Normal, dihedral angle, curvature

jjcao

The purpose of computing is insight, not numbers

Normal

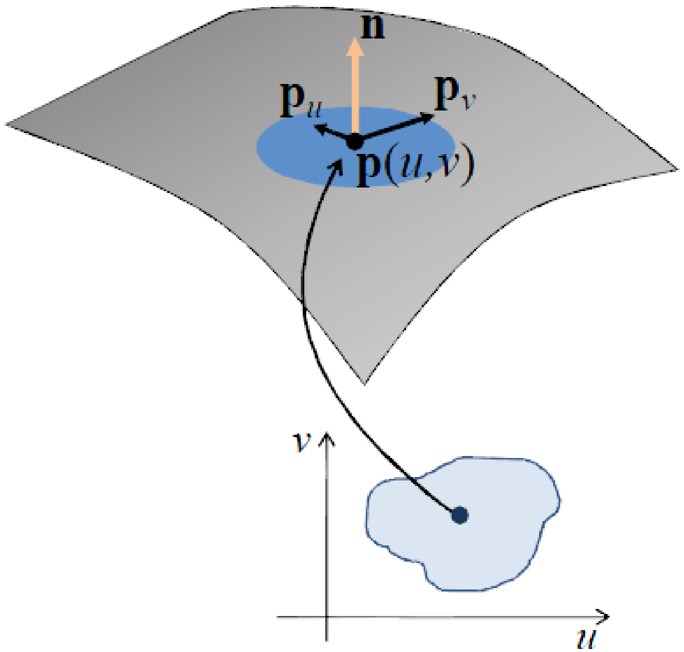
Surfaces

- Surface normal:

$$\mathbf{n}(u,v) = \frac{\mathbf{p}_u \times \mathbf{p}_v}{\|\mathbf{p}_u \times \mathbf{p}_v\|}$$

- Assuming regular parameterization, i.e.,

$$\mathbf{p}_u \times \mathbf{p}_v \neq 0$$



$$\mathbf{n}(T) = \frac{(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)}{\|(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)\|}$$

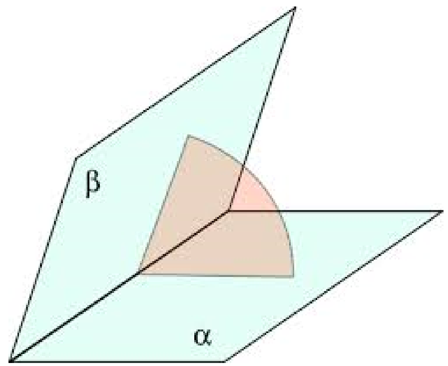
$$\mathbf{n}(v) = \frac{\sum_{T \in \mathcal{N}_1(v)} \alpha_T \, \mathbf{n}(T)}{\left\| \sum_{T \in \mathcal{N}_1(v)} \alpha_T \, \mathbf{n}(T) \right\|}.$$

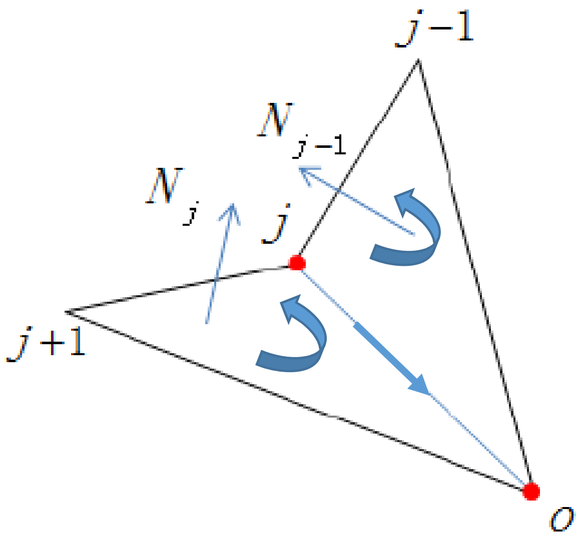
constant weights and area weights yield the result in the center; angle weights, the result on the right.

