance techniques. *IEEE Trans. Pattern Analysis and Machine Intelligence* 16, 11, 1114–1116.
- BESL, P. AND JAIN, A. 1986. Invariant surface characteristics for 3D object recognition in range images. *Computer Vision, Graphics and Image Processing* 33, 33–80.
- BESL, P. AND JAIN, A. 1988. Segmentation through variable-order surface fitting. *IEEE Trans. Pattern Analysis and Machine Intelligence* 10, 2, 167–192.
- BJÖRK, Å. 1996. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics.
- BLINN, J. 1997. The algebraic properties of second-order surfaces. In *Introduction to Implicit Surfaces*, J. Bloomenthal, Ed. Morgan Kaufmann, 52–97.
- BOCK, M. AND GUERRA, C. 1999. A geometric approach to the segmentation of range images. In *Proc. of 2nd International Conference on 3-D Imaging and Modeling, Ottawa, Canada*. 261–269.
- BOGGS, P., BYRD, R., AND SCHNABEL, R. 1987. A stable and efficient algorithm for non-linear orthogonal distance regression. *SIAM J. Scientific and Statistical Computing* 8, 6, 177–188.
- BOISSONNAT, J.-D. AND CAZALS, F. 2000. Smooth surface reconstruction via natural neighbor interpolation of distance functions. In *Proc. of ACM Symposium on Computational Geometry*. 223–232.

- BOLLE, R. AND COOPER, D. 1986. On optimally combining pieces of information, with application to estimating 3-D complex-object position from range data. *IEEE Trans. Pattern Analysis and Machine Intelligence* 8, 5, 619–638.
- BOLLES, R. AND FISCHLER, M. 1981. A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In *Proc. of International Joint Conference on Artificial Intelligence*. 637–643.
- BOOKSTEIN, F. 1979. Fitting conic sections to scattered data. *Computer Vision, Graphics, and Image Processing* 9, 56–71.
- BOULANGER, P. AND RIOUX, M. 1987. Segmentation of planar and quadric surfaces. In *SPIE Proceedings, Advances in Intelligent Robotics Systems, Boston, MA*. 848–861.
- BOURDET, P., LARTIGUE, C., AND LEVEAUX, F. 1993. Effects of data point distribution and mathematical model on finding the best-fit sphere to data. *Precision Engineering* 15, 3, 150–157.
- BRADY, M., PONCE, J., YUILLE, A., AND ASADA, H. 1985. Describing surfaces. *Computer Vision, Graphics and Image Processing* 32, 1–28.
- BRICAULT, I. AND MONGA, O. 1997. From volume medical images to quadratic surface patches. *Computer Vision and Image Understanding* 67, 1, 24–38.
- BRUCE, J., GIBLIN, P., AND TARI, F. 1996. Ridges, crests and sub-parabolic lines of evolving surfaces. *Int. J. Computer Vision* 18, 3, 195–210.
- CAI, L. 1989. A “small leakage” model for diffusion smoothing of image data. In *Proc. of International Joint Conference on Artificial Intelligence*. 1585–1590.
- CAO, X. AND SHRIKHANDE, N. 1991. Quadric surface fitting for sparse range data. In *Proc. of IEEE/SMC International Conference on Systems, Man and Cybernetics*. 123–128.
- CAO, X., SHRIKHANDE, N., AND HU, G. 1994. Approximate orthogonal distance regression method for fitting quadric surfaces to range data. *Pattern Recognition Letters* 15, 8, 781–796.
- CHAI, R., BOUAKAZ, S., AND VANDORPE, D. 1999. Extraction de descripteurs surfaciques pour la reconstruction 3D. *Revue de CFAO et d’informatique graphique* 14, 1, 93–107.
- CHEN, X. AND SCHMITT, F. 1992. Intrinsic surface properties from surface triangulation. In *Proc. of European Conference on Computer Vision*. 739–743.
