Curvature from Sliding.

Ganovelli F., Cignoni P., Scopigno R. †

Visual Computing Laboratory, ISTI-CNR

Abstract

We introduce a novel interpretation of the curvature over discrete surfaces that allows to compute a multi scale measure of principal directions and values of curvature and which is robust with regards to tessellation degeneracies and geometric and topological noise. The key observation is that there is a direct relation between the way a portion of surface can rotationally slide over a surface and the directions of principal curvature. We exploit such relation to setup a optimization problem that we use to actually compute the curvature values in a novel way.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

The knowledge of principal curvatures and principal direction plays a fundamental role in several applicative domains. In mesh fairing they are used to drive smoothing, simplification, edge flip optimization. They are used in feature detection and reverse engineering for their direct relation to the structure of a surface; in mesh processing they have been used to drive the process of automatic alignment of range maps [GMGP05].

Unfortunately, while the curvature is well defined for continuous surfaces, the same cannot be told for tessellated surfaces because its definition includes second order derivatives which are lost when approximating a smooth surface with a tessellation. The intended meaning of the curvature of tessellated surfaces is that it is the curvature of the continuous surface the tessellation represents. The problem is the same as the computation of the normal, just more difficult because of the extra differentiation needed to define the curvature.

The existing literature shows a large number of algorithms for computing the curvature values and/or principal directions. We can roughly classify them in three categories:
the **discrete** operators that estimate differential quantities directly from the one ring neighborhood of the vertex. In these methods the umbrella operator is ubiquitously used to provide estimation of the curvature based on mesh connectivity, specifically on the one-ring neighborhood of the vertex [Tau95, MDSB02, CSM03]. Discrete operators relies on the connectivity information and so are sensitive to topological noise;
the algorithms based on **fitting** that try to fit a parametric patch in the neighborhood of the vertex and hence derive the values [dC76, CP03];
the approaches based on **integral invariants** which exploit the relation from the euclidean neighborhood of the vertex and its differential characteristics [PWY*07, YLHP06]. Approaches based on integral invariants are more robust in that they do not use directly the connectivity information of the tessellation but only the locus of 3D points occupied by the surface, and more precisely the intersection between a ball centered at the vertex and the *local* volume. Intuitively the local volume is the region of space underneath the surface and it is well defined whenever the surface separates the ball in two spaces (outside and inside).

However, there are cases in which the local volume is not clearly defined because the surface may have small holes or duplicated parts or even single points (see Figure 1). These situations are unlikely in a modelled object but are the most common situation in 3D scanning, when the range maps have been acquired and aligned. Obviously we can apply mesh repairing algorithms to fix these situations, as for example it has been done in [GMGP05] where volumetric hole filling is used to close small holes.

In this paper we propose a novel method to compute scale dependent curvature values and principal directions that we call *curvature from sliding*. The key intuition is that if we

† name.surname@isti.cnr.it

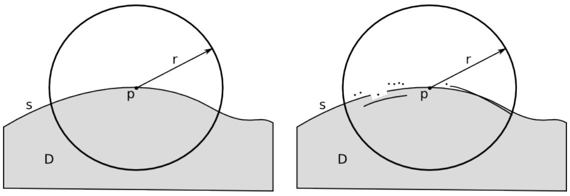

Figure 1: *Left: integral invariants are defined over the intersection of a ball with the local volume. Right: a situation where such intersection is not well defined.*

take a portion of surface around a vertex and make it *slide* by a small amount over the surface, the directions along which it slides "more easily" are the principal curvatures. We formalize this practical notion by calling "sliding" a small rotation of a surface patch around a candidate *obsculating circle* and by estimating how easily it slides by integrating the distance between the rotated and original surface patch. Therefore computing the curvature principal directions is a matter of finding the obsculating circle that gives the best slide, which turns out to be a simple non linear optimization problem of 2 variables. The size of the portion of surface we make slide gives a scale measure of the curvature.

The rest of the paper is organized as follows: Section 1 introduces the *curvature from sliding* method and shows why it is sound; results are presented and discussed in Section2 while future directions of work are sketched in Section 3.

1. Curvature from sliding

In this section we formalize the intuitive concept of *sliding* by considering the rotation of small pieces of surface along possible osculating circles.
Given a surface S and a point $p \in S$, let $B_r(p) = \{p' \in S|||p'-p|| < r\}$ be the intersection of a ball of radius r centered at p with the original surface S, and let assume the existence of a tangential frame with axes tX, tY, tZ where $tY = n(p)$.