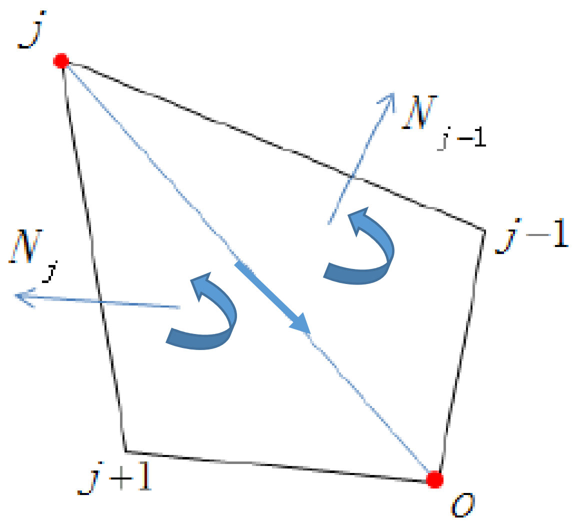
[Max 99] Nelson Max. Weights for Computing Vertex Normals from Facet Normals. Journal of Graphics, GPU, and Game Tools 4:2 (1999)
[Jin et al. 05] Shuangshuang Jin, Robert R. Lewis, and DavidWest. A Comparison of Algorithms for Vertex-Normal Computation. The Visual Computer 21:1{2 (2005)

Dihedral-angle

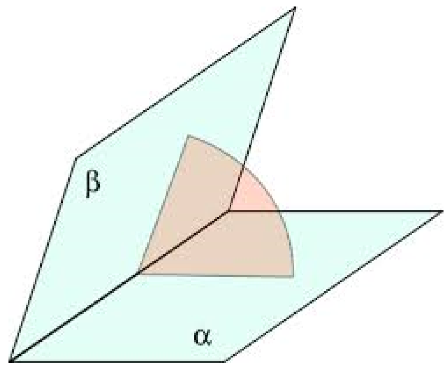
- $\varphi_{AB} = \mathrm{acos}(n_A \cdot n_B) \in$ [0, PI] (see geom3d)
  - sharp features (e.g. dihedral angles <90 degrees)
- Signed dihedral-angle: $\mathrm{dir} = (N_{j-1} \times N_j) \cdot e_{jo}$
  - dir>0, the edge is ridge;
  - dir<0, the edge is ravine.







dihedral-angle

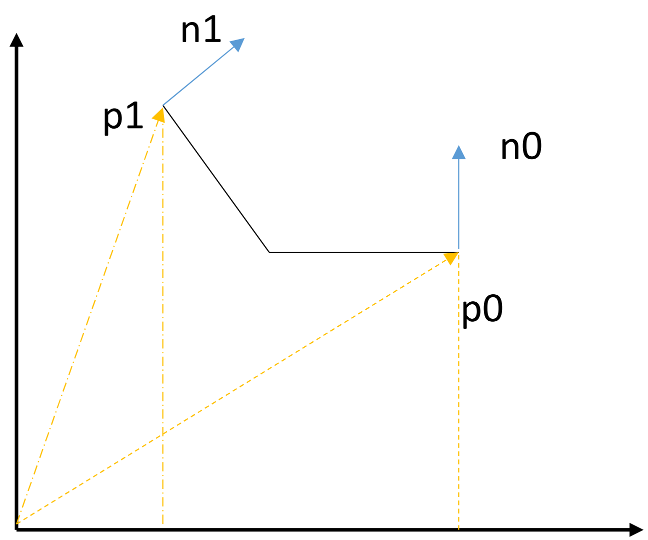


- $\varphi_{AB} = \mathrm{acos}(n_A \cdot n_B)$ ∈ [0, PI] (see geom3d)
- sharp features (e.g. dihedral angles <90 degrees)

- Signed dihedral-angle
  - A surface may depart from planarity by a positive or a negative dihedral angle (convex or concave).
  - vcg::face::DihedralAngleRad (FaceType &f, const int i)
  - Compute the signed dihedral angle between the normals of two adjacent faces.
  - It simply use the projection of the opposite vertex onto the plane of the other one.

