Recovering Surfaces from the Restoring Force

George Kamberov[1] and Gerda Kamberova[2]

[1] Stevens Institute of Technology, Hoboken, NJ 07030, USA
kamberov@cs.stevens-tech.edu
[2] Hofstra University, Hempstead, NY 11549, USA
kamberov@grasp.cis.upenn.edu

Abstract. We present a new theoretical method and experimental results for direct recovery of the curvatures, the principal curvature directions, and the surface itself by explicit integration of the Gauss map. The method does not rely on polygonal approximations, smoothing of the data, or model fitting. It is based on the observation that one can recover the surface restoring force from the Gauss map, and (i) applies to orientable surfaces of arbitrary topology (not necessarily closed); (ii) uses only first order linear differential equations; (iii) avoids the use of unstable computations; (iv) provides tools for filtering noise from the sampled data. The method can be used for stable extraction of surfaces and surface shape invariants, in particular, in applications requiring accurate quantitative measurements.

1 Introduction

In this paper we consider a classical computer vision problem: to what extent can we determine a surface and its properties from its Gauss map. We show that given the Gauss map $\mathbf{N}$ by solving a linear system of first order differential equations we can extract the mean curvature function H of the surface without building a surface model or a surface parameterization. Then we turn around and use H and $\mathbf{N}$ to determine the Gauss curvature, the principal curvature axes, and ultimately a global parameterization of the surface. The presented method works for orientable surfaces of arbitrary topology. The method is based on the solution of first order linear differential equations and explicit quadratic expressions. This leads to computationally stable discretization of the method. A discrete version of the presented method is used to recover the curvatures, the principal axes, and parameterizations from clouds of points and normals sampled from actual surfaces.

Surface parameterizations are a convenient tool for analyzing surface properties. They are used in many computer vision tasks, for example, in matching and for the computation of surface invariants. One typical approach is to, first, collect a sufficiently dense set of sampled surface points using range sensors, stereo, MRI, or CT techniques; second, to approximate the parameterization from an estimated 3D surface model; and third, to compute the differential invariants from the approximated surface. The 3D model is obtained by applying a marching cubes method, a Delaunay triangulation, or some model fitting or smoothing technique, [6, 7, 18, 2, 14, 17, 19]. The typical approaches do not come with robust and general error estimates. Another approach is to extract the curvature invariants directly from range, stereo, or photometric data without building the 3D model [6]. Other powerful methods for obtaining global parameterizations of closed surfaces involve superquadrics, deformable superquadrics, Brechbüller's constrained optimization algorithm for surfaces with spherical topology [13, 4].

Ultimately with all these methods differential invariants are computed using classical differential geometric methods. They involve taking second order derivatives, and solving general characteristic polynomials. An additional source of errors is the computational instability of the methods. For example, to compute the principal curvature vectors and the principal curvatures, the methods rely on diagonalizing general symmetric matrices (in fact the operators are often only close to symmetric due to noise and round off errors). The standard diagonalization routines introduce additional errors.

We set out to develop a method which (i) applies to orientable surfaces of arbitrary topology (not necessarily closed); (ii) uses only first order linear differential equations; (iii) avoids the use of unstable computations; (iv) provides tools for filtering noise from the sampled data.

The principal decision is what should be the data used to derive the surface invariants and the parameterization. We chose the Gauss map of the surface as the primary data from which everything else is inferred. The motivation for this choice stems from geometry, physical intuition, and the current practice in computer vision. Furthermore, the Gauss map is a first order invariant and so takes an intermediate position between the parameterization and the curvature invariants of a surface (one derivative in each direction).

The geometric motivation will be explained further in the next section but the basic idea is that a generic surface is determined up to scaling and translation by its Gauss map. The physics intuition is based on the realization that one can obtain the mean curvature function H of a surface by solving a differential equation involving only the Gauss map $\mathbf{N}$. Thus given $\mathbf{N}$ we can get the vector field $H\mathbf{N}$. Ever since the times of Laplace and Young it is known that if we think of the surface as an isotropic membrane with constant surface tension, then up to multiplication by a constant the vector field $H\mathbf{N}$ determines completely the restoring force which shapes the surface [16]. Finally the extraction of the Gauss map of a surface is a staple of computer vision.

Motivated by these observations we take a new approach to surface and surface shape recovery. First we remove outliers using integrability conditions, then we compute the mean curvature directly from the Gauss map, and then turn around and compute as many attributes as possible before we finally recover the immersion, instead of following the usual path, recover the immersion and then compute the rest of the attributes. The point in our approach is to reduce round off errors and other numeric noise, and also to exploit other useful ingredients which one can extract directly from the data set. An alternative discrete procedure which avoids solving differential equations is presented in [10].

2 Parameterized Surfaces in $\boldsymbol{R}^3$

Here we outline the necessary background theory and terminology. For a detailed exposition see [5]. A parameterized surface, S, in space is a vector-valued map, $\mathbf{f}$, from some two-dimensional domain M into Euclidean three space:

$$\mathbf{f} : M \rightarrow \boldsymbol{R}^3, \quad S = \mathbf{f}(M).$$

The domain M is often chosen to be a planar region endowed with some coordinates (u, v) but one can use any smooth 2D manifold. The differential, $d\mathbf{f}_p$, of $\mathbf{f}$ at a point $p \in M$ is a linear map that maps tangent vectors to tangent vectors, i.e., if $\mathbf{u}$ is the velocity (tangent) vector to a curve in M, $d\mathbf{f}_p(\mathbf{u})$ is the velocity (tangent) vector to the image of that curve in $S = \mathbf{f}(M)$. Thus

$$d\mathbf{f}_p : T_p(M) \rightarrow T_{\mathbf{f}(p)}(S) \subset \boldsymbol{R}^3$$

where $T_p(M)$ denotes the tangent plane to the abstract surface M and $T_{\mathbf{f}(p)}(S)$ is the range $d\mathbf{f}_p\,(T_p(M))$ of the differential $d\mathbf{f}_p$, respectively. The tangent plane $T_p(M)$ to a surface M at a point p is the linear space, that best approximates the surface at p. It is customary to omit the subscript p when discussing the differential or the tangent plane, and so we do.