Consider the plane π_ω, passing through $n(p)$ and forming an angle ω with the tX axis: we define the rotation $R(\delta, \omega)$ of α radiants around the axis perpendicular to π_ω at a distance $1/\delta$ from p along $n(p)$. This rotation $R(\delta, \omega)$ is simply the rotation corresponding to the osculating circle lying on π_ω around the axis orthogonal to the curvature direction.

If we apply the rotation $R(\delta, \omega)$ to a small piece of surface $B_r(p)$ we obtain a displaced piece of surface B' that will still match with the original surface only if the surface has a curvature corresponding to the osculating circle associated with $R(\delta, \omega)$.

To measure this matching we define a function $f(\delta, \omega) = Dist(B', S)$ that measures the distance between the rotated piece $B' = R(\delta, \omega) \cdot B_r(p)$ and the original surface S. The function $Dist$ can be defined in several ways, in this discussion we will use:

$$Dist(B', S) = \int_{B'} ||closest(p) - p|| dp$$

that can be efficiently approximated by sampling the displaced surface B' and, for each sample, finding the closest point in on the original surface S.

So far, we have defined the set of rotations corresponding to all the possible osculating circles for the point p, that we will call *sliding space* from now on, and a function f which tell us, for each rotation, how far it would be the portion of surface contained in $B_r(p)$ if rotated. The key observation is that the function $f(\delta, \omega)$ has two minima in correspondence of the two principal directions of curvature. As a very simple example, consider a cylinder with unitary radius and any point on its surface. In this case it is easy to see that the function f has two minima in $(\delta = 1.0, \omega = 0.0)$, i.e. you can make a portion of surface rotate about the axis of the cylinder, and in $(\delta = 0.0, \omega = \pi/2)$, i.e. you can make a portion of surface translate in the same direction as the axis. Figure 2 shows to views of the plot of f for the cylinder that shows the shape of f and its minima.

This is far to be a mathematical proof but serves as a hint that these considerations can be used to compute the curvature values and principal directions simply minimizing the value of f in the sliding space.

1.1. Finding the curvature

Following the above interpretation, finding the curvature at a point p is essentially a matter of minimizing $f(\delta, \omega)$, i.e. finding the rotations for which the surface smoothly slide. We recall that $f(\delta, \omega)$ measures how much the rotated portion of surface contained in the ball or radius r is away from the original surface. In general it cannot be guaranteed that f will have only the two minima corresponding to the principal curvatures, but, to choose the best candidates, we exploit the propriety of orthogonality of the principal directions to pick up two minima (δ_0, ω_0) and (δ_1, ω_1) such that $(\omega_1 - \omega_0) \equiv \frac{\pi}{2}$.

We remember that, similarly to curvature definition given in [YLHP06], the ball radius that we use for evaluating the f function, gives the scale at which the curvature is computed.

2. Results and discussion

The method presented in this papers was implemented in C++ supported with the library CMinPack [Dev03] for the optimization. Figure 3 shows the maximum curvature values mapped as vertex color and, in the botom row, the direction of minimum curvature for a portion of the model. In these experiments the ball radius is constant for all the points and chosen by the user, while the amount of rotation (referred as ω in Section 1) is chosen as $1/10$ of the ball radius.