- CHIVATE, N., GOODALL, C., AND JABLOKOW, A. 1994. Comparative evaluation of surface fitting techniques for implicit surfaces. In *Joint Statistical Meetings, Toronto*.
- CHIVATE, P. AND JABLOKOW, A. 1993. Solid-model generation from measured point data. *Computer-Aided Design* 25, 9, 587–600.
- DAI, M. AND NEWMAN, T. 1998. Hyperbolic and parabolic quadric surface fitting algorithms – Comparison between the least squares approach and the parameter optimization approach. Tech. Rep. TR-UAH-CS-1998-02, University of Alabama.
- DAI, M. AND NEWMAN, T. 1999. Accurate reference surface recovery and defect detection for hyperboloid and paraboloid surfaces. In *Proc. of 5th International Conference on Quality Control by Artificial Vision (Quebec, Canada)*. 165–170.
- DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH 99 Conference Proceedings*. Annual Conference Series. Addison Wesley, 317–324.
- DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. 2000. Discrete differential-geometry operators in  $n$ D. Preprint, The Caltech Multi-Res Modeling Group.
- DO CARMO, M. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, New Jersey.
- EGGERT, D., FITZGIBBON, A., AND FISHER, R. 1998. Simultaneous registration of multiple range views for use in reverse engineering of CAD models. *Computer Vision and Image Understanding* 69, 3, 253–272.
- FAN, T., MEDIONI, G., AND NEVATIA, R. 1987. Segmented descriptions of 3D surfaces. *IEEE J. Robotics and Automation* 3, 6, 527–538.
- FAN, T., MEDIONI, G., AND NEVATIA, R. 1989. Recognizing 3D objects using surface descriptions. *IEEE Trans. Pattern Analysis and Machine Intelligence* 11, 11, 1140–1157.
- FAUGERAS, O. AND HEBERT, M. 1986. The representation, recognition, and locating of 3-D objects. *The International Journal of Robotics Research* 5, 3, 27–51.

- FAUGERAS, O., HEBERT, M., AND PAUCHON, E. 1983. Segmentation of range data into planar and quadric patches. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 8–13.
- FERRIE, F., LAGARDE, J., AND WHAITE, P. 1993. Darboux frames, snakes, and super-quadrics: geometry from the bottom up. *IEEE Trans. Pattern Analysis and Machine Intelligence* 15, 8, 771–784.
- FERRIE, F., MATHUR, S., AND SOUCY, G. 1993. Feature extraction for 3-D model building and object recognition. In *Three-Dimensional Object Recognition Systems*, P. Flynn and A. Jain, Eds. Elsevier Press, 57–88.
- FISHER, R., FITZGIBBON, A., AND EGGERT, D. 1997. Extracting surface patches from complete range descriptions. In *Proc. of International Conference on Recent Advances in 3-D Imaging and Modeling, Ottawa, Canada*. 148–155.
- FITZGIBBON, A., EGGERT, D., AND FISHER, R. 1997. High-level CAD model acquisition from range images. *Computer-Aided Design* 29, 4, 321–330.
- FITZGIBBON, A. AND FISHER, R. 1993. Invariant fitting for arbitrary single-extremum surfaces. In *Proc. of British Machine Vision Conference, Surrey*. 569–578.
- FITZGIBBON, A. AND FISHER, R. 1995. A buyer's guide to conic fitting. In *Proc. of British Machine Vision Conference, Birmingham*. 513–522.
- FITZGIBBON, A., PILU, M., AND FISHER, R. 1999. Direct least squares fitting of ellipses. *IEEE Trans. Pattern Analysis and Machine Intelligence* 21, 5, 476–480.
- FLETCHER, R. 1987. *Practical Methods of Optimization*. John Wiley & Sons.
- FLYNN, P. AND JAIN, A. 1988. Surface classification: hypothesis testing and parameter estimation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 261–267.
- FLYNN, P. AND JAIN, A. 1989. On reliable curvature estimation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 110–116.