The map $\mathbf{f}$ is an immersion if its differential $d\mathbf{f}$ is an isomorphism and we say that S is an immersed parameterized surface.

2.1 Oriented Surfaces

In this paper, we consider only oriented surfaces, that is, there is a consistent way of identifying positively oriented frames in the tangent plane. (See Chapter 2-6 in [5].) Intuitively, a surface is oriented if one has chosen a counterclockwise direction of rotation in all tangent planes.

If S is an immersed parameterized surface and the domain M of the parameterization $\mathbf{f}$ is oriented, then we can define a continuous unit vector field

$$\mathbf{N} : M \rightarrow \boldsymbol{R}^3$$

such that $\mathbf{N}(p)$ is perpendicular to the plane $T_{\mathbf{f}(p)}(S) \subset \boldsymbol{R}^3$ for every point p in the domain M. The map $\mathbf{N}$ is called the Gauss map of the immersion.

The Gauss map $\mathbf{N}$ (the surface normal), the Gauss curvature, the mean curvature, and all other differential invariants are expressed in terms of the map $\mathbf{f}$ and its derivatives.

2.2 The Gauss Map

If M is an abstract oriented two dimensional manifold then the value of the Gauss map at a point $p \in M$ is defied by

$$\mathbf{N} = \frac{1}{\|d\mathbf{f}(\mathbf{v_1}) \times d\mathbf{f}\,(\mathbf{v_2})\|} d\mathbf{f}\,(\mathbf{v_1}) \times d\mathbf{f}\,(\mathbf{v_2})$$

where $(\mathbf{v}_1, \mathbf{v}_2)$ is a positively oriented frame of the tangent plane $T_p(M)$. Here $\times$ is the usual cross product in $\boldsymbol{R}^3$. In particular, if M is a planar domain with a fixed coordinate system (u, v), then

$$d\mathbf{f} = \frac{\partial \mathbf{f}}{\partial u} du + \frac{\partial \mathbf{f}}{\partial v} dv,$$

and the Gauss map is the vector-valued function

$$\mathbf{N} = \frac{1}{\|\frac{\partial \mathbf{f}}{\partial u} \times \frac{\partial \mathbf{f}}{\partial v}\|} \frac{\partial \mathbf{f}}{\partial u} \times \frac{\partial \mathbf{f}}{\partial v}. \tag{1}$$

In general, it is convenient to think of the Gauss map as a map from M to the unit sphere, $\mathbf{S^3}$,

$$\mathbf{N} : M \to \mathbf{S^3} \subset \boldsymbol{R}^3.$$

In our examples we use the fish-scales method designed by Šára and Bajcsy in [14] to extract samples of the Gauss map of surfaces in $\boldsymbol{R}^3$.

2.3 A Conformal Structure and a Complex Structure Induced by a Parameterization

A conformal structure on a surface is a choice of angles between tangent vectors. On an oriented surface, a conformal structure is equivalent to defining the operation, J, of rotating tangent vectors by ninety degrees counterclockwise in the tangent plane. This operation is also called a complex structure. For the general theory see [11].

A surface parameterization, $\mathbf{f} : M \to \boldsymbol{R}^3$, defines a complex structure J_f on the domain M. Indeed, let $\mathbf{v}$ be a vector tangent to M at some point $p \in M$, then $J_f(\mathbf{v})$ is the unique vector tangent to the domain satisfying

$$d\mathbf{f}\left(J_{\mathbf{f}}(\mathbf{v})\right) = \mathbf{N} \times d\mathbf{f}\left(\mathbf{v}\right).$$

Thus the defining relation for the complex structure J_f is

$$d\mathbf{f} \circ J_{\mathbf{f}} = \mathbf{N} \times d\mathbf{f}. \tag{2}$$

Suppose that M is an abstract oriented 2D manifold equipped with a complex structure J. A surface immersion $\mathbf{f} : M \to \boldsymbol{R}^3$ is called a conformal immersion if the induced complex structure coincides with the abstract complex structure J, $J_{\mathbf{f}} = J$.

2.4 The Differential Invariants: Mean Curvature, Gauss Curvature, Principal Axes and Principal Curvatures

Recall that the second fundamental form of $\mathbf{f}$ is a symmetric quadratic form defined by

$$\mathrm{I\!I}(\mathbf{u}, \mathbf{v}) = - < d\mathbf{N}(\mathbf{u}) | d\mathbf{f}(\mathbf{v}) >$$

where $< \cdot | \cdot >$ is the Euclidean scalar product in $\boldsymbol{R}^3$. At every point $p \in M$ there exists a positively oriented orthonormal frame $\{\mathbf{e_1}, \mathbf{e_2} = J_f(\mathbf{e_1})\}$, $\|d\mathbf{f}(\mathbf{e}_i)\| = 1$, of $T_p(M)$ in which the symmetric quadratic form $\mathrm{II}(\cdot, \cdot)$ is represented by a diagonal matrix

$$\begin{pmatrix} \mathrm{II}(\mathbf{e}_1, \mathbf{e}_1) & \mathrm{II}(\mathbf{e}_1, \mathbf{e}_2) \\ \mathrm{II}(\mathbf{e}_2, \mathbf{e}_1) & \mathrm{II}(\mathbf{e}_2, \mathbf{e}_2) \end{pmatrix}$$

where

$$\begin{aligned} \mathrm{II}(\mathbf{e_1}, \mathbf{e_2}) &= 0 \\ \mathrm{II}(\mathbf{e_2}, \mathbf{e_1}) &= 0 \\ \mathrm{II}(\mathbf{e_1}, \mathbf{e_1}) &= \kappa_1 \\ \mathrm{II}(\mathbf{e_2}, \mathbf{e_2}) &= \kappa_2. \end{aligned} \tag{3}$$

The vectors $\mathbf{e_1}$ and $\mathbf{e_2}$ are called principal curvature vectors, they define the principal axes, and the numbers κ_1, κ_2 are the principal curvatures. The mean curvature, H is the average of the principal curvature, and the Gauss curvature is the product of the principal curvatures.