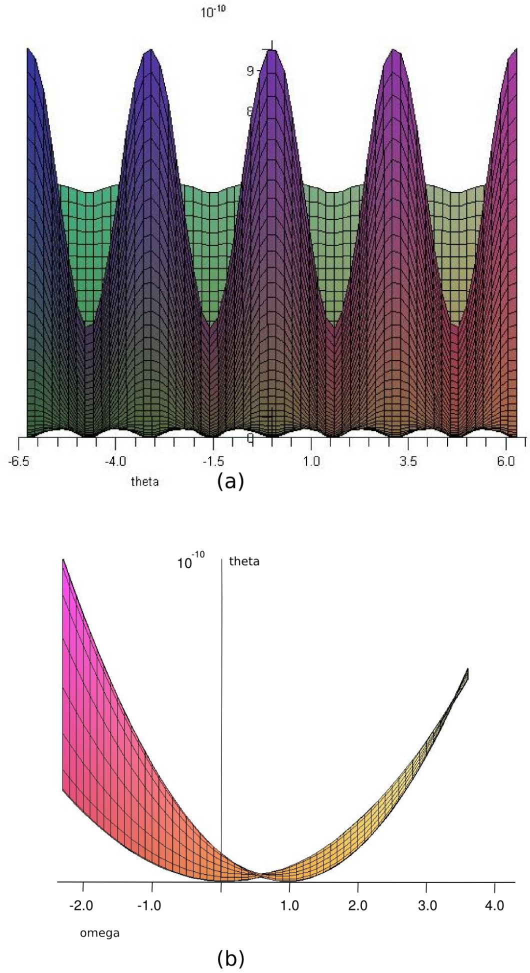

Figure 2: *Plot of the function f for the cylinder.*

Figure 4 shows the principal directions of curvature for a simple head model where half of the faces of one side were removed from the model. This is done in order to show that, contrary to PCA analysis, there is no need of a local volume. Note that the directions are little affected by this severe change of tessellation.

In figure 5 two aligned range scans are used. Note that the tessellation is regular and not the smooth and adaptive recostruction that we obtain at the end of the scanning pipeline and that most of the dataset consists of two overlapping surfaces. To compute the curvature with this data using volume integral approaches you would need first to merge the scans in a single surfaces and then fix the mesh to eliminate small connected components (up to single unconnected vertices) and close holes, while our algorithm uses the data as they are without preprocessing.

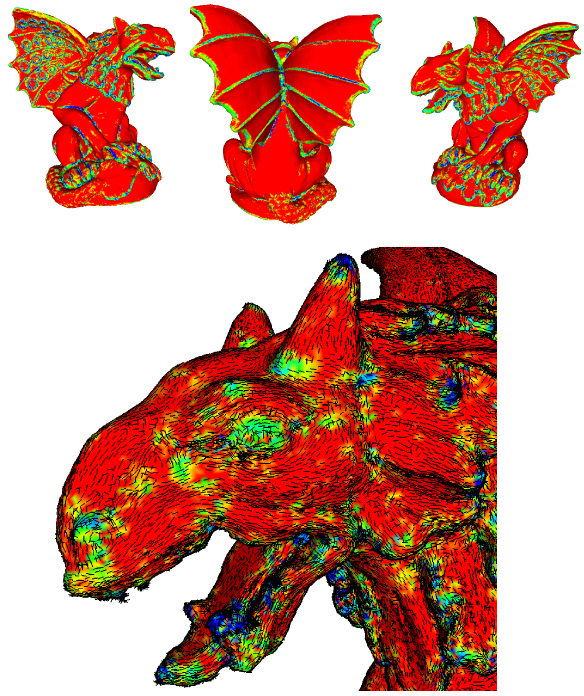

Figure 3: *Vertices of the gargoyle are colored according to the maximal curvature with a color ramp from red to blue. Black lines show the minimal curvature.*

3. Conclusions and Future Work

In this paper we have proposed a new approach for robustly computing curvatures based on the connection between sliding along a curve and principal curvature directions. The proposed approach works in a multi-scale approach in situations where previous approaches could fail. Its main strength is that it does not need to interpret the dataset as a surface, i.e. to close holes, to merge overlapping surfaces, but it gives a measure of curvature from raw data. The next step will be to provide an adaptive version of the algorithm which change the radius of the ball according to the local shape. This can be done by analyzing the final value of the optimization, and testing it aginst the expected value that can be calculated in the assumption the curvature is unchanged over all the point in the ball. If the final value is significantly higher than the expected value it means the ball must be shrink, while it can be enlarged if there is little difference.

Although a straightforward implementation of the proposed approach is not significantly competitive from a performance point of view (by using a generic Non Linear Minimization algorithm you can compute the curvature for roughly 200 vertices per second, whic means you could require about 2 mins for computing curvatures on a 50k tri mesh), we think that the described *sliding* interpretation of

