07 - Discrete Differential Geometry of Surfaces

(I - Curvature)

Acknowledgements: Daniele Panozzo

In this lecture

- Discrete integration
- Curvature in the discrete setting

Differential Geometry on Meshes

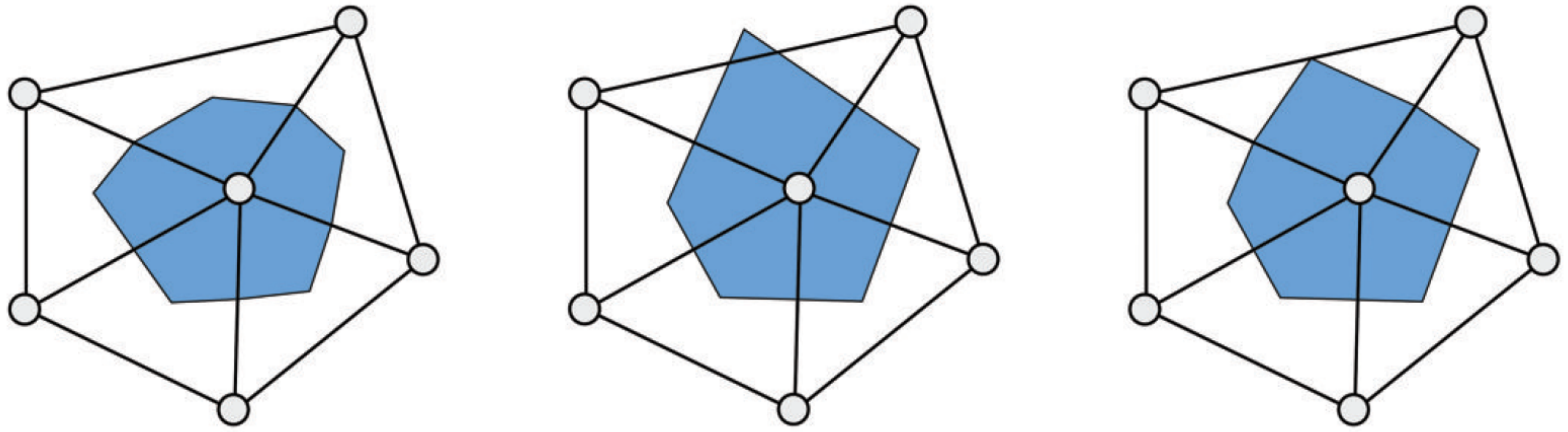
- Assumption:
  - meshes are piecewise linear approximations of smooth surfaces
- Meshes have straight-line edges and flat faces, they are just C^0 at their vertices and across edges
- Differential properties of the (unknown) approximated surface are estimated on the mesh
  - need to extend estimation across discontinuities (edges and vertices)