3 Computing the Differential Invariants: Theory

We now present the theoretical results for computing from the Gauss map and the conformal structure, the mean curvature, the differential of the immersion, the Gauss curvature, the principal axes, and finally the immersion itself. All proofs are in the Appendix. These results underline our new computational strategy: clean up outliers in the Gauss map using Theorem 3.3, compute H directly form N, then proceede to compute the rest. (For motivation see the last paragraphs of Section 3.2.)

Theorem 3.1. *Let* $\mathbf{N}$ *be the Gauss map of a parameterized surface* $\mathbf{f} : M \to \boldsymbol{R}^3$ *and let* J_f *be the induced complex structure. If* $\mathbf{f}$ *is twice continuously differentiable then the differential* $d\mathbf{N}$ *of the Gauss map satisfies*

$$d\mathbf{N} = -H d\mathbf{f} + \omega, \tag{4}$$

where H *is the mean curvature, and* ω *is a* $\boldsymbol{R}^3$*-valued one form from the tangent plane to the Euclidean three space,*

$$\omega : T(M) \to \boldsymbol{R}^3, \tag{5}$$

such that, for every vector $\mathbf{v}$ *tangent to the domain* M *the image* $\omega(\mathbf{v})$ *satisfies*

$$\omega(\mathbf{v}) \perp \mathbf{N} \tag{6}$$

$$\omega\left(J_f(\mathbf{v})\right) = -\mathbf{N} \times \omega(\mathbf{v}). \tag{7}$$

Therefore we have the following corollaries expressing the differential invariants in terms of the Gauss map and the complex structure.

Corollary 3.1. *(Mean curvature) Let* $\mathbf{N}$ *be the Gauss map of a parameterized surface* $\mathbf{f} : M \to \boldsymbol{R}^3$ *and let* J_f *be the induced complex structure. If* $\mathbf{f}$ *is twice continuously differentiable, then*

$$-Hd\mathbf{f} = \frac{1}{2}\left(d\mathbf{N} - \mathbf{N} \times d\mathbf{N} \circ J_f\right) \tag{8}$$

We now come to the key observation that we can compute the mean curvature of a conformal immersion directly from the Gauss map.

3.1 Computing the Mean Curvature

Theorem 3.2. *Let* M *be an abstract 2D manifold equipped with a complex structure* J*, and* $\mathbf{N}$ *be the Gauss map of a conformal immersion of* M *in* R^3*. Let* τ *be a the one-from defined by*

$$\tau := \frac{1}{2}\left(d\mathbf{N} - \mathbf{N} \times d\mathbf{N} \circ J\right). \tag{9}$$

Then

1. *The mean curvature vanishes precisely when the form* τ *is trivial, that is,* $H(p) = 0$ *if and only if* $\tau_p = \mathbf{0}$*.*
2. *Away from its zero locus,* H *is a non-vanishing solution of the linear system of first order differential equations*

$$H\,d\tau = dH \wedge \tau, \tag{10}$$

where $\wedge$ *denotes the wedge product of 1-forms.*

In local cooridnates (x, y) we represent the form τ as $\tau = \tau_x dx + \tau_y dy$ and the system (10) as

$$\frac{\partial H}{\partial x}\tau_y - \frac{\partial H}{\partial y}\tau_x = H\left(\frac{\partial \tau_x}{\partial y} - \frac{\partial \tau_y}{\partial x}\right).$$

The system (10) is over-determined and admits non-vanishing solutions only if its coefficients satisfy competability conditions. These conditions amount to a test whether a vector field $\mathbf{N}$ is indeed the Gauss map of an immersion. If the conditions are satisfied then one can integrate for H starting from the value at an arbitrary chosen point. Note that the system of equations determines H up to a constant non-zero multiple, so one needs initial data to nail down an unique solution. This degree of freedom is a manifestation of the invariance of the Gauss map under global scaling.

The competability conditions have an important application in computer vision as a filter.

3.2 The Gauss Map Filter

Theorem 3.3. *Let* $\mathbf{N}$ *be Gauss map of a conformal immersion. Then at every point* p *in the domain of the immersion,* $\mathbf{N}$ *must satisfy either the equations*

$$\frac{1}{2}\left(d\mathbf{N}_p - \mathbf{N}(p) \times (d\mathbf{N}_p \circ J)\right) = \mathbf{0} \tag{11}$$

or the system of equations

$$\left\langle \frac{\partial^2 \mathbf{N}}{\partial x^2} + \frac{\partial^2 \mathbf{N}}{\partial y^2} \middle| \mathbf{N} \right\rangle = \left|\frac{\partial N}{\partial x}\right|^2 + \left|\frac{\partial N}{\partial y}\right|^2 \tag{12}$$

$$\left(\frac{\partial^2 \mathbf{N}}{\partial x^2} + \frac{\partial^2 \mathbf{N}}{\partial y^2}\right) \times \mathbf{N} = \left(\frac{\partial \tau_x}{\partial y} - \frac{\partial \tau_y}{\partial x}\right) \tag{13}$$

where (x, y) *is an arbitrary coordinate system defined in a neighborhood of* p *and satisfying* $\frac{\partial}{\partial y} = J\left(\frac{\partial}{\partial x}\right)$*, and* τ *is the 1-form defined in* (9).

The system (12), (13) expresses the competability conditions necessary for finding non-vanishing solutions H of (10).