- FLYNN, P. AND JAIN, A. 1991. CAD-based computer vision: from CAD models to relational graphs. *IEEE Trans. Pattern Analysis and Machine Intelligence* 13, 2, 114–132.
- FOLEY, J., VAN DAM, A., FEINER, S., AND HUGHES, J. 1992. *Computer Graphics, Principles and Practice*. Addison-Wesley, Reading, MA.
- GOPI, M., KRISHNAN, S., AND SILVA, C. 2000. Surface reconstruction based on lower dimensional localized Delaunay triangulation. *Computer Graphics Forum* 19, 3, 467–478. Proc. of Eurographics.
- GOURAUD, H. 1971. Continuous shading of curved surfaces. *IEEE Trans. Computers* 20, 6, 623–629.
- GRIMSON, W., LOZANO-PEREZ, T., NOBLE, N., AND WHITE, S. 1993. An automatic tube inspection system that finds cylinders in range data. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition, New York*. 446–452.
- HALL, E., TIO, J., MCPHERSON, C., AND SADJADI, F. 1982. Measuring curved surfaces for robot vision. *IEEE Computer* 15, 12, 42–54.
- HAMANN, B. 1993. Curvature approximation for triangulated surfaces. *Computing Suppl.* 8, 139–153.
- HAN, J., VOLZ, R., AND MUDGE, T. 1987. Range image segmentation and surface parameter extraction for 3D object recognition of industrial parts. In *Proc. of International Conference on Robotics and Automation*. 1582–1589.
- HEBERT, M. AND PONCE, J. 1982. A new method for segmenting 3D scenes into primitives. In *Proc. of International Conference on Pattern Recognition*. 836–838.
- HECKBERT, P. AND GARLAND, M. 1999. Optimal triangulation and quadric-based surface simplification. *Journal of Computational Geometry: Theory and Applications* 14, 1–3, 49–65.
- HILTON, A., ILLINGWORTH, J., AND WINDEATT, T. 1995. Statistics of surface curvature estimates. *Pattern Recognition* 28, 8, 1201–1221.
- HOFFMAN, R. AND JAIN, A. 1987. Segmentation and classification of range images. *IEEE Trans. Pattern Analysis and Machine Intelligence* 9, 5, 608–620.
- HOOVER, A., JEAN-BAPTISTE, G., JIANG, X., FLYNN, P., BUNKE, H., GOLDFOF, D., BOWYER, K., EGGERT, D., FITZGIBBON, A., AND FISHER, R. 1996. An experimental comparison of range image segmentation algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence* 18, 7, 673–689.
- HOPPE, H., DEROSE, T., DUCHAMP, T., McDONALD, J., AND STUETZLE, W. 1992. Surface reconstruction from unorganized points. In *SIGGRAPH 92 Conference Proceedings*. Annual Conference Series. Addison Wesley, 71–78.

- HOSCHEK, J., DIETZ, U., AND WILKE, W. 1998. A geometric concept of reverse engineering of shape: approximation and feature lines. In *Mathematical Methods for Curves and Surfaces II*, M. Daehlen, T. Lyche, and L. Schumaker, Eds. Vanderbilt University Press, 253–262.
- JAIN, A. AND HOFFMAN, R. 1988. Evidence-based recognition of 3D objects. *IEEE Trans. Pattern Analysis and Machine Intelligence* 10, 6, 783–802.
- JIANG, X. AND BUNKE, H. 1999. Edge detection in range images based on scan line approximation. *Computer Vision and Image Understanding* 73, 2, 183–199.
- JONES, G. AND ILLINGWORTH, J. 1994. Robust segmentation of curved surfaces from range data. In *Proc. of 7th International Conference on Progress in Image Analysis*. 453–456.
- KARBACHER, S. AND HÄUSLER, G. 1998. A new approach for modeling and smoothing of scattered 3D data. In *Three-Dimensional Image Capture and Applications*, R. Ellson and J. Nurre, Eds. SPIE Proceedings, vol. 3313. The International Society for Optical Engineering, 168–177.