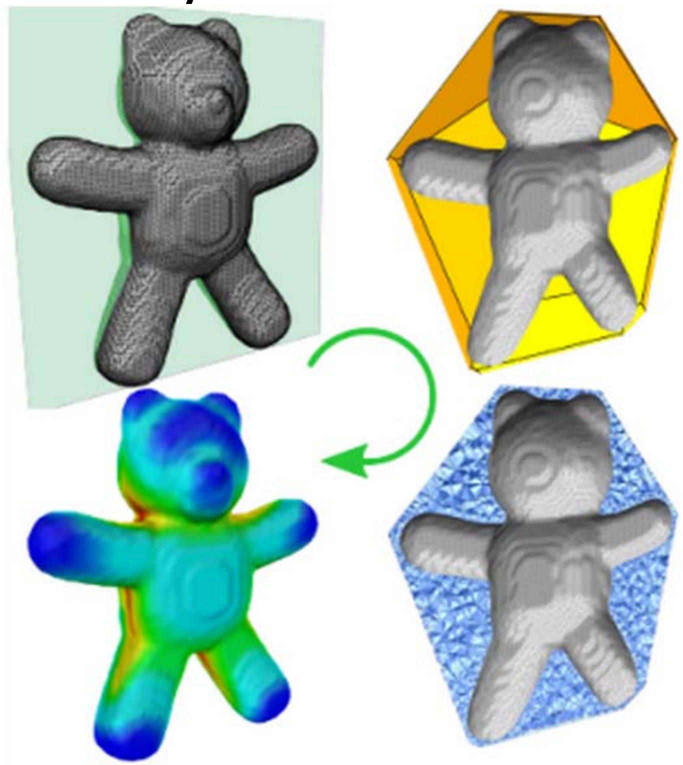
Signed dihedral-angle v.s. Convex & Concave

- vcg::face::DihedralAngleRad (FaceType &f, const int i)
- It simply use the projection of the opposite vertex onto the plane of the other one.
    - dist01 = n0*p0- n0*p1;
    - dist10 = n1*p1- n1*p0
    - // just to be sure use the sign of the largest in absolute value;
    - if(fabs(dist01) > fabs(dist10)) sign = dist01;
    - else sign=dist10;
- Positive for convex & negative for concave



Global concavity

- cvpr13_Efficient Computation of Shortest Path-Concavity for 3D Meshes, has source code



Curvature

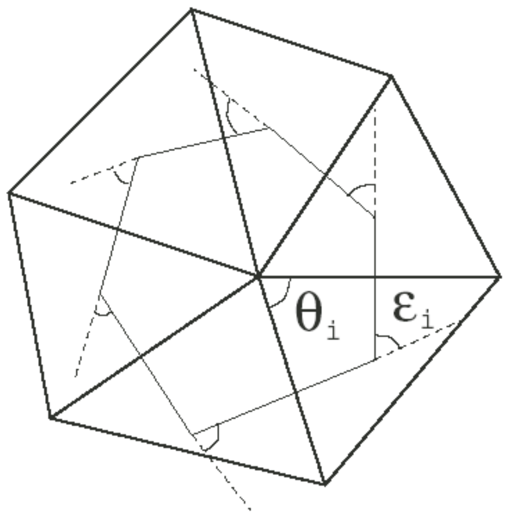
$$\Delta_S \mathbf{x} = -2H\mathbf{n}.$$

$$H(v_i) = \frac{1}{2}\|\Delta \mathbf{x}_i\|$$

Gauss-Bonnet theorem

$$\int_M K\, dA + \int_{\partial M} k_g\, ds = 2\pi\chi(M)$$

$$K(v_i) = \frac{1}{A_i}\left(2\pi - \sum_{v_j \in \mathcal{N}_1(v_i)} \theta_j\right)$$