In practice we use the equations in Theorem 3.3 to filter out noise in the Gauss map sampled by the sensors and the Gauss map extraction algorithms. A given sample is declared valid only if it satisfies at least one of the systems within given a threshold range.

Once the mean curvature is computed we are ready to compute the Gauss curvature the principal axes, and a parameterization (immersion). The usual approach is to recover the immersion and then compute the rest using the expression for the immersion. We adopt a slightly different tack. The motivation for this twist comes from two directions. First we want to avoid computations which may introduce round off errors and other numeric noise – if we stuck with the classical approach we would have to integrate and then differentiate numerically. Second, once we have $\mathbf{N}$ and H we can extract other ingredients needed for the computations directly from the sampled data. To clarify these points let us look at the theory we propose to use to compute the Gauss curvature and the principal axes.

3.3 The Gauss Curvature and the Principal Axes

For the rest of the paper let $\mathbf{N}$ be Gauss map of a twice differentiable conformal immersion $\mathbf{f}$. The complex structure J on the domain M of $\mathbf{f}$ coincides with the induced structure $J_{\mathbf{f}}$ and so for the rest of the paper we use only the notation J for the complex structure. Let ω be the $\boldsymbol{R}^3$-valued 1-form introduced in Theorem 3.1. Therefore $\omega = dN + Hdf$. We show how to express ω purely in terms of $d\mathbf{N}$ and the complex structure and relate it to the principal curvature axes and the Gauss curvature.

Corollary 3.2. *(The form ω and principal axes) Let* $\mathbf{N}$ *be the Gauss map of a parameterized surface* $\mathbf{f} : M \to \boldsymbol{R}^3$*, and let J be the complex structure. If* $\mathbf{f}$ *is twice continuously differentiable, then*

$$\omega = \frac{1}{2} \left(d\mathbf{N} + \mathbf{N} \times d\mathbf{N} \circ J \right). \tag{14}$$

Furthermore, $\omega(\mathbf{u})$ is collinear to $d\mathbf{f}(\mathbf{u})$ if and only if the vector $\mathbf{u}$ is collinear to a principal curvature vector. Thus the quadratic form $< \omega(\cdot)|d\mathbf{f}(\cdot) >$ is symmetric and trace-free (i.e., has zero trace), and its eigenvalues are precisely $\pm\frac{1}{2}(\kappa_1 - \kappa_2)$, where κ_1 and κ_2 are the principal curvatures.

We can estimate the principal curvature vectors by solving

$$\frac{1}{2} \left(d\mathbf{N}(\mathbf{u}) + \mathbf{N} \times d\mathbf{N}(J(\mathbf{u})) \right) = \lambda \, d\mathbf{f}(\mathbf{u}) \tag{15}$$

for the scalar λ and the vector $\mathbf{u}$. This amounts to diagonalizing a symmetric trace free matrix representing the quadratic form $< \omega(\cdot)|d\mathbf{f}(\cdot) >$. The diagonalization of such matrices is more stable than the diagonalization of general matrices.

Corollary 3.3. *(Gauss curvature:) Let* $\mathbf{N}$ *be the Gauss map of a parameterized surface* $\mathbf{f} : M \to \boldsymbol{R}^3$ *and let J_f be the induced complex structure. Let H be the mean curvature. Let* $\mathbf{f}$ *be twice continuously differentiable, ω be the one form defined in (5), and λ^2 be the sum of the squares of the eigenvalues of the quadratic form, $< \omega(\cdot)|d\mathbf{f}(\cdot) >$. Then, the Gauss curvature, K, satisfies*

$$K = H^2 - \lambda^2. \tag{16}$$

Equation (16) gives a stable method for computing the Gauss curvature K. We do not need to diagonalize the quadratic form matrix. To compute λ^2, we can chose any orthonormal basis of the tangent plane to the surface in $\mathbf{R^3}$, then we represent the quadratic form $< \omega(\cdot)|d\mathbf{f}(\cdot) >$ as matrix A, and set λ^2 as follows

$$A = \begin{pmatrix} a & b \\ b & -a \end{pmatrix}, \; \lambda^2 = a^2 + b^2. \tag{17}$$

Remark 3.1. The upshot of all this is that to compute curvatures and principal axes we need the Gauss map, the complex structure, and estimates for the differential $d\mathbf{f}$. Furthermore, the computations are more stable then the ones based on the usual geometric formulae.

Finally, we turn to estimating the differential $d\mathbf{f}$, and the immersion itself. The idea is that in practice we can either solve for them or use estimates provided from the data, or use a combination of both methods.

3.4 Recovering $d\mathbf{f}$ and the Conformal Immersion

We begin by noticing that it is very easy to obtain $d\mathbf{f}$ in the closure of the region $M' = \{p \mid H(p) \neq 0\}$ where the mean curvature is not zero, $H \neq 0$, that is, where the form τ defined in (9) is nondegenerate. Indeed, from (8) we get

$$d\mathbf{f} = \frac{1}{H}\tau, \quad \text{if } H \neq 0.$$

By continuity we get $d\mathbf{f}$ in the closed domain $\overline{M'}$. This leaves us with the task of determining the differential $d\mathbf{f}$ on open sets on which the mean curvature vanishes identically, if such exist. To do this precisely we solve a Dirichlet problem for a linear elliptic differential equation. The boundary data is provided from the boundary values of $d\mathbf{f}$ along the boundary $\partial M'$ and from assumptions about the surface edge properties if the surface has boundary components where H vanishes identically, or if the surface has an end within the region where H is identically zero. The later case can be safely disregarded in computer vision applications because of the natural clipping that takes place as we sample surfaces with sensors. The exact mathematical method will be presented in a forthcoming paper. It is based on the classical Weierstrass representation techniques for minimal surfaces and the new techniques introduced in a recent preprint [9].