- KARRAS, G. 1992. On the orientation of digital elevation models in biostereometrics. In *Surface Topography and Spinal Deformity*, A. Alberti, B. Drerup, and E. Hierholzer, Eds. G. Fischer Verlag, Stuttgart, 162–165.
- KOBBELT, L. 1997. Discrete fairing. In *The Mathematics of Surfaces VII*, T. Goodman, Ed. Oxford University Press, 101–131.
- KOENDERINK, J. 1990. *Solid Shape*. MIT Press, Cambridge, MA.
- KÓS, G., MARTIN, R., AND VÁRADY, T. 2000. Methods to recover constant radius rolling ball blends in reverse engineering. *Computer-Aided Geometric Design* 17, 127–160.
- KÖSTER, K. AND SPANN, M. 2000. MIR: an approach to robust clustering – Application to range image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 22, 5, 430–444.
- KRSEK, P., LUKÁCS, G., AND MARTIN, R. 1998. Algorithms for computing curvatures from range data. In *The Mathematics of Surfaces VIII*, R. Cripps, Ed. Information Geometers, 1–16.
- KRSEK, P., PAJDLA, T., AND HLAVAC, V. 1997. Estimation of differential structures on triangulated surfaces. In *Proc. of the 21st Workshop of the Austrian Association for Pattern Recognition*, R. Oldenbourg, Ed. 157–164.
- LAGARDE, J. 1997. Constraints and their satisfaction in the recovery of local surface structure. M.S. thesis, Department of Electrical Engineering, McGill University.
- LANG, V., BELYAEV, A., BOGAEVSKI, I., AND KUNII, T. 1997. Fast algorithms for ridge detection. In *Proc. of International Conference on Shape Modeling and Applications, Japan*.
- LEE, C.-K., HARALICK, R., AND DEGUCHI, K. 1993. Estimation of curvature from sampled noisy data. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition, New York*. 536–541.
- LEE, K.-M., MEER, P., AND PARK, R.-H. 1998. Robust adaptive segmentation of range images. *IEEE Trans. Pattern Analysis and Machine Intelligence* 20, 2, 200–205.
- LEJEUNE, A. AND FERRIE, F. 1996. Finding the parts of objects in range images. *Computer Vision and Image Understanding* 64, 2, 230–247.
- LENGAGNE, R., FUA, P., AND MONGA, O. 1996. Using crest lines to guide surface reconstruction from stereo. In *Proc. of International Conference on Pattern Recognition*. 9–13.
- LEONARDIS, A., GUPTA, A., AND BAJCSY, R. 1995. Segmentation of range images as the search for geometric parametric models. *Int. J. of Computer Vision* 14, 3, 253–277.
- LIANG, P. AND TODHUNTER, J. 1990. Representation and recognition of surface shapes in range images: a differential geometric approach. *Computer Vision, Graphics, and Image Processing* 52, 1, 78–109.
- LIN, C. AND PERRY, M. 1982. Shape description using surface triangulation. In *Proc. of IEEE Workshop on Computer Vision, Representation and Control, Rindge, NH*. 38–43.
- LÓPEZ, A., LUMBRERAS, F., SERRAT, J., AND VILLANUEVA, J. 1999. Evaluation of methods for ridge and valley detection. *IEEE Trans. Pattern Analysis and Machine Intelligence* 21, 4, 327–335.
- LUKÁCS, G. AND ANDOR, L. 1998. Computing natural division lines on free-form surfaces based on measured data. In *Mathematical Methods for Curves and Surfaces II*, M. Daehlen, T. Lyche, and L. Schumaker, Eds. Vanderbilt University Press, 319–326.
- LUKÁCS, G., MARTIN, R., AND MARSHALL, A. 1998. Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. In *Proc. of European Conference on Computer Vision*. Lecture Notes in Computer Science, vol. 1406. Springer-Verlag, 671–686. Long version: Report GML 1997/5, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest.

