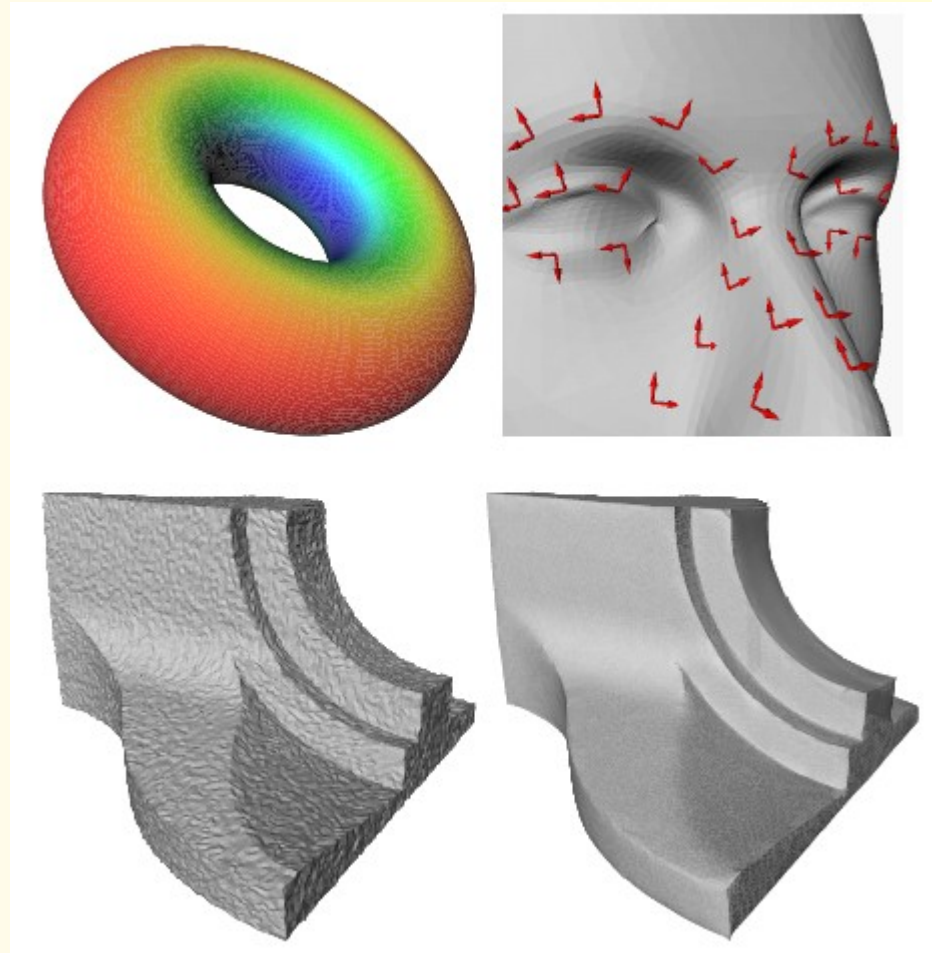# Discrete Differential Geometry Operators for Triangulated 2-Manifolds

Mark Meyer, Mathieu Desbrun, Peter Schroder, and Alan H. Barr
Presented by Christopher Batty

Basic Premise

How can we extend differential geometry to meshes?

Goal – Consistent operators for first and second order differential properties:
- Normal
- Curvatures (Mean, Gaussian, Principal),
- Principal Directions

We've already looked at a couple of approaches.

Why?

Numerous mesh-processing applications:
- Mesh smoothing/enhancement
- Re-meshing
- Parameterization
- Subdivision
- Simplification
- Etc.

Also, applications in higher dimensions :
- Denoising of data

Approach

Take properties at vertices to be spatial average of a local region of the mesh (integrate), as seen in class.
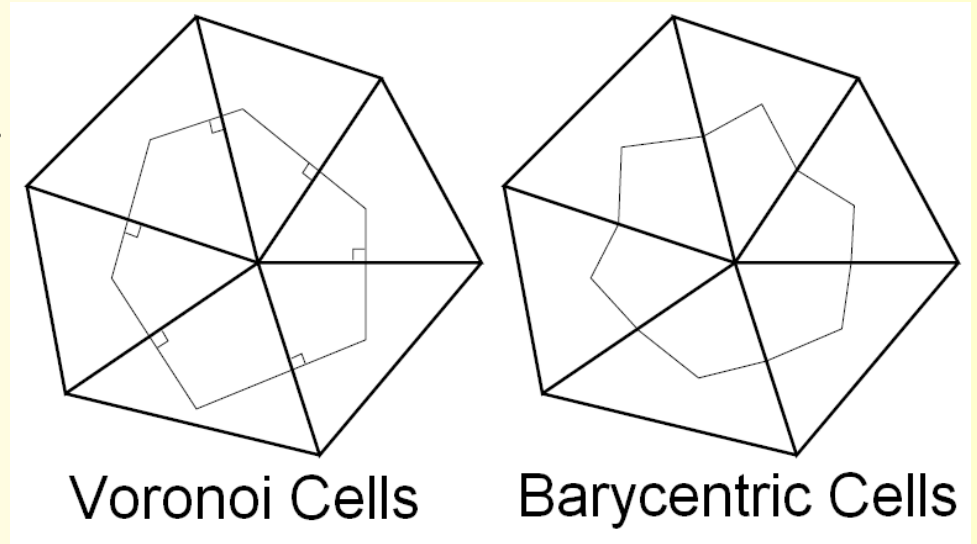
Choose regions carefully to ensure maximum accuracy.