In principle once we obtain the differential $d\mathbf{f}$ then a parameterization is obtained by integrating the differential. In practice of course, we have to use quadratures to evaluate the integrals from the discrete data but that amounts to using the differential itself to determine explicitly the displacements between the discrete points on the surface recovered via the immersion.

In applications where the initial data set is a cloud of 3D points sampled from the surface of the object we have been estimating $d\mathbf{f}$ in the region where τ is degenerate, equivalently, where $H \equiv 0$, directly from the local displacements between sample points weighted by local scale units in available neighborhood directions, [10]. See Fig. 1.

4 Examples and Future Work

The proposed method has been tested on various types of data: from MRI images, stereo images, range images, and computer generated surfaces. We show examples of each of these categories here. In all cases, a fish-scales procedure, [14], extracts the Gauss map and neighborhood stratification from the 3D sampled cloud of surface points.

MRI: The data sets are from the data base of Gill Barequet, Dept of CS, Tel Aviv University and the surface points are extracted manually by Bernhard Geiger, INRIA, Sophia Anapolis. The 3D surface points of human hip joint and cartilage are extracted from MRI images. We show the normals at the sampled points that are input for out method, and the results that the method has produced, Fig. 2.

Stereo data: The 3D clouds of points sampled from human faces are obtained using stereo [1, 14]. See Fig. 4 and Fig. 5.

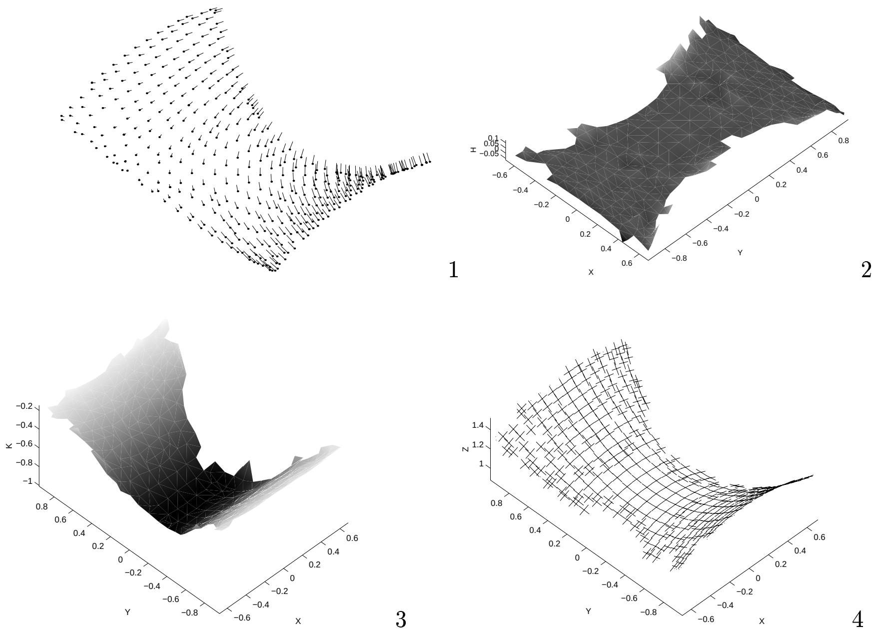

Fig. 1. Catenoid: (1)Gauss map; (2) and (3) the recovered mean curvature surface, (x, y, H), and the Gauss curvature surface, (x, y, K); (4) the principal axes superimposed on the catenoid surface.

Computer generated: We have tested the approach on various computer generated surfaces and reported the results elsewhere. Here we just show results for the catenoid, Fig. 1. See [10] for details.

Range data: The 3D cloud of points is sampled from the surface of the head of a mannequin. The 3D data are collected by a Cyberware scanner. The range data set is generated at the GRASP Laboratory, University of Pennsylvania. For our results see Fig. 3

Thus we demonstrated, that we can handle various types of input data, from sparse (MRI example) to dense (Range data example), and from clean (Computer generated example) to noisy (Stereo example). Our next goal is to do rigorous performance evaluation of the method, including propagation of errors, empirical evaluation based on ground truth, and empirical comparisons with other existing methods.

Appendix: Proofs

Recall the definitions of the differential invariants from Section 2. Note that the equations (3) are equivalent to

$$d\mathbf{f}\left(\mathbf{e_1}\right) \perp d\mathbf{f}\left(\mathbf{e_2}\right) \tag{18}$$