- MANGAN, A. AND WHITAKER, R. 1999. Partitioning 3D surface meshes using watershed segmentation. *IEEE Trans. Visualization and Computer Graphics* 5, 4, 308–321.
- MARTIN, R. 1998. Estimation of principal curvatures from range data. *Int. J. of Shape Modeling* 4, 3,4, 99–109.
- MATHUR, S. AND FERRIE, F. 1997. Edge localization in surface reconstruction using optimal estimation theory. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 833–838.
- MAX, N. 1999. Weights for computing vertex normals from facet normals. *Journal of Graphics Tools* 4, 2, 1–6.
- MCIVOR, A. 1998. A cubic facet based method for estimating the principal quadric. In *Proc. of Image and Vision Computing New Zealand, Auckland*. 136–141.
- MCIVOR, A. AND VALKENBURG, R. 1997. A comparison of local surface geometry estimation methods. *Machine Vision and Applications* 10, 1, 17–26. Long version: Industrial Research Limited Report 520, Auckland, New Zealand, March 1996.
- MCIVOR, A. AND WALTENBERG, P. 1998. Recognition of simple curved surfaces from 3D surface data. In *Proc. of Asian Conference on Computer Vision, Hong Kong*. Vol. 1. 434–441. Long version: Industrial Research Limited Report 690, Auckland, New Zealand, March 1997.
- MEDIONI, G. AND PARVIN, B. 1986. Segmentation of range images into planar surfaces by split and merge. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 415–417.
- MEEK, D. AND WALTON, D. 2000. On surface normal and Gaussian curvature approximations given data sampled from a smooth surface. *Computer-Aided Geometric Design* 17, 521–543.
- MENCL, E. AND MÜLLER, H. 1998. Interpolation and approximation of surfaces from three-dimensional scattered data points. In *State of the Art Reports, Eurographics'98*. 51–67.
- MILLER, J. AND STEWART, C. 1996. MUSE: robust surface fitting using unbiased scale estimates. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 300–306.
- MOORE, D. AND WARREN, J. 1991. Approximation of dense scattered data using algebraic surfaces. In *Proc. of IEEE Hawaiian International Conference on System Sciences*. 681–690.
- MULLER, Y. AND MOHR, R. 1984. Planes and quadrics detection using Hough transform. In *Proc. of International Conference on Pattern Recognition, Montreal*. Vol. 2. 1101–1103.
- NEWMAN, T., FLYNN, P., AND JAIN, A. 1993. Model-based classification of quadric surfaces. *Computer Vision, Graphics, and Image Processing: Image Understanding* 58, 2, 235–249.
- NGUYEN, V.-D., NZOMIGNI, V., AND STEWART, C. 1999. Fast and robust registration of 3D surfaces using low curvature patches. In *Proc. of 2nd International Conference on 3-D Imaging and Modeling, Ottawa, Canada*. 201–208.
- NOURSE, B., HAKALA, D., HILLYARD, R., AND MALRAISON, P. 1980. Natural quadrics in mechanical design. In *Proc. of Autofact West, Anaheim, CA*. Vol. 1. 363–378.
- OHTAKE, Y., BELYAEV, A., AND BOGAEVSKI, I. 2000. Polyhedral surface smoothing with simultaneous mesh regularization. In *Proc. of International Conference on Geometric Modeling and Processing, Hong Kong*. 229–237.
- OHTAKE, Y., BELYAEV, A., AND BOGAEVSKI, I. 2001. Mesh regularization and adaptive smoothing. *Computer-Aided Design* 33, 789–800.
- OSHIMA, M. AND SHIRAI, Y. 1983. Object recognition using three dimensional information. *IEEE Trans. Pattern Analysis and Machine Intelligence* 5, 4, 353–361.
- PENNEC, X., AYACHE, N., AND THIRION, J.-P. 2000. Landmark-based registration using features identified through differential geometry. In *Handbook of Medical Imaging*, I. Bankman, Ed. Academic Press.