Differential Geometry on Meshes

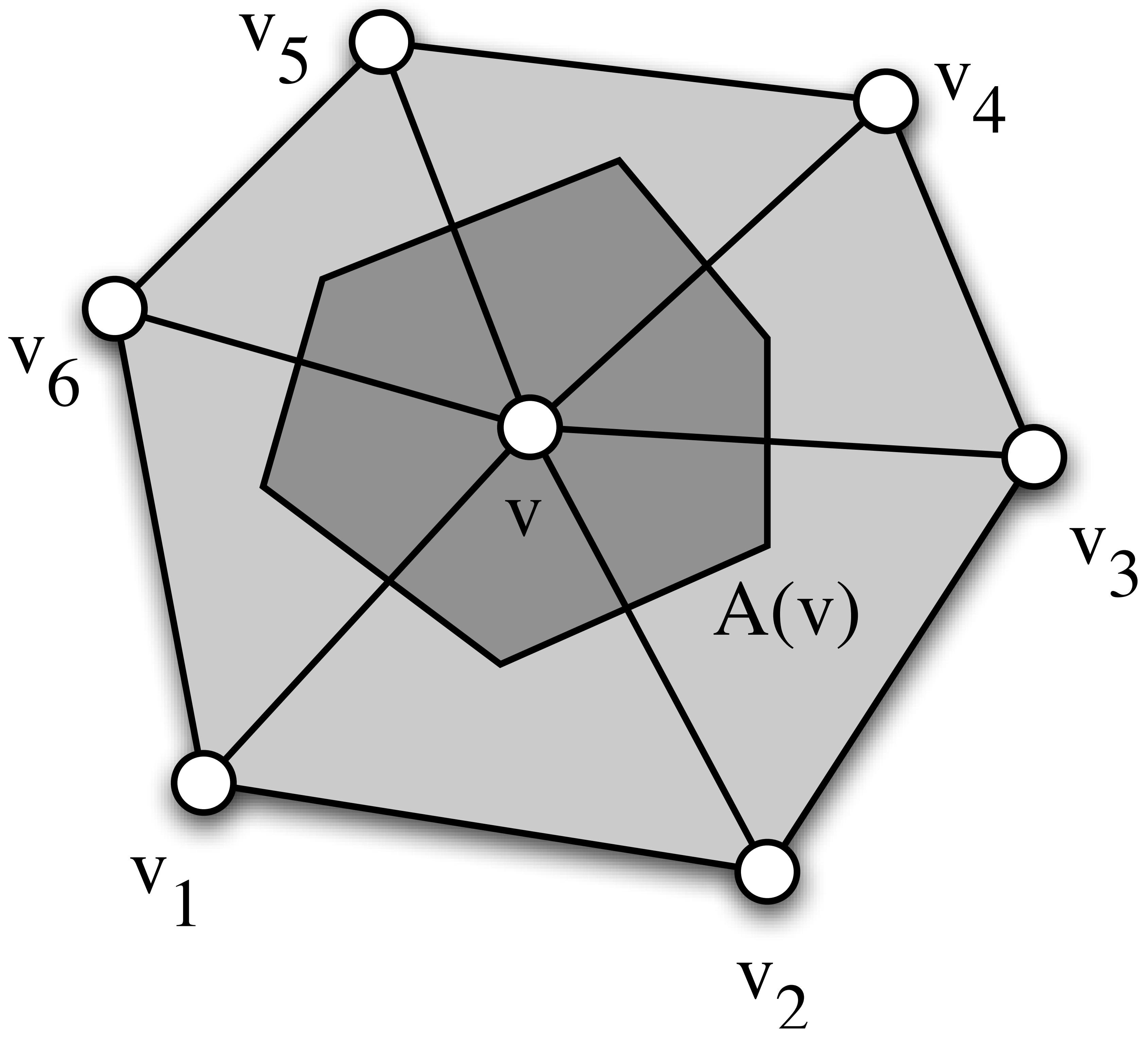
- Surface fitting: compute a differential property at point p analytically, by fitting a smooth surface to a local neighborhood $N(p)$
  - May be too slow for interactive setting
- Discrete integration: compute a differential property at point p as spatial average of a local neighborhood $N(p)$
  - For a mesh vertex v_i, the neighborhood is usually its k-ring $N_k(v_i)$ for some given value of k
- Rule of thumb: the larger the neighborhood, the more smoothing is introduced
  - more stable to noise
  - small details are lost

Discrete integration

- Vertex neighborhoods that partition the mesh into disjoint regions are often preferred:
  - barycentric cells
  - Voronoi cells
  - mixed Voronoi cells



Barycentric cell   Voronoi cell   Mixed Voronoi cell



Surface normal via discrete integration

- Surface normal of a triangle $t = (v_i, v_j, v_k)$ is the normal of the plane containing t , i.e.:

$$\mathbf{n}(t) = \frac{(v_j - v_i) \times (v_k - v_i)}{||(v_j - v_i) \times (v_k - v_i)||}$$

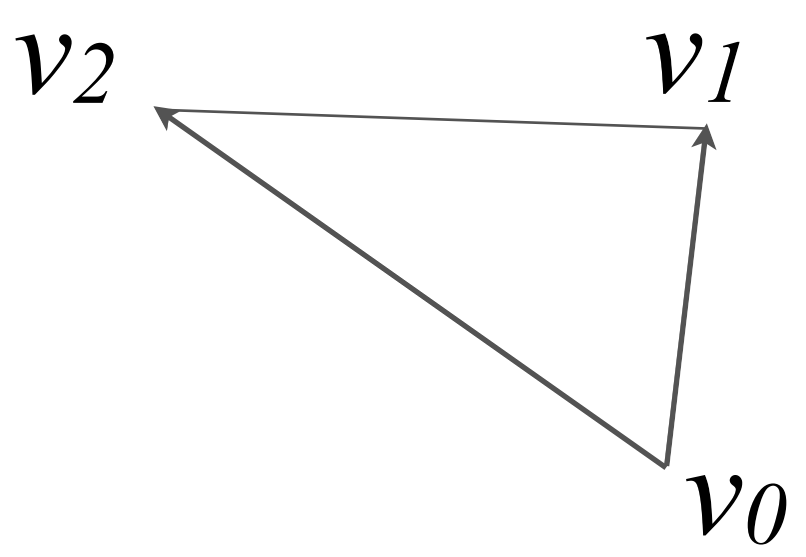
- Surface normal at a vertex v is computed as a weighted average of normals of its incident faces:

$$\mathbf{n}(v) = \frac{\sum_{t \in N_1(v)} \alpha_t \mathbf{n}(t)}{|| \sum_{t \in N_1(v)} \alpha_t \mathbf{n}(t)||}$$

where weight α_t is the area of the portion of averaging region related to t

Area and volume via discrete integration

- Total area of a mesh: $\sum_{f \in M} A_f$



- For a triangle mesh:

$$A_f = \frac{1}{2}|(v_1 - v_0) \times (v_2 - v_0)|$$

- For the volume we use the divergence theorem:

$$\iiint_{\Omega} \nabla \cdot \mathbf{F}(\mathbf{u})dV = \iint_{\partial\Omega} \mathbf{F}(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u})dA$$

Area and volume via discrete integration

- Set $\mathbf{F}(\mathbf{x}) = \frac{1}{3}\mathbf{x}$ then we have $\mathrm{div}\mathbf{F}(\mathbf{x}) = 1$ everywhere
- Therefore the volume inside a mesh M is

$$V_M = \iiint_{\Omega_M} 1 dV = \iiint_{\Omega_M} \nabla \cdot (\frac{1}{3}\mathbf{u}) dV = \iint_M \frac{1}{3}\mathbf{u} \cdot \mathbf{n}(\mathbf{u}) dA$$

- For a polygonal mesh, the last integral is computed face-by-face:

$$V_M = \frac{1}{3} \sum_{f \in M} \mathbf{x}_f \cdot \mathbf{n}_f A_f$$

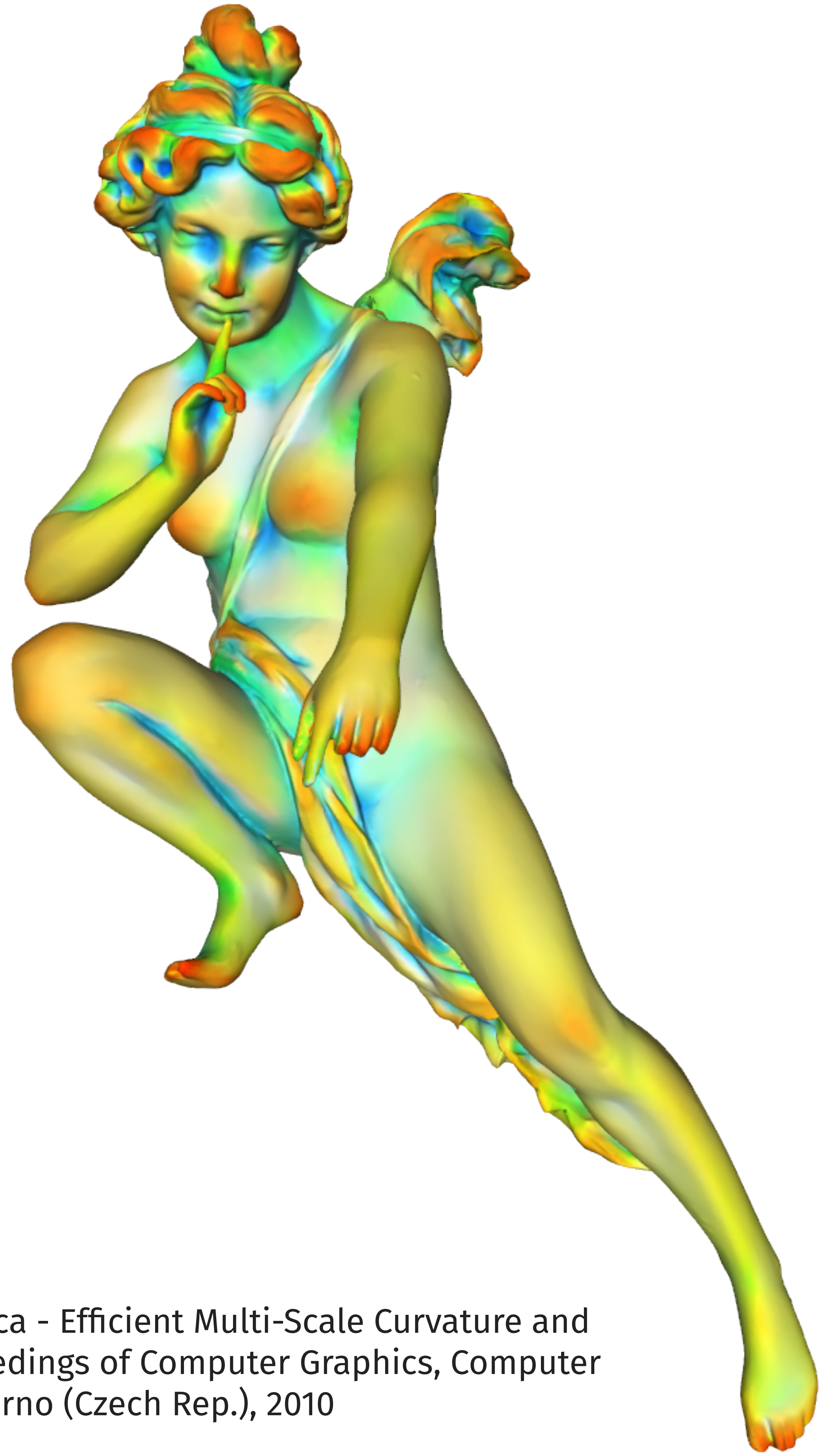
where x_f is any point inside face f (all points have the same projection on $\mathbf{n}_f$)