Derive expressions for differential operators that respect the same properties as continuous counterparts.
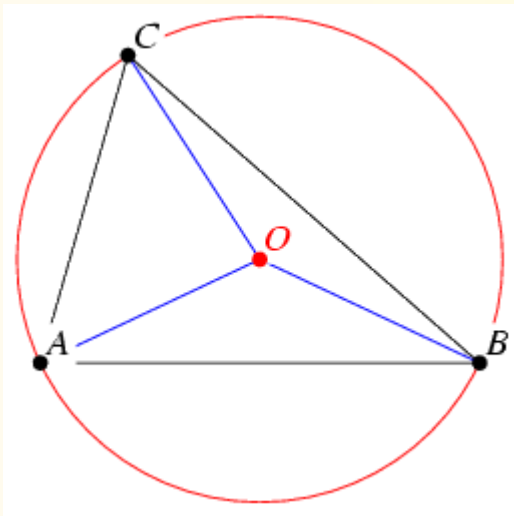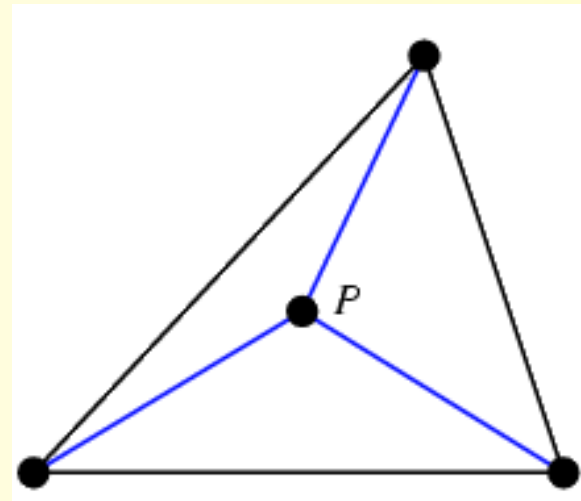
Choosing Regions

Split edges at midpoints.

How to join midpoints?

Voronoi Cells   Barycentric Cells

Either connect to:
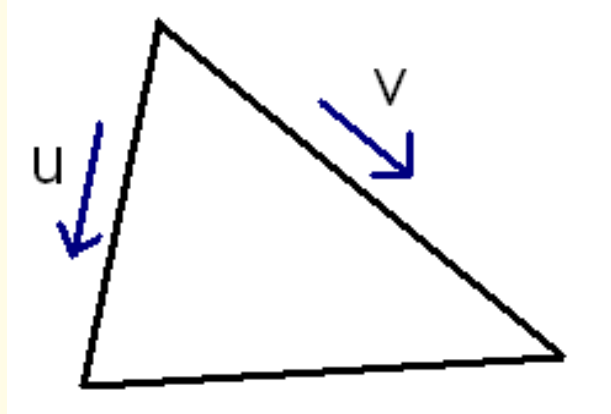   circumcenter   barycenter (avg vertices).

Mean Curvature Normal

Given a point P, the mean curvature normal (AKA Laplace-Beltrami) operator is:

$$\mathbf{K}(P) = 2\kappa_H(P)\,\mathbf{n}(P)$$

Gives both the mean curvature and unit normal at the vertex.

Mean Curvature Normal

For a triangle mesh, parameterized by u,v, along edges, over a region M, this is a Laplacian:

$$\iint_{\mathcal{A}_M} \mathbf{K}(\mathbf{x})\ dA = -\iint_{\mathcal{A}_M} \Delta_{u,v}\mathbf{x}\ du\ dv$$

where:

$$\Delta_{u,v}\mathbf{x} = \mathbf{x}_{uu} + \mathbf{x}_{vv}$$

Gauss' Theorem

Gauss' theorem lets us convert an area integral into a line integral around the boundary.

$$\iint_{\mathcal{A}_M} \Delta_{u,v}\mathbf{x}\ du\ dv = \int_{\partial \mathcal{A}_M} \nabla_{u,v}\mathbf{x} \cdot \mathbf{n}_{u,v}\ dl$$

Since our mesh is piece-wise linear, the gradient of x is a constant over each triangle.
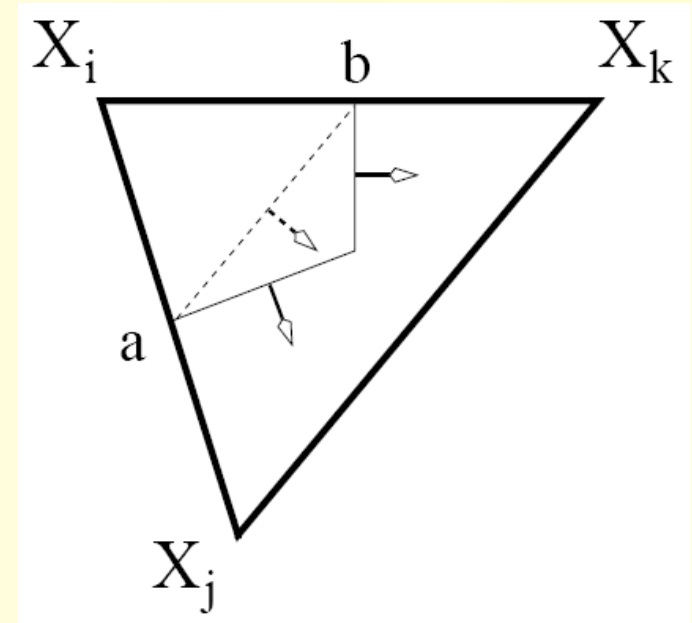
Mean Curvature Normal

Because the gradient is constant, our choice of region will not affect the integral's value.

Thus we can simply use a straight line.