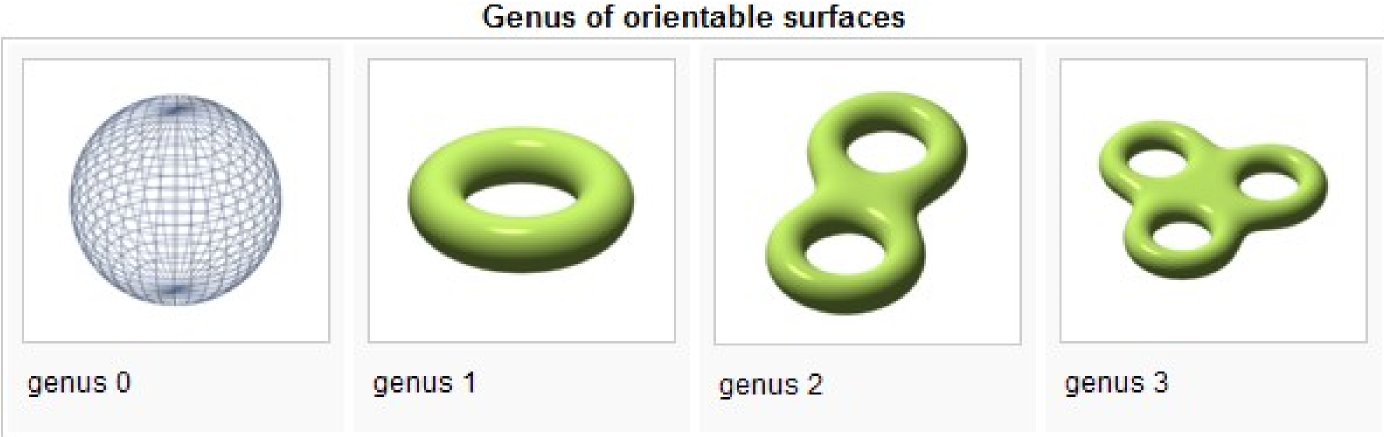


$$k_H = \frac{k_1+k_2}{2}, k_G = k_1 \cdot k_2$$

$$\kappa_{1,2}(v_i) = H(v_i) \pm \sqrt{H(v_i)^2 - K(v_i)}.$$

Euler characteristic and Genus

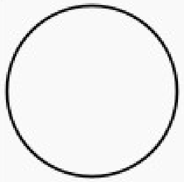
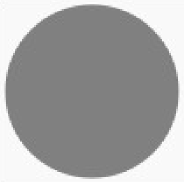
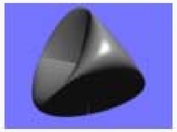
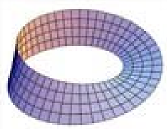
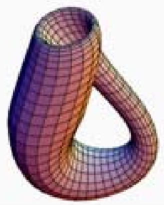
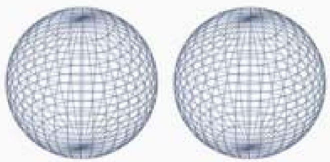
- Genus



- Euler characteristic
  - X=V -E + F (surfaces of polyhedra, or planar graphs)
    - 2: any convex polyhedron's surface
    - 1: a triangular mesh homeomorphic to a disc
  - X=2-2g (closed orientable surfaces) (g: genus)
  - X=2-2g-b (orientable surfaces with b boundary components)

Euler characteristic

- The Euler characteristic can be calculated easily for general surfaces by finding a polygonization of the surface

Name	Image	Euler characteristic
Interval		1
Circle		0
Disk		1
Sphere		2
Torus (Product of two circles)		0
Double torus		-2
Triple torus		-4
Real projective plane		1
Möbius strip		0
Klein bottle		0
Two spheres (not connected) (Disjoint union of two spheres)		2 + 2 = 4

Euler characteristic

X=V -E + F (surfaces of polyhedra, or planar graphs)

> 2: any convex polyhedron's surface
>
> 1: a triangular mesh homeomorphic to a disc

X=2-2g (closed orientable surfaces) (g: genus)

X=2-2g-b (orientable surfaces with b boundary components)

Gauss-Bonnet Theorem

- Suppose M is a compact two-dimensional Riemannian manifold with boundary ∂M. Let K be the Gaussian curvature of M, and let k_g be the geodesic curvature of ∂M. Then
- $\iint_M KdA + \int_{\partial M} k_g ds = 2\pi\chi(M)$
- Where dA is the element of area of the surface, and ds is the line element, along the boundary of M. $\chi(M)$ is the Euler characteristic of M.
- If the boundary ∂M is piecewise smooth, then we interpret the integral $\int_{\partial M} k_g ds$ as the sum of the corresponding integrals along the smooth portions of the boundary, plus the sum of the angles by which the smooth portions turn at the corners of the boundary.