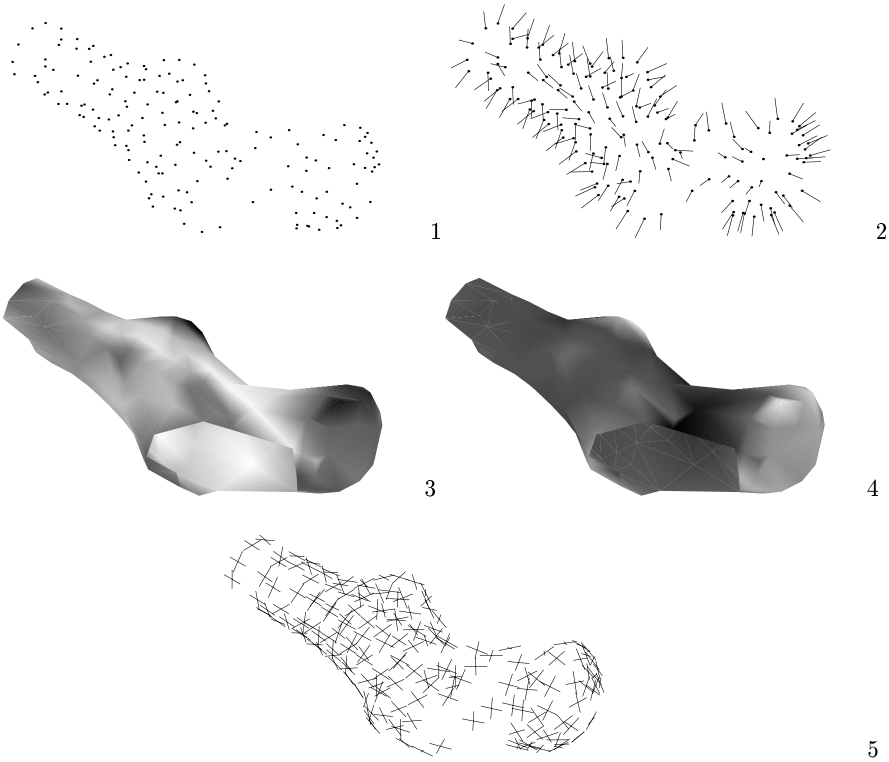

Fig. 2. MRI data, human hip joint: (1) the 3D cloud of points extracted from the MRI data; (2) the Gauss map; (3) the restored surface shaded by the recovered mean curvature values; (4) the restored surface shaded by the recovered Gauss curvature; (5) the principal directions at at the 3D surface points. The mean curvature values range from -3.6622 to -0.0441 with a mean value of -1.4332. The Gauss curvature values range from -6.4579 to 10.8133 with a mean value of 0.4488. Lighter shades in (4) represent higher, positive Gauss curvatures.

$$d\mathbf{N}\left(\mathbf{e_1}\right) = -\kappa_1\, d\mathbf{f}\left(\mathbf{e_1}\right) \tag{19}$$

$$d\mathbf{N}\left(\mathbf{e_2}\right) = -\kappa_2\, d\mathbf{f}\left(\mathbf{e_2}\right) \tag{20}$$

Proof of Theorem 3.1 The form of equations (19) and (20) suggests that the one form $d\mathbf{N}$ can be represented as

$$d\mathbf{N} = A\, d\mathbf{f} + \omega \tag{21}$$

for some coefficient A and some $\boldsymbol{R}^3$-valued one-from ω. We decide to look for a form ω satisfying the condition

$$\omega(J_f(\mathbf{u})) = -\mathbf{N} \times \omega(\mathbf{u}). \tag{22}$$

This choice for ω can be motivated by the decomposition of symmetric tensors into diagonal and trace-free components. A direct way to motivate our choice is

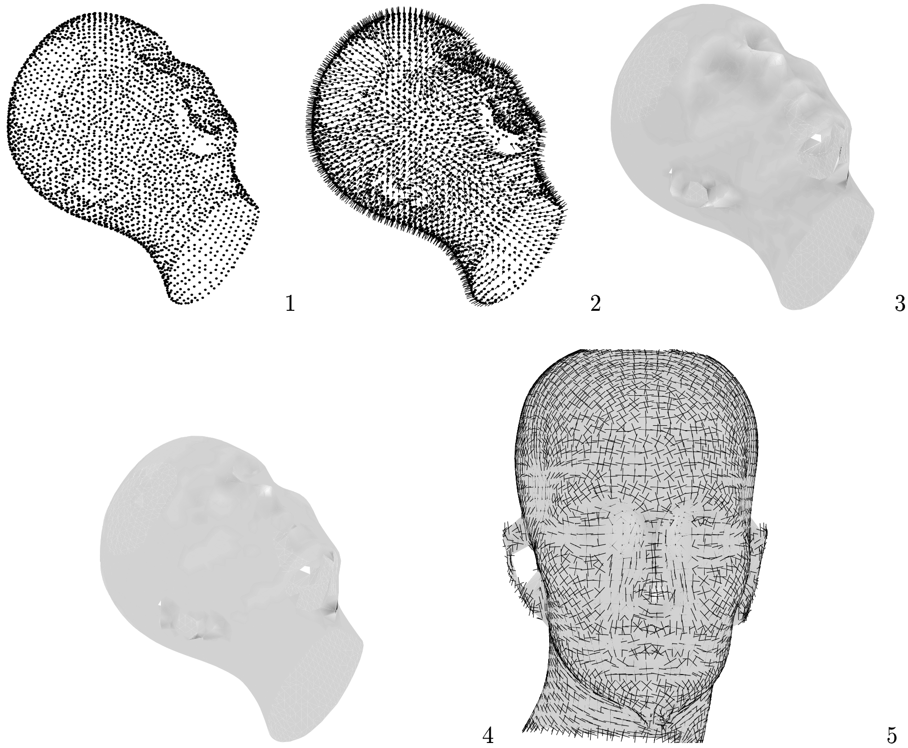

Fig. 3. Range data, mannequin head: (1) the 3D cloud of points extracted by Cyberware; (2) the Gauss map; (3) the restored surface shaded by the recovered mean curvature values; (4) the restored surface shaded by the recovered Gauss curvature; (5) the principal directions at at the 3D surface points. Lighter shades in (4) represent higher, positive Gauss curvatures. One can clearly identify the curvature lines, despite the bad Postscript conversion: in the electronic version of this image there were no holes in the curvature lines.

to notice that the form $d\mathbf{f}$ satisfies

$$d\mathbf{f}\,(J_f(\mathbf{u})) = \mathbf{N} \times d\mathbf{f}\,(\mathbf{u}). \tag{23}$$

That is, $d\mathbf{f}$ relates a counter-clockwise rotation by ninety degrees in $T_p(M)$ to a counterclockwise rotation by ninety degrees around the axes $\mathbf{N}$ in $\boldsymbol{R}^3$. On the other hand, the condition (22) guarantees that the form ω relates a counter-clockwise rotation by ninety degrees in $T_p(M)$ to a clockwise rotation by ninety degrees around the axes $\mathbf{N}$ in $\boldsymbol{R}^3$. The representation (21) accounts for the possibility that the one-form $d\mathbf{N}$ may be a combination of forms which rotate in different directions around the $\mathbf{N}$ axes. From (21) and (19) and (20) we obtain

$$\omega\,(\mathbf{e_i}) = (\kappa_i + A)\; d\mathbf{f}\,(\mathbf{e_i})\,, \quad i = 1, 2.$$