For the integral in one triangle we then have:

$$\int_{\partial \mathcal{A}_{\mathrm{M}} \cap T} \nabla_{u,v}\mathbf{x} \cdot \mathbf{n}_{u,v}\, dl = \nabla_{u,v}\mathbf{x} \cdot [\mathbf{a} - \mathbf{b}]_{u,v}^{\perp} = \frac{1}{2}\nabla_{u,v}\mathbf{x} \cdot [\mathbf{x}_j - \mathbf{x}_k]_{u,v}^{\perp}$$
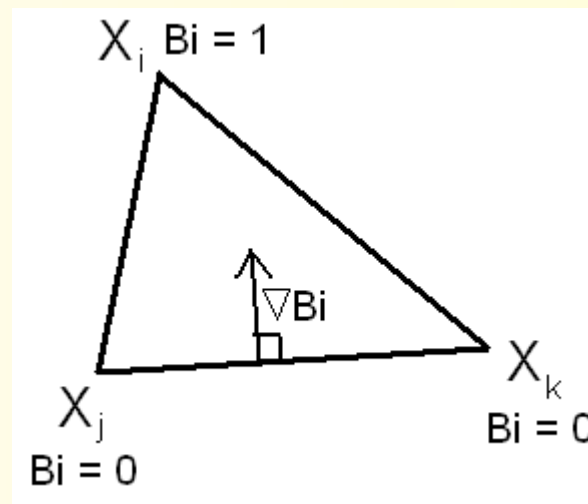
Mean Curvature Normal

Now, note that any point in the triangle is a linear combination of its vertices.

$$\mathbf{x} = \mathbf{x}_i \; B_i(u,v) + \mathbf{x}_j \; B_j(u,v) + \mathbf{x}_k \; B_k(u,v)$$

Taking the gradient gives:

$$\nabla_{u,v}\mathbf{x} = \mathbf{x}_i \; \nabla_{u,v}B_i(u,v) + \mathbf{x}_j \; \nabla_{u,v}B_j(u,v) + \mathbf{x}_k \; \nabla_{u,v}B_k(u,v)$$

Mean Curvature Normal

The gradient of the B's are perpendicular to the opposing edge and sum to 0, so rearranging:

$$\nabla_{u,v}\mathbf{x} = \frac{1}{2\mathcal{A}_T}\left[(\mathbf{x}_j - \mathbf{x}_i)\left([\mathbf{x}_i - \mathbf{x}_k]_{u,v}^{\perp}\right)^T + (\mathbf{x}_k - \mathbf{x}_i)\left([\mathbf{x}_j - \mathbf{x}_i]_{u,v}^{\perp}\right)^T\right]$$
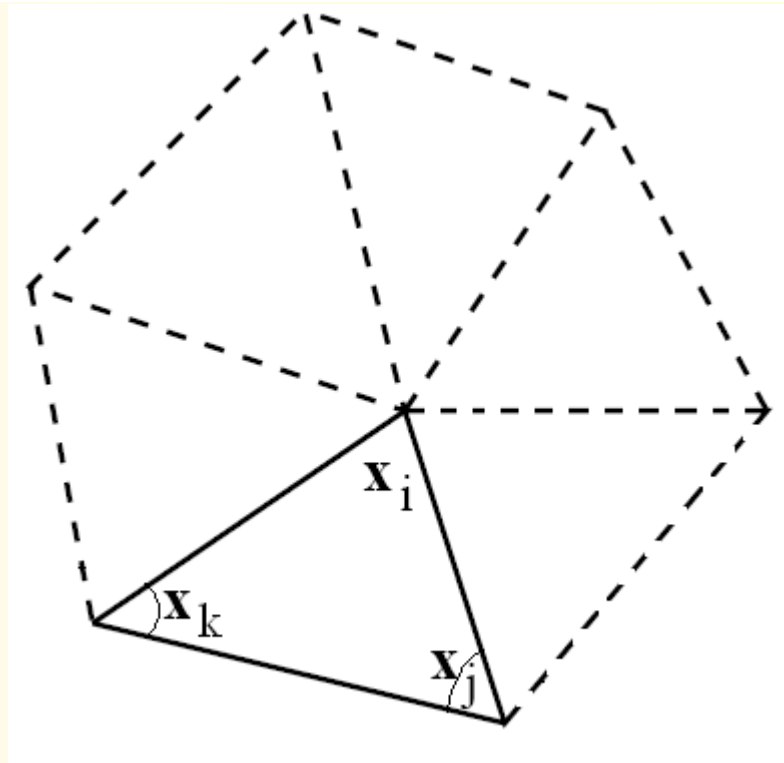
Applying this to the earlier integral gives:

$$\int_{\partial\mathcal{A}\cap T}\nabla_{u,v}\mathbf{x}\cdot\mathbf{n}_{u,v}\,dl = \frac{1}{4\mathcal{A}_T}\left[\left([\mathbf{x}_i - \mathbf{x}_k]\cdot[\mathbf{x}_j - \mathbf{x}_k]\right)_{u,v}(\mathbf{x}_j - \mathbf{x}_i)\right.$$
$$\left. +\left([\mathbf{x}_j - \mathbf{x}_i]\cdot[\mathbf{x}_j - \mathbf{x}_k]\right)_{u,v}(\mathbf{x}_k - \mathbf{x}_i)\right]$$

Now, the dot product is proportional to a cosine, and the A_T term is proportional to a sine, so...