Discrete curvature

Shape operator via surface fitting

The radius r of the neighborhood of each point p is used as a scale parameter

1. gather all vertices in a local neighborhood of radius r
2. set a local (moving) frame at p: $(\mathbf{u},\mathbf{v},\mathbf{n}_p)$ where $\mathbf{n}_p$ is the surface normal at p and $\mathbf{u},\mathbf{v}$ are any two orthogonal unit vectors spanning the tangent plane



D Panozzo, E Puppo, L Rocca - Efficient Multi-Scale Curvature and Crease estimation - Proceedings of Computer Graphics, Computer Vision and Mathematics, Brno (Czech Rep.), 2010

Shape operator via surface fitting

3. discard all vertices v_i such that $\mathbf{n}_i \cdot \mathbf{w} < 0$
4. express all vertices of the neighborhood the local frame
5. fit to these points a polynomial of degree two through p (least squares fitting)

$$f(u,v) = au^2 + bv^2 + cuv + du + ev$$

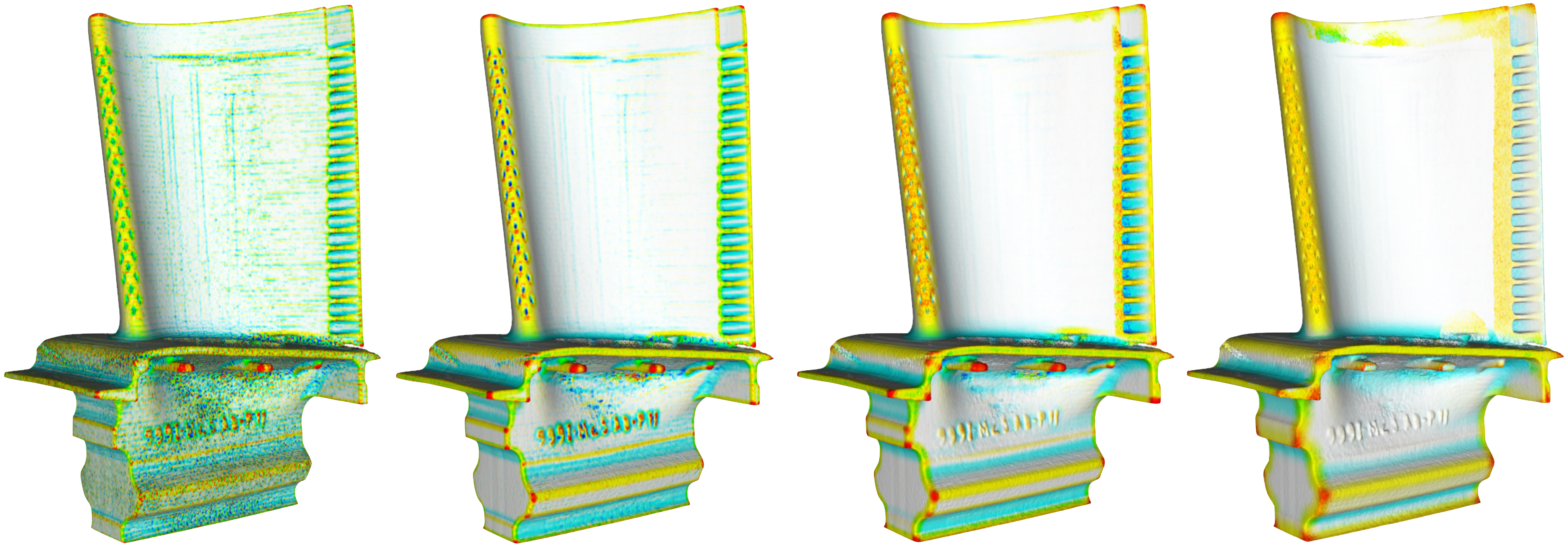
6. Shape operator at p is computed analytically via first and second fundamental forms of f at the origin

Shape operator via surface fitting

- Second order polynomial is sufficient to capture curvature
  - convergent to exact solution upon mesh refinement
- Since f is a polynomial, coefficients E, F, G, L, M, N of the fundamental forms can be computed easily in closed form (they depend just on first and second derivatives of f)
- The shape operator is expressed with respect to frame $(\mathbf{u},\mathbf{v},\mathbf{n}_p)$
- SVD decomposition of the shape operator provides principal curvatures and principal curvature directions
  - the Darboux frame can be set from such directions