Gauss-Bonnet Theorem

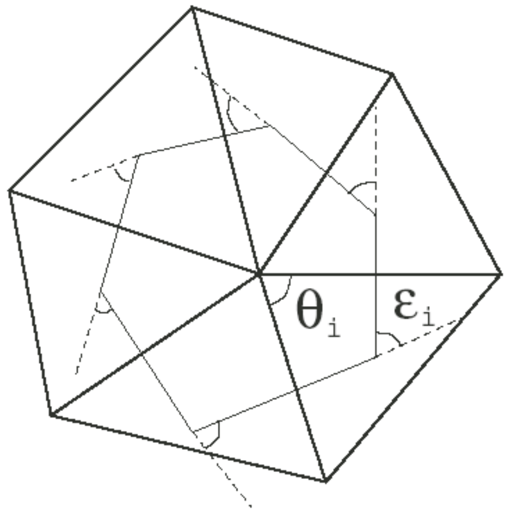
- For a closed surface M
  - Total Gauss Curvature:

$$\int_M K dA = 2\pi x(M)$$, where V-E+F=2-2g=χ

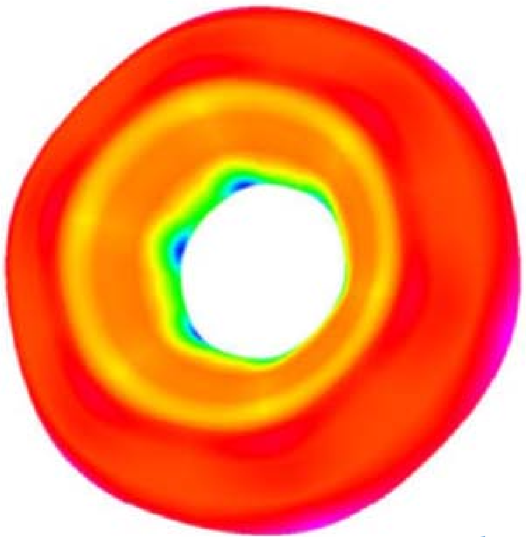
- For a surface patch
  - For voronoi region:

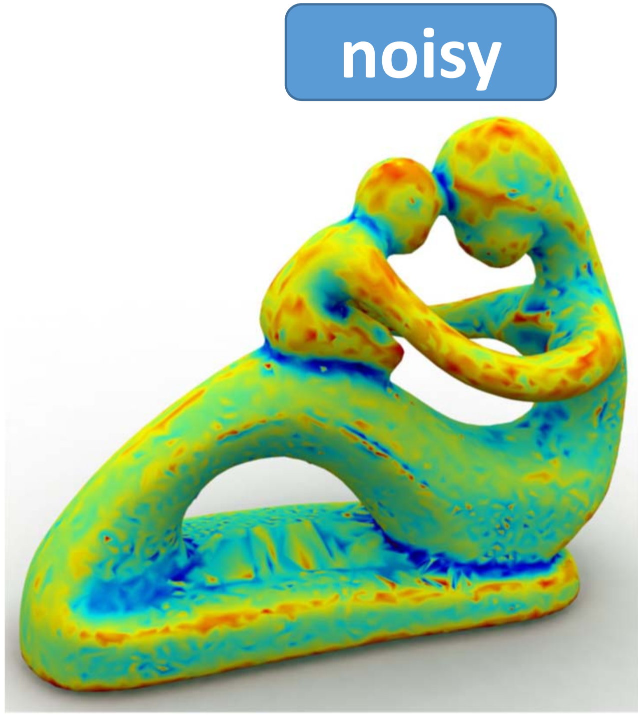
$$\iint_{A_M} \kappa_G dA + \int_{\partial A_M} \kappa_g dl = 2\pi$$