Mean Curvature Normal

Replace with a cotangent (= cos/sin) to get:

$$\int_{\partial \mathcal{A}_{M} \cap T} \nabla_{u,v} \mathbf{x} \cdot \mathbf{n}_{u,v} \, dl = \frac{1}{2} [cot_{u,v} \angle(\mathbf{x}_k)(\mathbf{x}_j - \mathbf{x}_i) + cot_{u,v} \angle(\mathbf{x}_j)(\mathbf{x}_k - \mathbf{x}_i)]$$
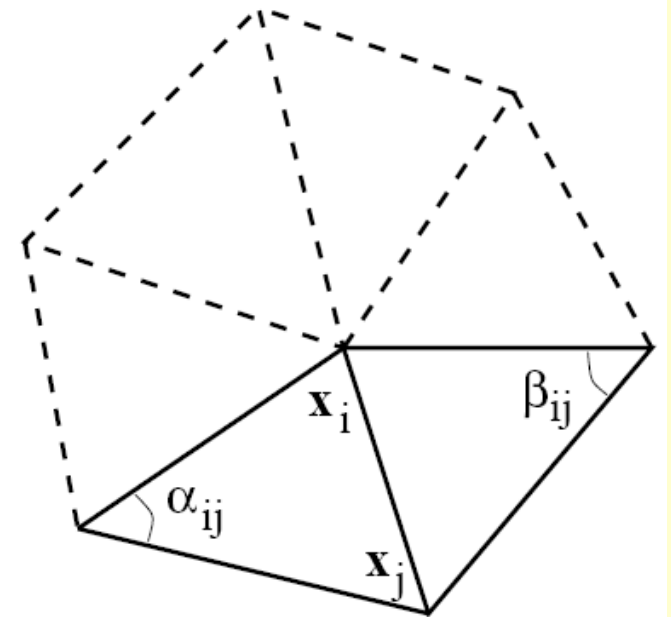
Mean Curvature Normal

And finally...

$$\iint_{\mathcal{A}_M} \mathbf{K}(\mathbf{x}) dA = \frac{1}{2} \sum_{j \in N_1(i)} \left(cot\ \alpha_{ij} + cot\ \beta_{ij} \right) \left(\mathbf{x}_i - \mathbf{x}_j \right)$$

So what have we done exactly?

Derived a simple expression for mean curvature normal depending only on vertex positions and angles of the adjacent triangles.

Error Analysis

Consider a C^2 surface, tiled with our triangulated C^0 surface.

We can express the error between our spatial average and the correct pointwise average as:

$$E = \sum_i \iint_{\mathcal{A}_i} \left\| \mathbf{K}(\mathbf{x}) - \mathbf{K}(\mathbf{x}_i) \right\|^2 dA$$

If C_i is the Lipschitz constant for K(x), then:

$$\leq \sum_i \iint_{\mathcal{A}_i} C_i^2 \left\| \mathbf{x} - \mathbf{x}_i \right\|^2 dA$$

Error Analysis

Taking the C_i's outside the summation:

$$E \leq C_{max}^2 \sum_i \iint_{\mathcal{A}_i} \|\mathbf{x} - \mathbf{x}_i\|^2 \, dA$$
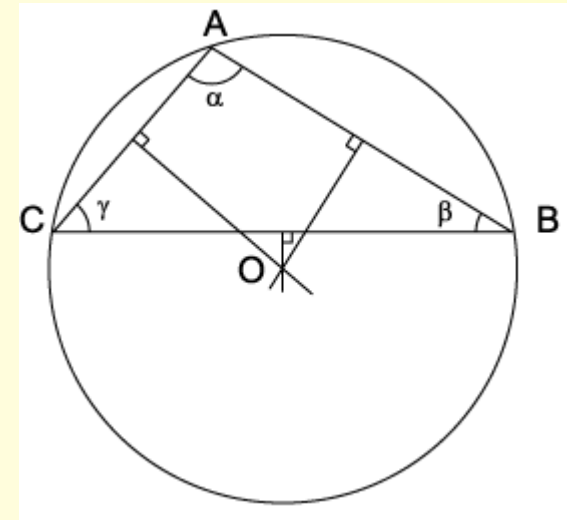
Recall that Voronoi regions minimize $\|x-x_i\|$ by definition.

Assuming C varies with ratio ε from patch to patch, we are guaranteed that the Voronoi region is no more than O(ε) from the true optimal patch.

But...

This is only valid for non-obtuse triangles.