Shape operator via surface fitting

- Curvatures extracted at different scales

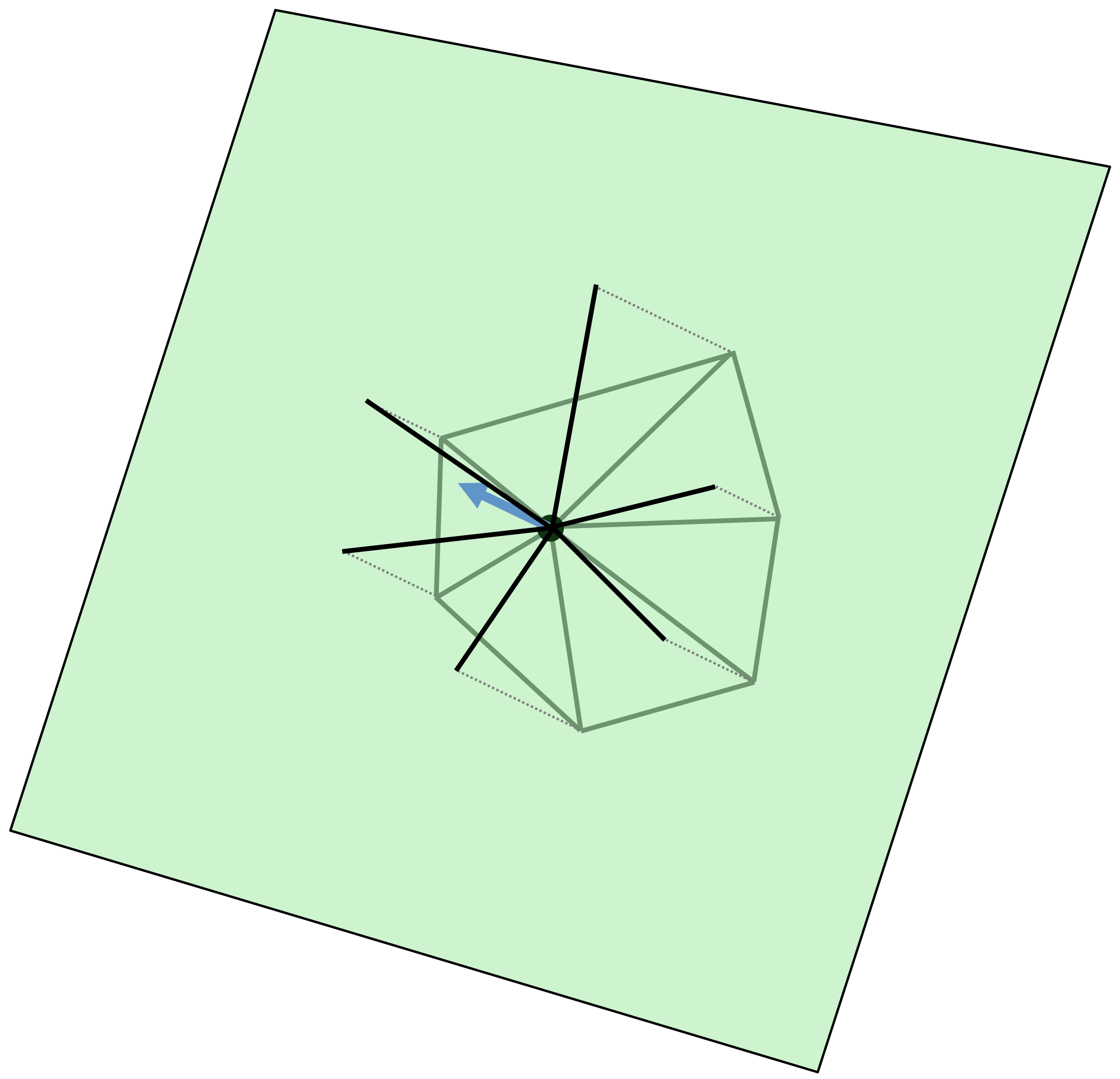


Shape operator via normal curvature

1. Project all edges (v_i, v_j) incident at v_i on the tangent plane to obtain tangent directions $\mathbf{w}_{ij}$

