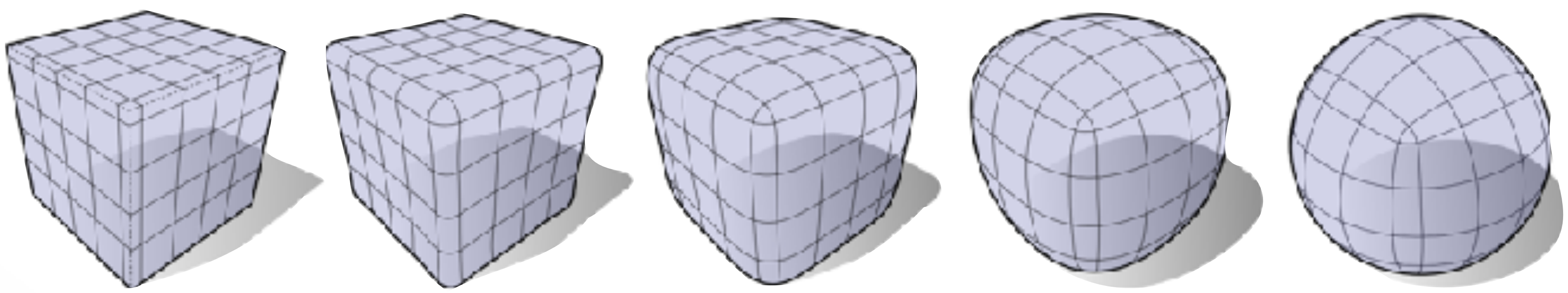
4.2 Discrete Differential Geometry



Hao Li

http://cs621.hao-li.com

Outline

- **Discrete Differential Operators**
- Discrete Curvatures
- Mesh Quality Measures

Differential Operators on Polygons

Differential Properties

- Surface is sufficiently differentiable
- Curvatures → 2nd derivatives

Differential Operators on Polygons

Differential Properties

- Surface is sufficiently differentiable
- Curvatures → 2nd derivatives

Polygonal Meshes

- Piecewise linear approximations of smooth surface
- Focus on Discrete Laplace Beltrami Operator
- Discrete differential properties defined over $\mathcal{N}(\mathbf{x})$

Local Averaging

Local Neighborhood $\mathcal{N}(\mathbf{x})$ of a point $\mathbf{x}$

- often coincides with mesh vertex v_i
- n-ring neighborhood $\mathcal{N}_n(v_i)$ or local geodesic ball

Local Averaging

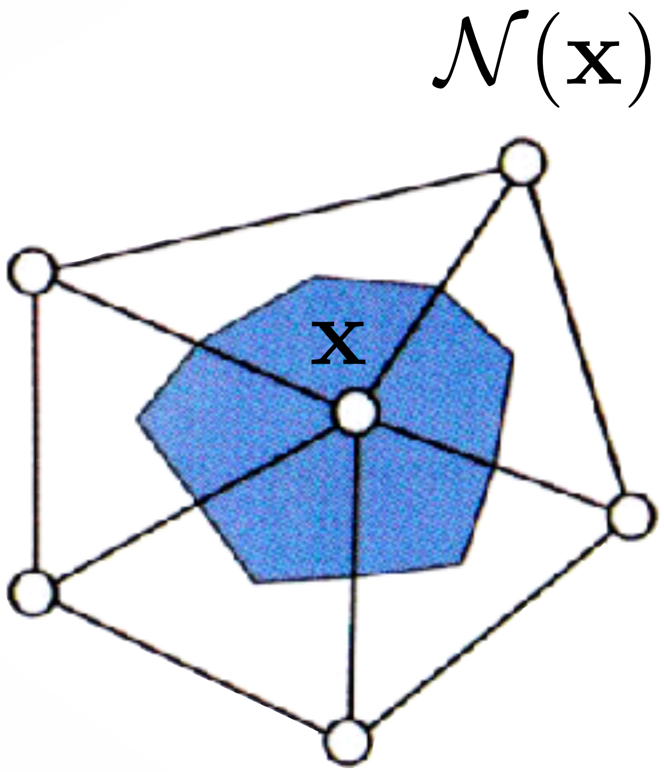
Local Neighborhood $\mathcal{N}(\mathbf{x})$ of a point $\mathbf{x}$