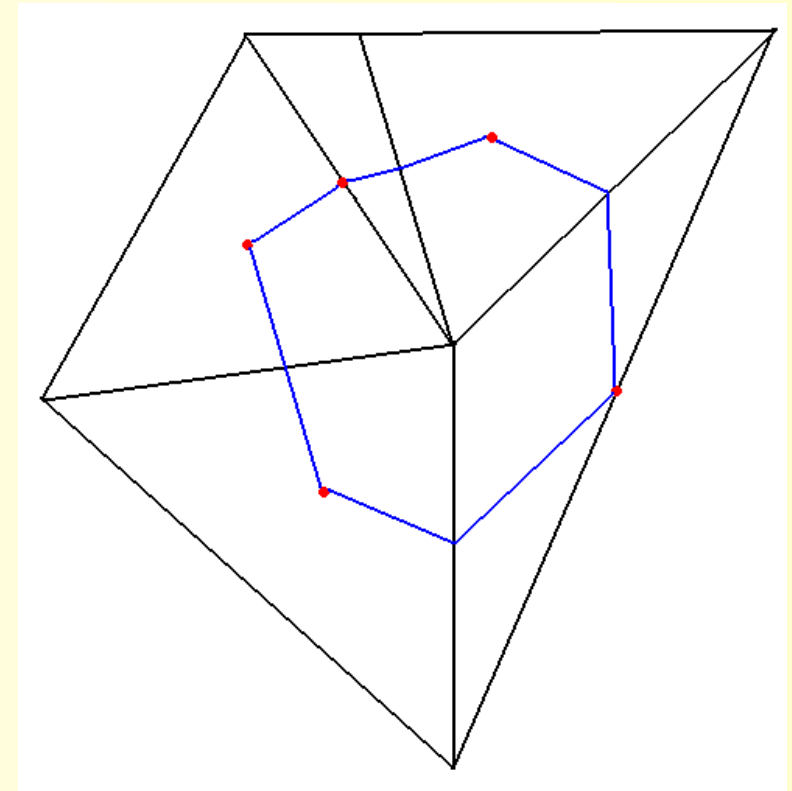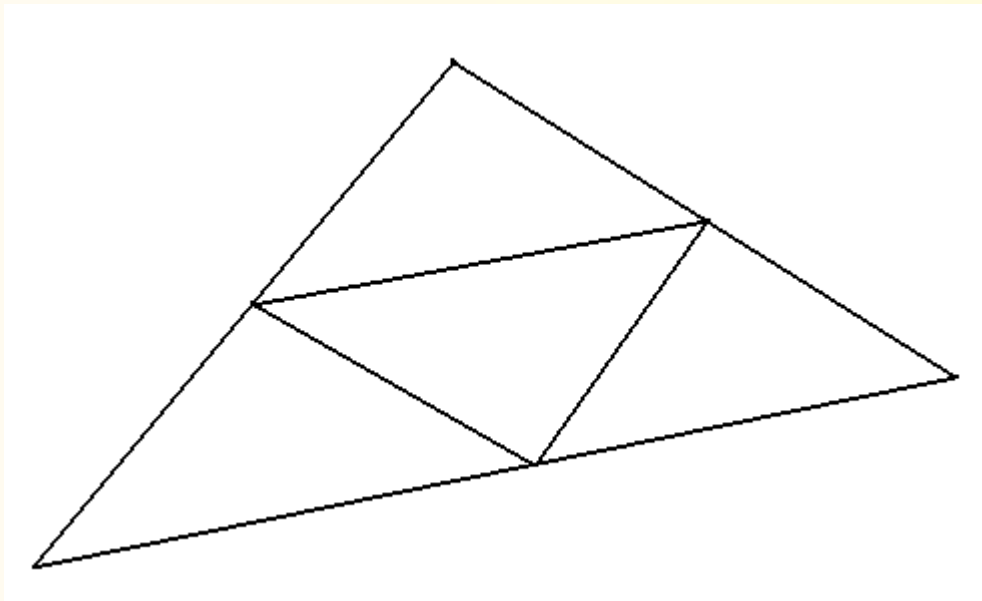For obtuse triangles the circumcenter will lie outside of the triangle.

Work-around: Use the midpoint of the edge opposite to the obtuse angle as the center.

Gives a non-overlapping tiling of the surface using Voronoi cells where possible.

Obtuse Triangles

Example of mixed areas around a vertex:

Area of non-Voronoi regions is either $A_{triangle}/2$ or $A_{triangle}/4$:

Summary So Far

Area computation:

$$\mathcal{A}_{\text{Mixed}} = 0$$

For each triangle T from the 1-ring neighborhood of $\mathbf{x}$

    If T is non-obtuse,   // Voronoi safe

        // Add Voronoi formula (see Section 3.3)

        $\mathcal{A}_{\text{Mixed}} + = $ Voronoi region of $\mathbf{x}$ in T

    Else   // Voronoi inappropriate

        // Add either area(T)/4 or area(T)/2

        If the angle of T at $\mathbf{x}$ is obtuse

            $\mathcal{A}_{\text{Mixed}} + = \text{area}(T)/2$

        Else

            $\mathcal{A}_{\text{Mixed}} + = \text{area}(T)/4$

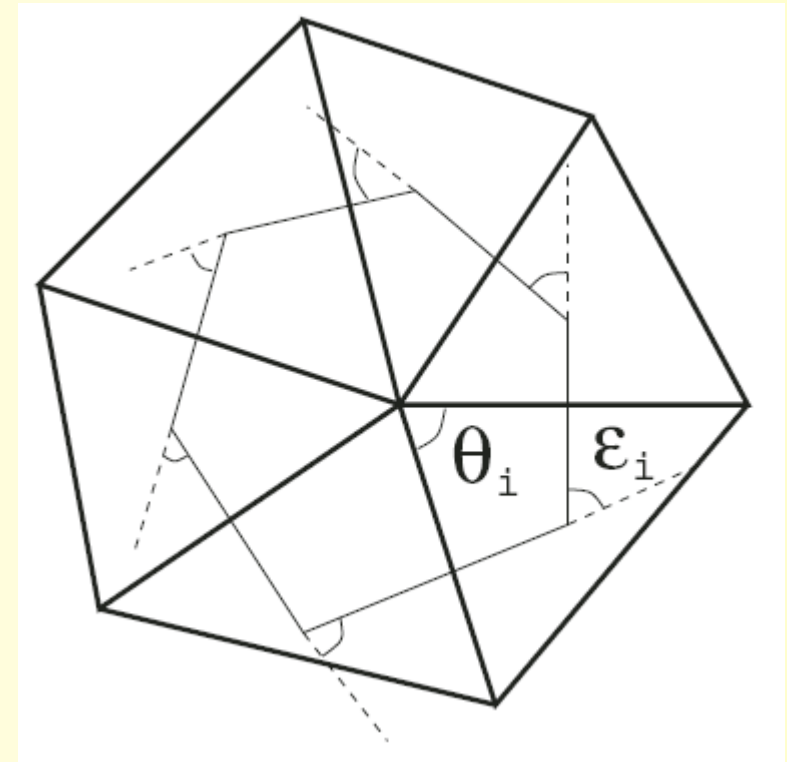Mean curvature normal operator:

$$\mathbf{K}(\mathbf{x}_i) = \frac{1}{2\mathcal{A}_{\text{Mixed}}} \sum_{j \in N_1(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) \ (\mathbf{x}_i - \mathbf{x}_j)$$

Gaussian Curvature

Easy! We looked at it in class already – use the Gauss-Bonnet theorem (but with new region definition).