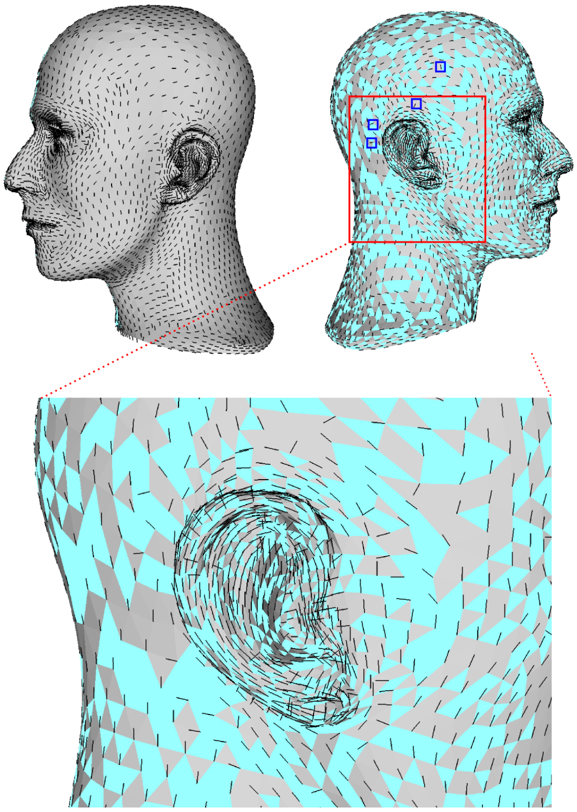

Figure 4: *A simple model of a head with adaptive triangulation. The light green triangles those removed prior to compute the curvature directions. Note that even with this serious perturbation the values of the direction remain fairly coherent with the original model (a few of the exceptions are marked with a blue rectangle).*

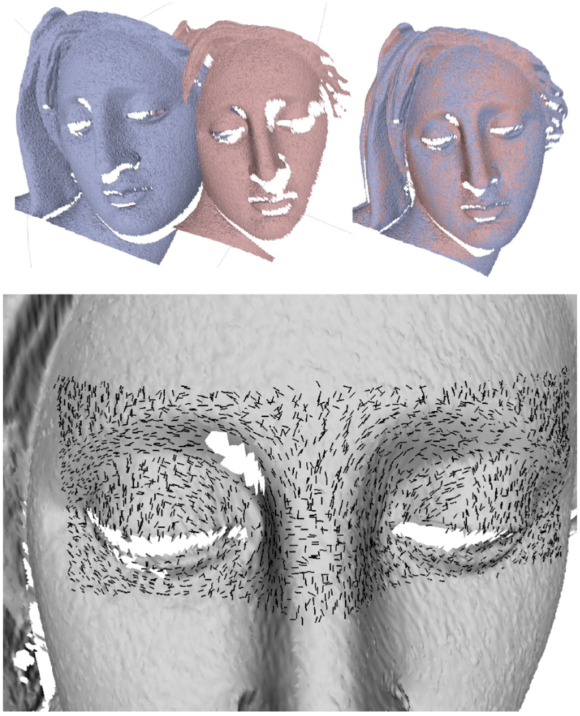

Figure 5: *In the upper row the two meshes are shown separately and aligned. In the aligned configuration the principal direction of curvature are computed.*

curvature of surfaces can shed an interesting light on this problem that could lead to further results.

References

[CP03] CAZALS F., POUGET M.: Estimating differential quantities using polynomial fitting of osculating jets. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 177–187.

[CSM03] COHEN-STEINER D., MORVAN J.-M.: Restricted delaunay triangulations and normal cycle. In *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry* (New York, NY, USA, 2003), ACM Press, pp. 312–321.

[dC76] DO CARMO M. P.: *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[Dev03] DEVERNAY F.: Cminpack. More info on: `http://devernay.free.fr/hacks/cminpack.html`, 2003.

[GMGP05] GELFAND N., MITRA N. J., GUIBAS L. J., POTTMANN H.: Robust global registration. In *In M. Desbrun and H. Pottmann, editors, Eurographics Association, ISBN 3-905673-24-X.* (2005), pp. 197–206.

[MDSB02] MEYER M., DESBRUN M., SCHRËĘDER P., BARR A.: Discrete differential geometry operators for triangulated 2-manifolds. In *InternatioPotWalYan.etal07nal Workshop on Visualization and Mathematics* (2002).

[PWY*07] POTTMANN H., WALLNER J., YANG Y.-L., LAI Y.-K., HU S.-M.: Principal curvatures from the integral invariant viewpoint. *Computer Aided Geometric Design 24*, 8-9 (2007), 428–442.

[Tau95] TAUBIN G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In *ICCV* (1995), pp. 902–907.

[YLHP06] YANG Y.-L., LAI Y.-K., HU S.-M., POTTMANN H.: Robust principal curvatures on multiple scales. In *Eurographics Symposium on Geometry Processing* (Cagliari, Sardinia, Italy, 2006), Sheffer A., Polthier K., (Eds.), Eurographics Association, pp. 223–226.