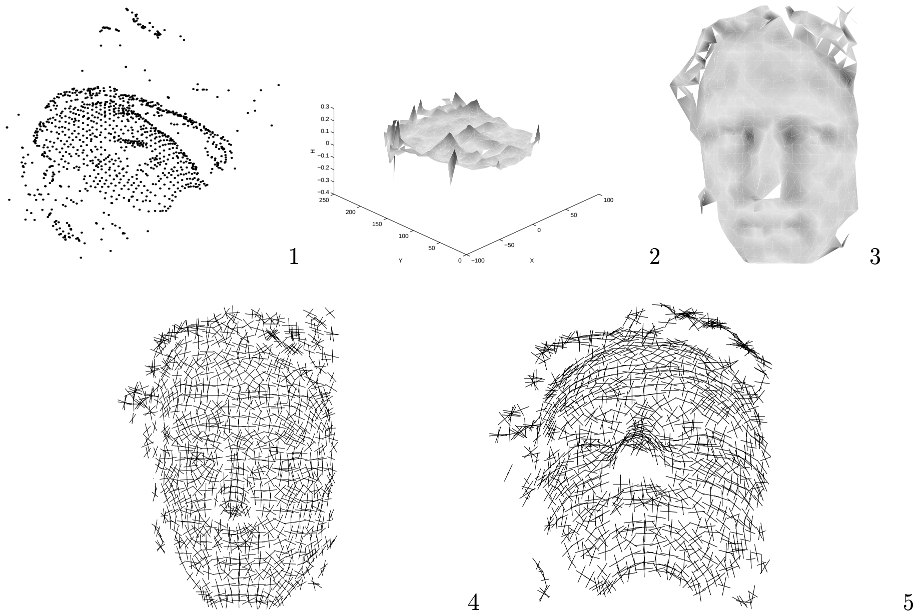

Fig. 4. Stereo data, "face". (1) The reconstructed 3D points, from stereo. (2) The surface graph (x, y, H) of recovered mean curvature. (3) The recovered mean curvature surface (x, y, H) frontal view. Note that we use as height the curvature values. (4) and (5) The principal axes attached at the 3D points. We can identify well-formed curvature lines in nonplanar regions (on the chin, the eyebrow areas, the cheeks).

These identities show that (6) holds. Furthermore combining the identities with (22) and $\mathbf{e_2} = J_f(\mathbf{e_1})$ we obtain

$$\begin{aligned}
\omega(\mathbf{e_2}) &= (\kappa_2 + A)\; d\mathbf{f}(\mathbf{e_2}) \\
-\mathbf{N} \times \omega(\mathbf{e_1}) &= \mathbf{N} \times (\kappa_2 + A)\; d\mathbf{f}(\mathbf{e_1}) \\
-\mathbf{N} \times (\kappa_1 + A)\; d\mathbf{f}(\mathbf{e_1}) &= \mathbf{N} \times (\kappa_2 + A)\; d\mathbf{f}(\mathbf{e_1})
\end{aligned}$$

The last identity implies that the tangential vector $(\kappa_1 + \kappa_2 + 2A)\; d\mathbf{f}(\mathbf{e_1})$ is colinear to the normal $\mathbf{N}$. This can only happen if it is the zero vector in $\boldsymbol{R}^3$, that is, $A = -\frac{1}{2}(\kappa_1 + \kappa_2) = -H$.

Proof of Corollaries 3.1,3.2,3.3.

From (4), (2), and (22) we get

$$\begin{aligned}
\mathbf{N} \times d\mathbf{N} \circ J_f &= -H\mathbf{N} \times (\mathbf{N} \times d\mathbf{f}) - \mathbf{N} \times (\mathbf{N} \times \omega) \\
&= H\, d\mathbf{f} + \omega.
\end{aligned}$$

The identities (8) and (14) follow directly from (4) and the identity $\mathbf{N} \times d\mathbf{N} \circ J_f = H\, d\mathbf{f} + \omega$. Rewriting (4) in the form $\omega = d\mathbf{N} + H\, d\mathbf{f}$ we conclude that $\omega(\mathbf{u})$ is colinear to $d\mathbf{f}(\mathbf{u})$ if and only if the later is colinear to $d\mathbf{N}(\mathbf{u})$, that is, if and only