Gaussian Curvature operator:

$$\kappa_G(\mathbf{x}_i) = \left(2\pi - \sum_{j=1}^{\#f} \theta_j\right)/\mathcal{A}_{\text{Mixed}}$$

Principal Curvatures

We now have both the mean and Gaussian curvatures, and we know the relationships:

$$\kappa_G = \kappa_1 \kappa_2 \qquad\qquad \kappa_H = (\kappa_1 + \kappa_2)/2$$
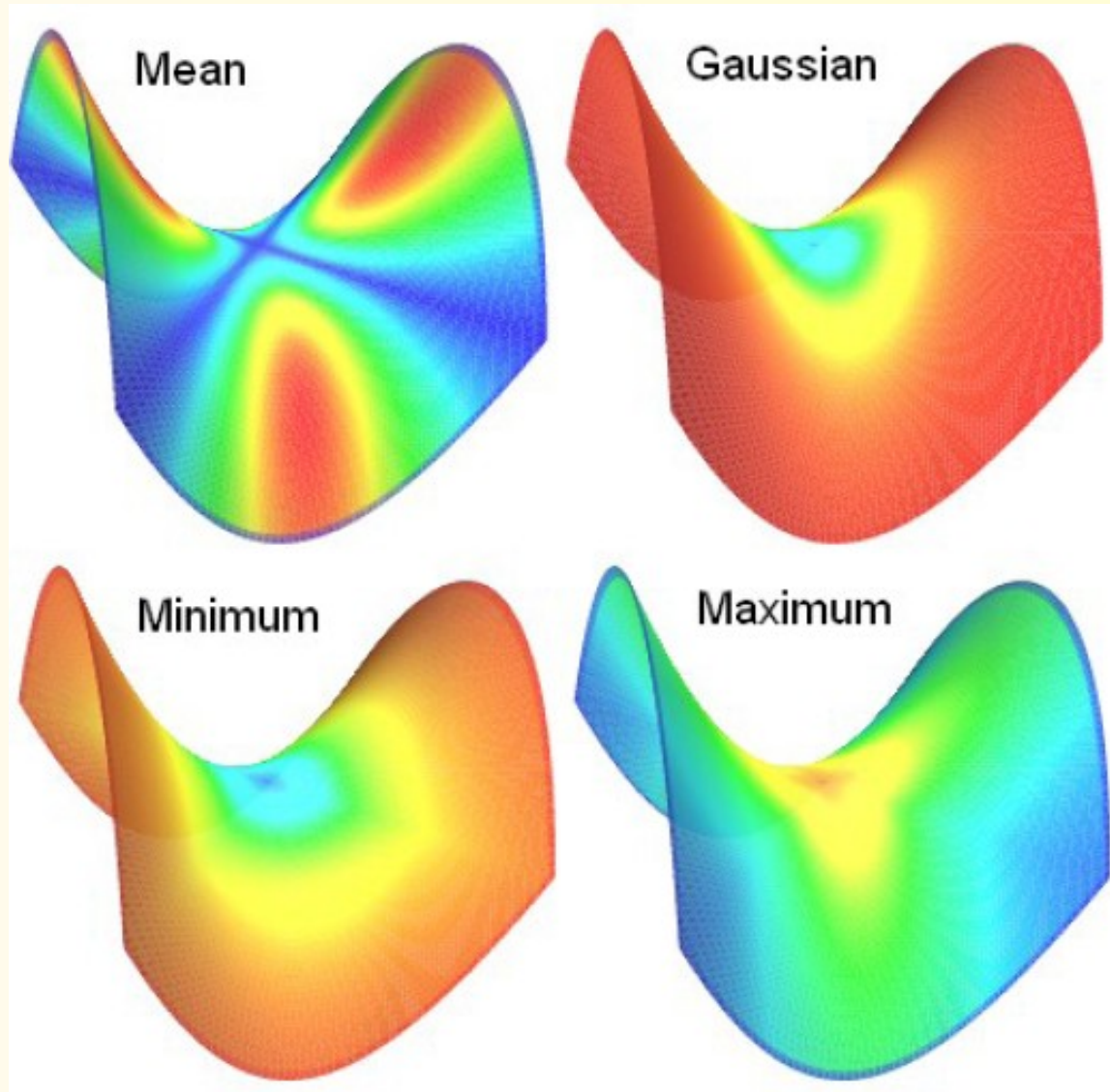
We can rearrange and solve.

Principal Curvature operators:

$$\kappa_1(\mathbf{x}_i) = \kappa_H(\mathbf{x}_i) + \sqrt{\Delta(\mathbf{x}_i)}$$

$$\kappa_2(\mathbf{x}_i) = \kappa_H(\mathbf{x}_i) - \sqrt{\Delta(\mathbf{x}_i)}$$

with: $\quad \Delta(\mathbf{x}_i) = \kappa_H^2(\mathbf{x}_i) - \kappa_G(\mathbf{x}_i)$

Curvature Visualization

Principal Directions

Before I get into (more) math, here's the overview:

Mean curvature formula can be seen as a weighted sum of normal curvatures along adjacent edges.

Use these samples of curvature normals to fit an ellipse with least squares, to determine a curvature tensor.

Extract eigenvectors from this tensor – these are the principal directions.