2. For each direction $\mathbf{w}_{ij}$ evaluate normal curvature κ_{ij} by local curve fitting

3. Average results from normal curvatures to compute the shape operator

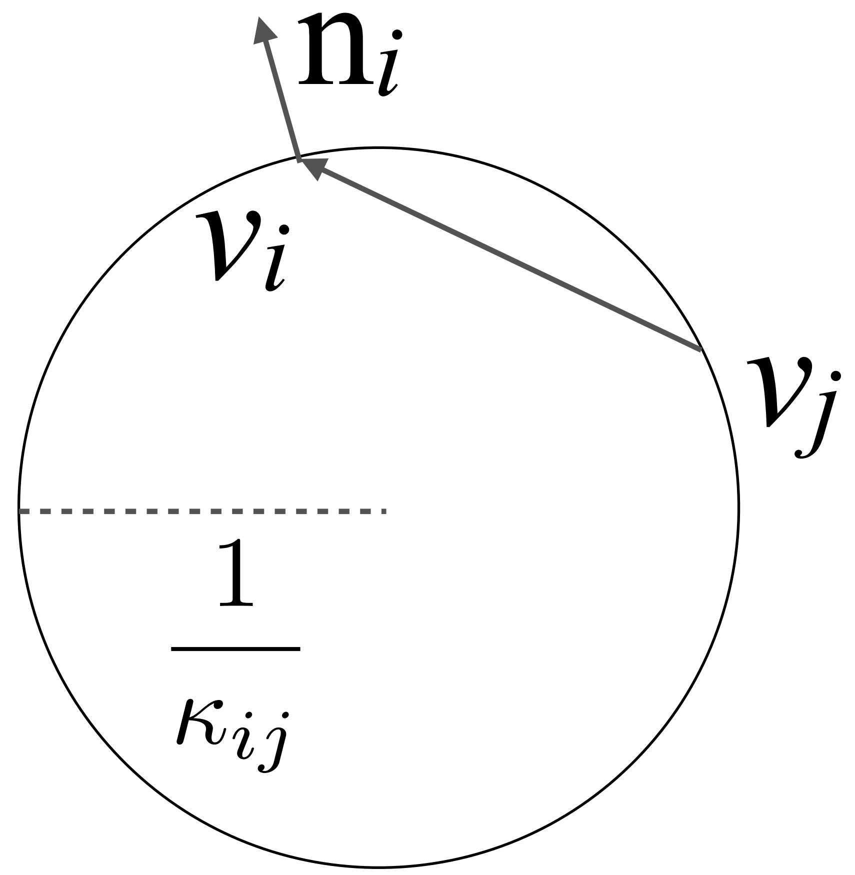


MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Springer-Verlag, Heidelberg, Germany, 35–57.