$$\iint_{A_M} \kappa_G dA = 2\pi - \sum_{j=1}^{\# f} \theta_j$$



Curvature





K=k1k2

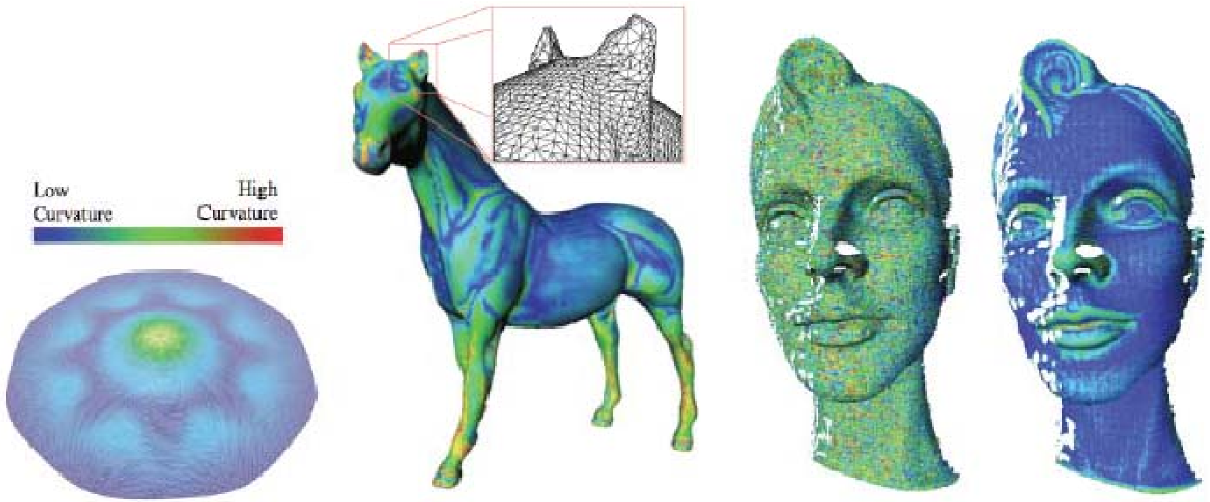
Curvature Computation

- Approaches:
  - Discrete differential geometry: Normal Cycle
  - Smooth differential geometry: Jet-fitting
- Implementation:
  - toolbox_mesh/compute_curvature
  - CGAL
  - MeshLab
  - 3d-workspace

Links and literature

- M. Meyer, M. Desbrun, P. Schroeder, A. Barr *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*, VisMath, 2002

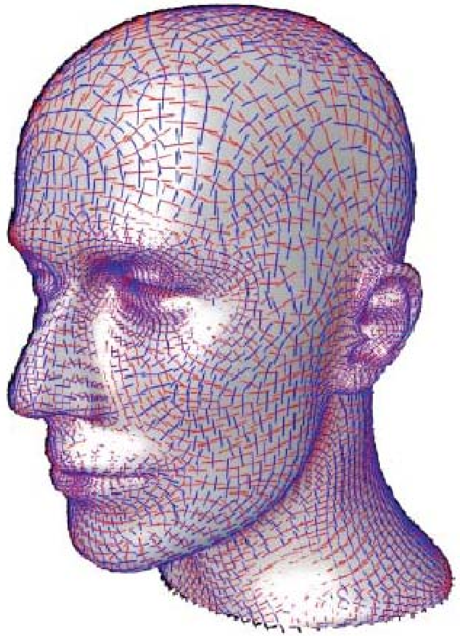
- *[Hamann 93]* simple way to determine principal curvature and direction using least-squared paraboloid fitting
    - No easy way to selecting an appropriate tangent plane



Links and literature

- P. Alliez, *Estimating Curvature Tensors on Triangle Meshes*, Source code!

- *[Taubin 95]* introduced a complete derivation of surface properties approximating curvature tensors for polyhedral surface



principal directions

References

- Gaussian curvature: tog06_Salient geometric features for partial shape matching and similarity.