Mean Curvature, Interpreted

$$\kappa_H(\mathbf{x}_i) = \frac{1}{2}\left(2\kappa_H(\mathbf{x}_i)\mathbf{n}\right)\cdot\mathbf{n} = \frac{1}{2}\mathbf{K}(\mathbf{x}_i)\cdot\mathbf{n}$$

$$= \frac{1}{4\mathcal{A}_{\text{Mixed}}}\sum_{j\in N_1(i)}\left(cot\ \alpha_{ij} + cot\ \beta_{ij}\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)\cdot\mathbf{n}$$

$$= \frac{1}{4\mathcal{A}_{\text{Mixed}}}\sum_{j\in N_1(i)}\left(cot\ \alpha_{ij} + cot\ \beta_{ij}\right)\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\|\mathbf{x}_i - \mathbf{x}_j\|^2}\left(\mathbf{x}_i - \mathbf{x}_j\right)\cdot\mathbf{n}$$

$$= \frac{1}{\mathcal{A}_{\text{Mixed}}}\sum_{j\in N_1(i)}\left[\frac{1}{8}\left(cot\ \alpha_{ij} + cot\ \beta_{ij}\right)\|\mathbf{x}_i - \mathbf{x}_j\|^2\right]\kappa^N_{i,j},$$

Where:
$$\kappa^N_{i,j} = 2\frac{\left(\mathbf{x}_i - \mathbf{x}_j\right)\cdot\mathbf{n}}{\|\mathbf{x}_i - \mathbf{x}_j\|^2}$$

This an expression for curvature along an edge.

Interpretation

Right angle implies:

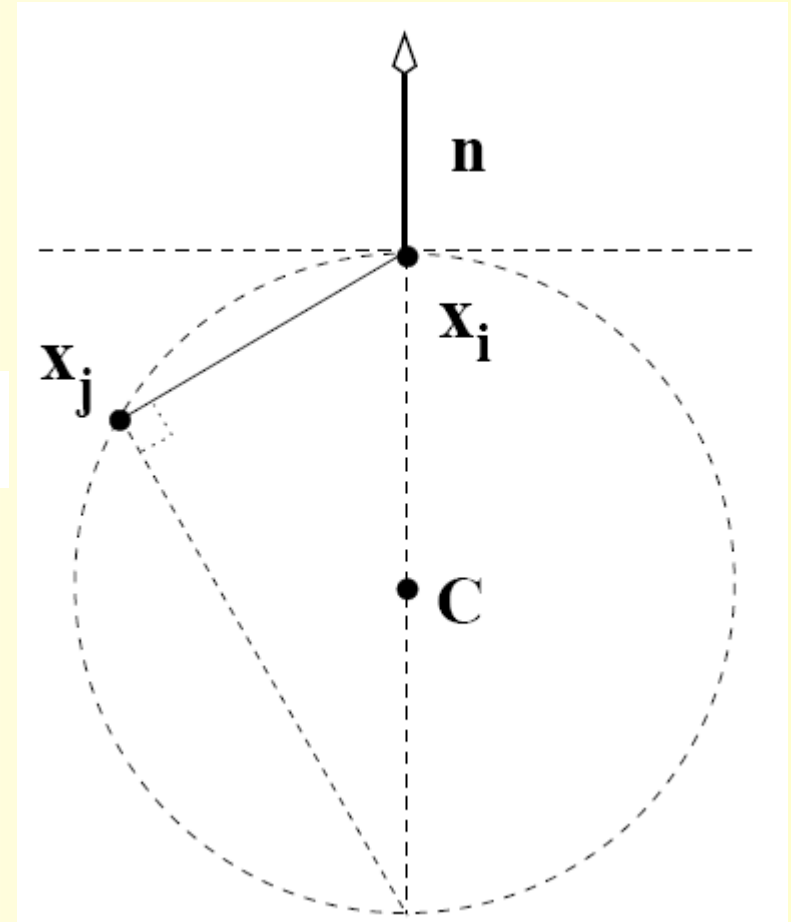$$\left(\mathbf{x}_i - \mathbf{x}_j\right) \cdot \left(\mathbf{x}_i - \mathbf{x}_j - 2R\,\mathbf{n}\right) = 0$$

Rearranged:

$$R = \|\mathbf{x}_i - \mathbf{x}_j\|^2 / \left(2\left(\mathbf{x}_i - \mathbf{x}_j\right) \cdot \mathbf{n}\right)$$

which is the inverse of what we just saw:

$$\kappa_{i,j}^N = 2\frac{\left(\mathbf{x}_i - \mathbf{x}_j\right) \cdot \mathbf{n}}{\|\mathbf{x}_i - \mathbf{x}_j\|^2}$$

Hence each $K^N_{i,j}$ is an estimate of curvature along an edge.

So?

So, our mean curvature formula can be interpreted as weighted combination of normal curvature samples along neighbouring edges.

$$\kappa_H(\mathbf{x}_i) = \sum_{j \in N_1(i)} w_{ij} \ \kappa_{i,j}^N$$

Let's fit an ellipse to these curvature samples to find a least-squares approximation of the curvature tensor.

Curvature Tensor

The (symmetric) curvature tensor looks like:

$$B = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

Applied to a tangent vector, it should give the curvature normal along that direction.

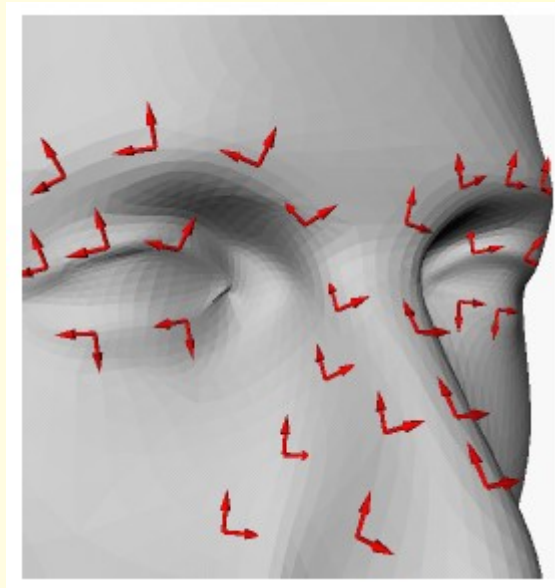$$\mathbf{d}_{i,j}^{T} \, B \, \mathbf{d}_{i,j} \; = \; \kappa_{i,j}^{N}$$

We just project each edge onto the tangent plane to get $\mathbf{d}_{i,j}$'s.