- PINKALL, U. AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics* 2, 1, 15–36.
- PONCE, J. AND BRADY, J. 1987. Toward a surface primal sketch. In *Three-Dimensional Machine Vision*, T. Kanade, Ed. Kluwer Publishers, 195–240.
- POWELL, M., BOWYER, K., JIANG, X., AND BUNKE, H. 1998. Comparing curved-surface range image segmenters. In *Proc. of International Conference on Computer Vision*. 286–291.
- PRATT, V. 1987. Direct least-squares fitting of algebraic surfaces. In *Computer Graphics (SIGGRAPH 87 Conference Proceedings)*. Addison Wesley, 27–31.
- PRESS, W., FLANNERY, B., TENKOLSKY, S., AND VETTERLING, W. 1988. *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge University Press.

- REQUICHA, A. AND VOELCKER, H. 1982. Solid modeling: a historical summary and contemporary assessment. *IEEE Computer Graphics and Applications* 2, 1, 9–24.
- ROBERTSON, C., FISHER, R., WERGI, N., AND ASHBROOK, A. 1999. An improved algorithm to extract surfaces from complete range descriptions. In *Proc. of WMC'99 (World Manufacturing Conference)*. 592–598.
- ROSIN, P. 1996. Analysing error of fit functions for ellipses. *Pattern Recognition Letters* 17, 1461–1470.
- RÖSSL, C., KOBELT, L., AND SEIDEL, H.-P. 2000. Extraction of feature lines on triangulated surfaces using morphological operators. In *Proc. of Smart Graphics'00, AAAI Spring Symposium, Stanford University*. 71–75.
- SACCHI, R., POLIAKOFF, J., THOMAS, P., AND HÄFELE, K.-H. 1999. Curvature estimation for segmentation of triangulated surfaces. In *Proc. of 2nd International Conference on 3-D Imaging and Modeling, Ottawa, Canada*. 536–543.
- SACHS, E., ROBERTS, A., AND STOOPS, D. 1991. 3-Draw: a tool for designing 3D shapes. *IEEE Computer Graphics and Applications* 11, 6, 18–26.
- SAMSON, P. AND MALLET, J.-L. 1997. Curvature analysis of triangulated surfaces in structural geology. *Journal of Mathematical Geology* 29, 3, 391–412.
- SANDER, P. AND ZUCKER, S. 1990. Inferring surface trace and differential structure from 3-D images. *IEEE Trans. Pattern Analysis and Machine Intelligence* 12, 9, 833–854.
- SANDER, P. AND ZUCKER, S. 1992. Singularities of principal direction fields from 3D images. *IEEE Trans. Pattern Analysis and Machine Intelligence* 14, 3, 309–317.
- SANDERSON, A. 1996. Shape recovery of volume data with deformable B-spline models. Ph.D. thesis, University of Utah.
- SAPIDIS, N. AND BESL, P. 1995. Direct construction of polynomial surfaces from dense range images through region growing. *ACM Trans. Graphics* 14, 2, 171–200.
- SOUCY, G. AND FERRIE, F. 1997. Surface recovery from range images using curvature and motion consistency. *Computer Vision and Image Understanding* 65, 1, 1–18.
- STOKELY, E. AND WU, S.-Y. 1992. Surface parameterization and curvature measurement of arbitrary 3-D objects. *IEEE Trans. Pattern Analysis and Machine Intelligence* 14, 8, 833–840.
- TANG, C.-K. AND MEDIONI, G. 1998. Inference of integrated surface, curve, and junction descriptions from sparse 3D data. *IEEE Trans. Pattern Analysis and Machine Intelligence* 20, 11, 1206–1223.
- TANG, C.-K. AND MEDIONI, G. 1999. Robust estimation of curvature information from noisy 3D data for shape description. In *Proc. of International Conference on Computer Vision, Corfu, Greece*. 426–433.
- TAUBIN, G. 1991. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 13, 11, 1115–1138.