Shape operator via normal curvature

- Step 2: normal curvature at v_i along edge (v_i, v_j) can be estimated by fitting an osculating circle through v_i and v_j tangent to the plane defined by normal $\mathbf{n}_i$ and vector $\mathbf{w}_{ij}$:

$$\kappa_{ij} = 2\frac{(v_i - v_j) \cdot \mathbf{n}_i}{||v_i - v_j||^2}$$



Shape operator via normal curvature

- Step 3: once κ_{ij} has been computed for all v_j's neighbors of v_i, the second fundamental form can be estimated by least squares fitting of equations

$$\kappa_{ij} = \mathbf{t}_{ij}^T \begin{bmatrix} L & M \\ M & N \end{bmatrix} \mathbf{t}_{ij}$$

where $\mathbf{t}_{ij}$ denotes the normalized $\mathbf{w}_{ij}$ (unit-length projection to the tangent plane at v_i of vector v_j-v_i) expressed in a local coordinate frame $(\mathbf{u},\mathbf{v},\mathbf{n}_i)$ with origin at v_i

- Note: the first fundamental form w.r.t. $(\mathbf{u},\mathbf{v},\mathbf{n}_i)$ is the identity

Curvature tensor via discrete integration

- A more accurate estimation of the curvature tensor at v can be computed by averaging curvature tensors defined along edges of a neighborhood $N(v)$
- For a single edge $\mathbf{e}$ curvature is considered zero along $\mathbf{e}$, and maximum according to the dihedral edge $\beta(\mathbf{e})$ formed by the two incident faces across $\mathbf{e}$

$$\mathbf{C}(v) = \frac{1}{|N(v)|} \sum_{e \in N(v)} \beta(\mathbf{e}) \frac{||\mathbf{e} \cap N(v)||}{||\mathbf{e}||^2} \mathbf{e}\mathbf{e}^T$$

David Cohen-Steiner and Jean-Marie Morvan. 2003. Restricted Delaunay triangulations and normal cycle. In Proceedings of the nineteenth annual symposium on Computational geometry (SCG '03). Association for Computing Machinery, New York, NY, USA, 312–321.

Curvature tensor via discrete integration

- Intuition behind formula:
  - $\frac{\mathbf{e}\mathbf{e}^T}{\|\mathbf{e}\|^2}$ is a unit curvature tensor aligned with edge $\mathbf{e}$
  - $\beta(\mathbf{e})$ is the discrete curvature angle across $\mathbf{e}$ (remember that a curvature angle is a curvature times a length)
  - $\|\mathbf{e} \cap N(v)\|$ is the weight of contribution of edge $\mathbf{e}$
  - the summation is a discrete integral
  - $|N(v)|$ is the area of integration (to average)

$$\mathbf{C}(v) = \frac{1}{|N(v)|} \sum_{e \in N(v)} \beta(\mathbf{e}) \frac{||\mathbf{e} \cap N(v)||}{||\mathbf{e}||^2} \mathbf{e}\mathbf{e}^T$$

Discrete Gaussian curvature

- A discrete estimation of Gaussian curvature at a vertex v is given by the *angle defect* of its incident triangles

$$K(v) = \frac{1}{A(v)} \left(2\pi - \sum_{v_j \in N(v)} \theta_j \right)$$

where θ_j denotes the angle between vv_j and vv_{j+1}

- Analogous to the discrete curvature at a vertex of a polyline
  - angle defect is a quantity integrated over the discrete integration area

Discrete Gauss-Bonnet Theorem

- Total Gaussian curvature is fixed for a given topology

$$\int_{\mathcal{M}} K \, dA = \sum_i A_i K(\mathbf{v}_i) = \sum_i \left[2\pi - \sum_{j \in \mathcal{N}(i)} \theta_j \right] == 2\pi \chi(\mathcal{M})$$

Thank you