Curvature Tensor

Apply least squares minimization to the error to get B.

$$E(a, b, c) = \sum_j w_j \left(\mathbf{d}_{i,j}^T \ B \ \mathbf{d}_{i,j} - \kappa_{i,j}^N \right)^2$$

Extract the eigenvectors from B, and voila!
It *is* just that easy.
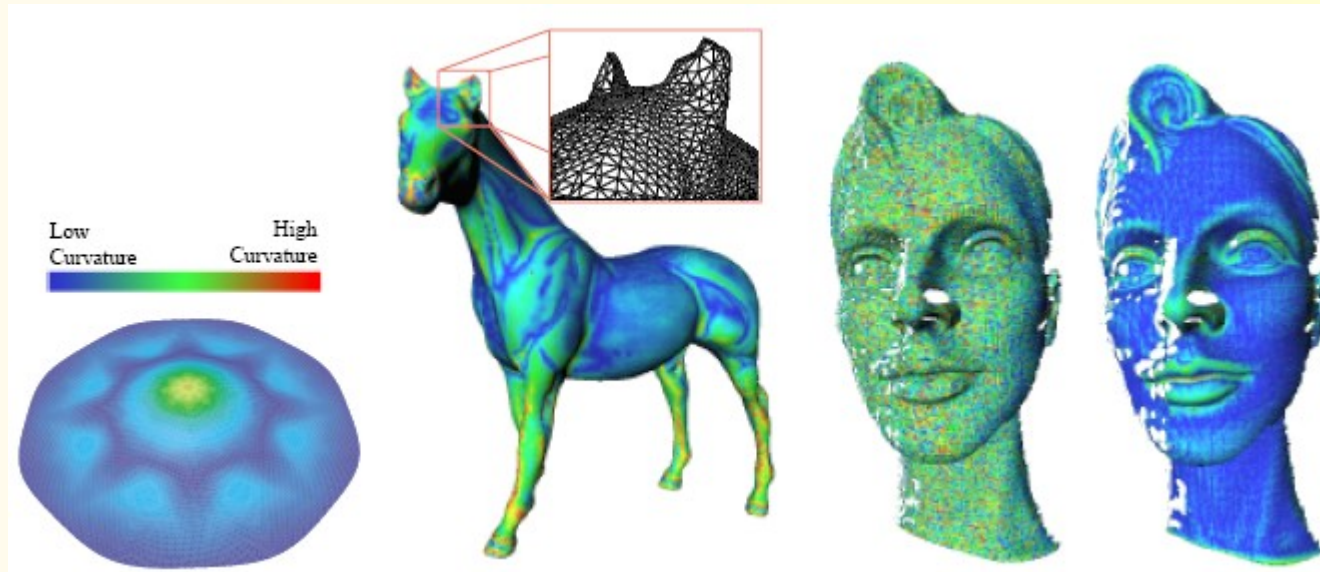
Numerical Results

Accuracy roughly equivalent to finite differences, without the restriction of regular meshes.

Compared against analytical surfaces at varying resolutions:
- Less than 1.3% error for Gaussian curvature
- Less than 0.07% error for mean curvature

Irregular sampling reduces accuracy, but "in practice, the rate at which the error increases is low."

Applications
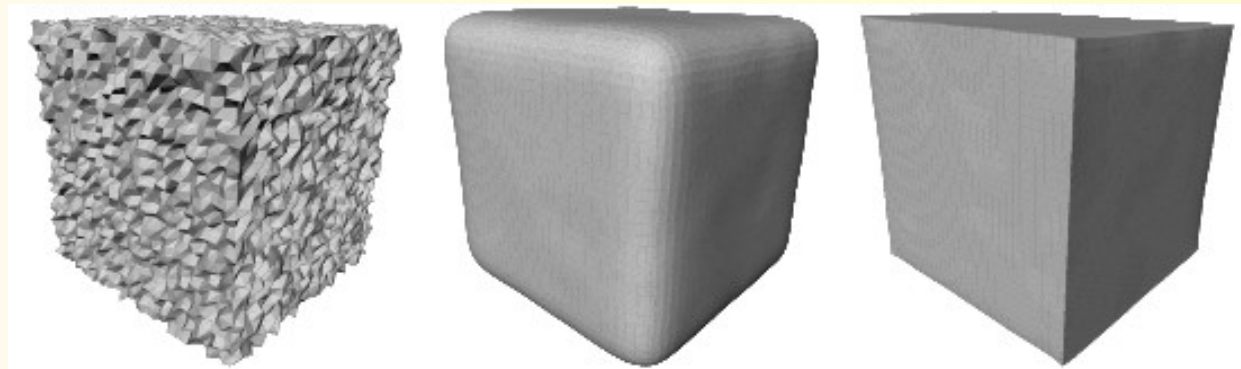
Conclusions

Is this approach an improvement?

Gives a more convincing justification for choice of both surface normal and the region type used for calculations.

Better numerical results than existing approaches.

Graceful degradation in the presence of irregular meshes (claimed).

References

M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential geometry operators for triangulated 2-manifolds," in VisMath, 2002.

http://www.wikipedia.org

http://mathworld.wolfram.com/