- TAUBIN, G. 1993. An improved algorithm for algebraic curve and surface fitting. In *Proc. of International Conference on Computer Vision*. 658–665.
- TAUBIN, G. 1995a. Curve and surface smoothing without shrinkage. In *Proc. of International Conference on Computer Vision*. 852–857.
- TAUBIN, G. 1995b. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proc. of International Conference on Computer Vision*. 902–907.
- TAUBIN, G. 1995c. A signal processing approach to fair surface design. In *SIGGRAPH 95 Conference Proceedings*. Annual Conference Series. Addison Wesley, 351–358.
- TAYLOR, R., SAVINI, M., AND REEVES, A. 1989. Fast segmentation of range imagery into planar regions. *Computer Vision, Graphics, and Image Processing: Image Understanding* 45, 1, 42–60.
- THIRION, J.-P. AND GOURDON, A. 1995. Computing the differential characteristics of isointensity images. *Computer Vision and Image Understanding* 61, 2, 190–202.
- THOMPSON, W., DE ST. GERMAIN, J., STARK, S., AND HENDERSON, T. 1999. Feature-based reverse engineering of mechanical parts. *IEEE Trans. Robotics and Automation* 15, 1, 57–66.
- THÜRMER, G. AND WÜTHRICH, C. 1998. Computing vertex normals from polygonal facets. *Journal of Graphics Tools* 3, 1, 43–46.
- TRUCCO, E. AND FISHER, R. 1992. Computing surface-based representations from range images. In *Proc. of IEEE International Symposium on Intelligent Control*. 275–280.

- TRUCCO, E. AND FISHER, R. 1995. Experiments in curvature-based segmentation of range data. *IEEE Trans. Pattern Analysis and Machine Intelligence* 17, 2, 177–182.
- VÁRADY, T., KÓS, G., AND BENKŐ, P. 1998. Reverse engineering regular objects: simple segmentation and surface fitting procedures. *Int. J. of Shape Modeling* 4, 3 & 4, 127–142.
- VÁRADY, T., MARTIN, R., AND COX, J. 1997. Reverse engineering of geometric models – An introduction. *Computer-Aided Design* 29, 4, 255–269.
- WATANABE, K. AND BELYAEV, A. 2000. Detection of curvature extrema on polygonal surfaces. In *Proc. of International Conference on Human and Computer*. 11–14.
- WELCH, W. AND WITKIN, A. 1994. Free-form shape design using triangulated surfaces. In *SIGGRAPH 94 Conference Proceedings*. Annual Conference Series. Addison Wesley, 247–256.
- WERGHI, N., FISHER, R., ASHBROOK, A., AND ROBERTSON, C. 1999. Object reconstruction by incorporating geometric constraints in reverse engineering. *Computer-Aided Design* 31, 6, 363–399.
- WERGHI, N., FISHER, R., AND ROBERTSON, C. 1998. Modelling objects having quadric surfaces incorporating geometric constraints. In *Proc. of European Conference on Computer Vision*. Vol. 2. 185–201.
- WERGHI, N., FISHER, R., ROBERTSON, C., AND ASHBROOK, A. 2000. Faithful recovering of quadric surfaces from 3D range data by global fitting. *Int. J. of Shape Modelling* 6, 1, 65–78.
- YANG, M. AND LEE, E. 1999. Segmentation of measured point data using a parametric quadric surface approximation. *Computer Aided Design* 31, 7, 449–458.
- YOKOYA, Y. AND LEVINE, M. 1989. Range image segmentation based on differential geometry: a hybrid approach. *IEEE Trans. Pattern Analysis and Machine Intelligence* 11, 6, 643–649.
- YOSHIMI, T. AND TOMITA, F. 1994. Robust estimation for range image segmentation and reconstruction. In *Proc. of IAPR Workshop on Machine Vision Applications, Kawasaki, Japan*. 506–509.

Received Month Year; revised month year; accepted Month Year.