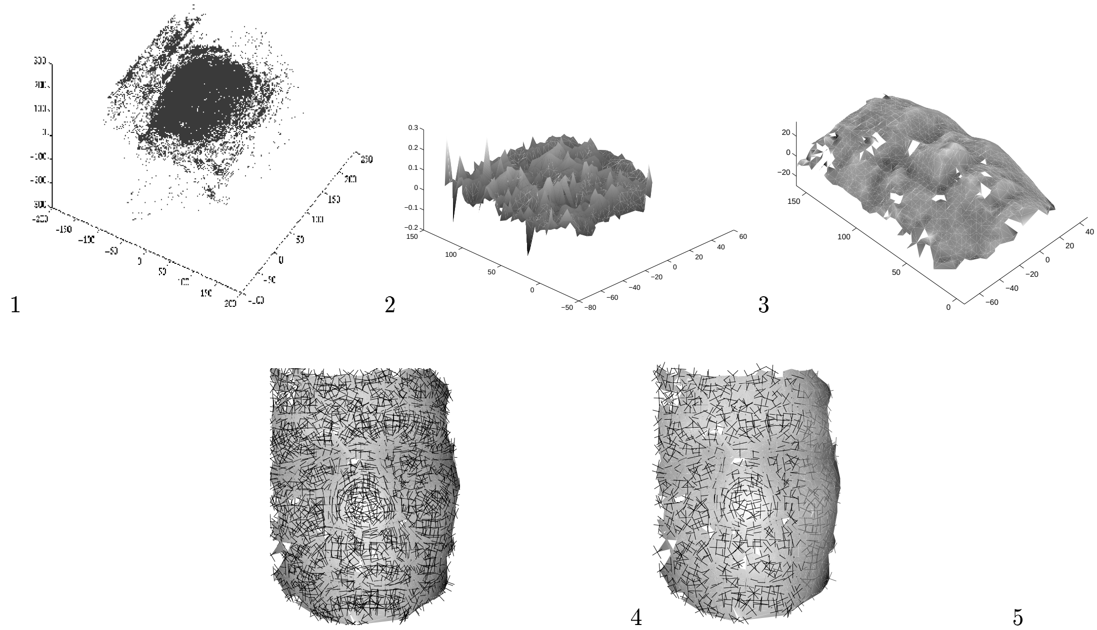

Fig. 5. Stereo data, "face 2". (1) The reconstructed 3D points, from stereo. (2) The surface graph (x, y, H). (3) The recovered surface shaded by mean curvature. (We used a simple polygonization to render the surface.) (4) and (5) The principal axes attached at the 3D points. We can identify well-formed curvature lines in non-planar regions (on the chin, the eyebrow areas, the cheeks). In (4) we show the principle frames at all sample points. In (5) we show only half the points. Note that the conversion to Postscript lead to image quality loss.

if $\mathbf{u}$ is parallel to a principal curvature vector. Furthermore, from equations (19) and (20) we obtain

$$\omega\left(\mathbf{e_1}\right) = \frac{k_2 - k_1}{2}\, d\mathbf{f}\left(\mathbf{e_1}\right), \quad \omega\left(\mathbf{e_2}\right) = \frac{k_1 - k_2}{2}\, d\mathbf{f}\left(\mathbf{e_2}\right).$$

Proof of Theorem 3.2 Conclusion 1 follows directly from (8), that is, from

$$-Hdf = \tau. \tag{24}$$

To obtain Conclusion 2 differentiate (24) and multiply both sides by H.

Proof of Theorem 3.3 In the regions where $H = 0$, that is, τ is degenerate we must have $\tau \equiv \mathbf{0}$ which is the same as (11). The rest of the proof follows from splitting the system $Hd\tau = dH \wedge \tau$ into pieces tangential and perpendicular to $\mathbf{N}$ and restating them them in terms of the derivatives of the Gauss map.

References

[1] Bajcsy, R., Enciso, R., Kamberova, G., Nocera, L., Šara, R.: 3D Reconstruction of Environments for Virtual Collaboration, *Proc. 4th IEEE Workshop on Applications of Computer Vision*, Princeton, NJ,(1998).

[2] Boissonnat,J.D., Cazals, F.: Smooth surface reconstruction via natural neighbour interpolation of distance functions. In *ACM Proc XVIth Annl Symp. in Comp. Geom.* Hong Kong (2000).

[3] Besl, P., Jain, R.: Invariant surface characteristics and 3d object recognition in range images. *CVGIP*, 33:33–80, (1986).

[4] Quicken, M., Brechbüller, C. , et al: Parameterization of Closed Surfaces for Parametric Surface Description. Proceedings IEEE CVPIR 2000, volume 1, 354–360.

[5] DoCarmo, M.: *Differential Geometry of Curves and Surfaces.* Prentice-Hall, (1976).

[6] Devernay, F.: "Computing Differential Properties of 3-D Shapes from Stereoscopic Images without 3-D Models", INRIA, RR-2304, Sophia Antipolis, (1994).

[7] Feldmar, J., Ayache, N.: *Registration of Smooth Surfaces Using Differential Properties.* Number 801. Springer-Verlag, (1994).

[8] Ikeuchi, K., Herbert, M.: Spherical Representations: from EGI to SAI CMU Technical report: CMU-CS-95-197 (1995).

[9] Kamberov, G., Norman, P., Pinkall, U., Pedit, F.: Quaternionions, Spinors, and Surfaces to appear in Contemporary Mathematics, AMS

[10] Kamberov, G., Kamberova, G.: Shape Invariants and Principal Directions from 3D Points and Normals, Proceedings of 10th Intl. Conf. in Central Europe on Computer Graphics and Visualization, (2002), Plzen. Journal of WSCG, Volume 10, (2002), Pages 537-544.

[11] Kobayashi, S., Nomidzu, K.: Foundations of Differential Geometry. Volume 2. Chapter IX. Interscience Publ. New York. (1969).

[12] Koenderink, J.: *Solid Shape* MIT press, (1990).

[13] Metaxas, D., Terzpoulos, D.: Dynamic 3D models with local and global deformations: deformable superquadrics, IEEE PAMI, 13(7):703-714, (1991).

[14] Šára, R., Bajcsy, R.: "Fish-Scales: Representing Fuzzy Manifolds," *Proc. Int. Conference on Computer Vision*, Bombay, India, Narosa Publishing House, (1998).

[15] Shum, H., Herbert, M., Ikeuci, K.: On 3d shape synthesis. In *Proc. Image Under. Workshop*, volume 2, pages 1103–1112, (1996).

[16] Sommerfeld, A.: *Mechanics of Defeormable Bodies.* Lectures in Theoretical Physics volume 2, pp 122-124, Academic Press, New York, (1950).

[17] Taubin, G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proc. ICCV.* IEEE Comp. Soc. Press, (1995).

[18] Worthington, P., Hancock, E.: Histogram-based Object Recognition Using Shape from Shading. *Proc. IEEE Conf. CVPR 2000*, volume 1, 643–648 (2000).

[19] Yuen, P., et al: Curvature and Torsion Feature Extraction from 3-D Meshes and Multiple Scales. *IEEE Proc. Vis. Image Signal Process*, volume 147, No 5, 454–462 (2000).