- often coincides with mesh vertex v_i
- n-ring neighborhood $\mathcal{N}_n(v_i)$ or local geodesic ball

Neighborhood size

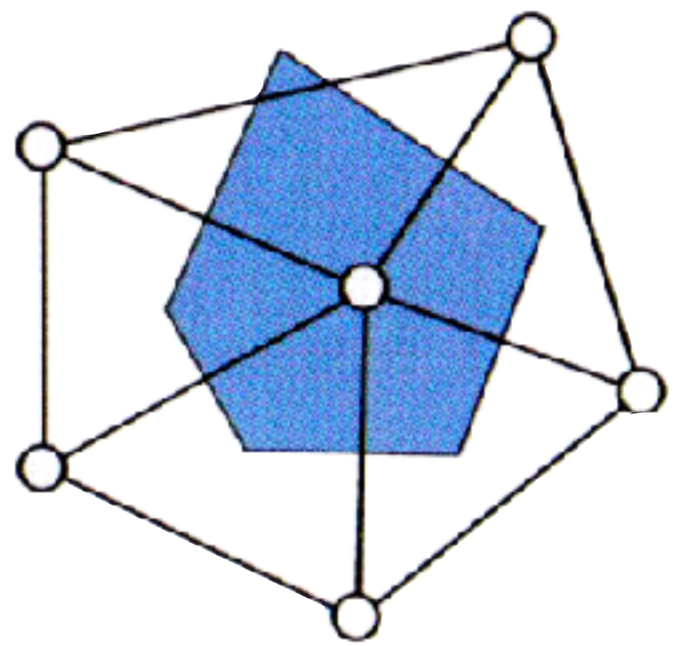
- Large: smoothing is introduced, stable to noise
- Small: fine scale variation, sensitive to noise

Local Averaging: 1-Ring



Barycentric cell

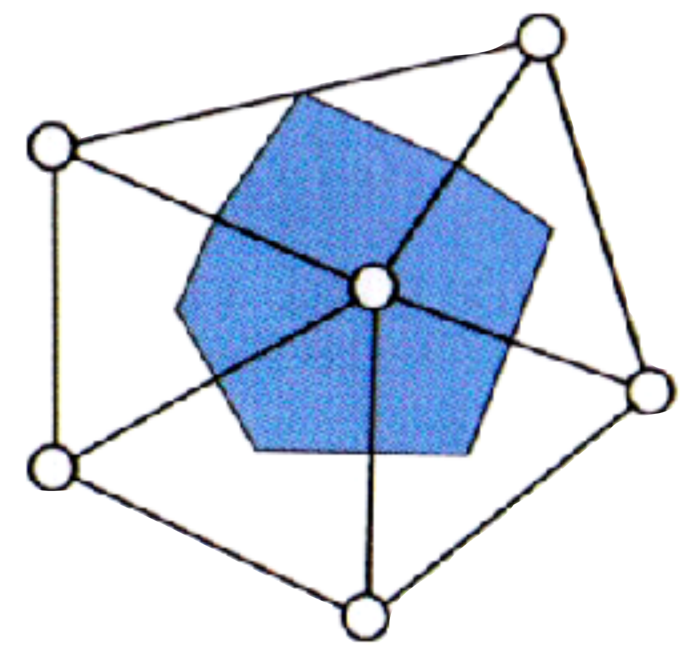
(barycenters/edgemidpoints)



Voronoi cell

(circumcenters)

tight error bound



Mixed Voronoi cell

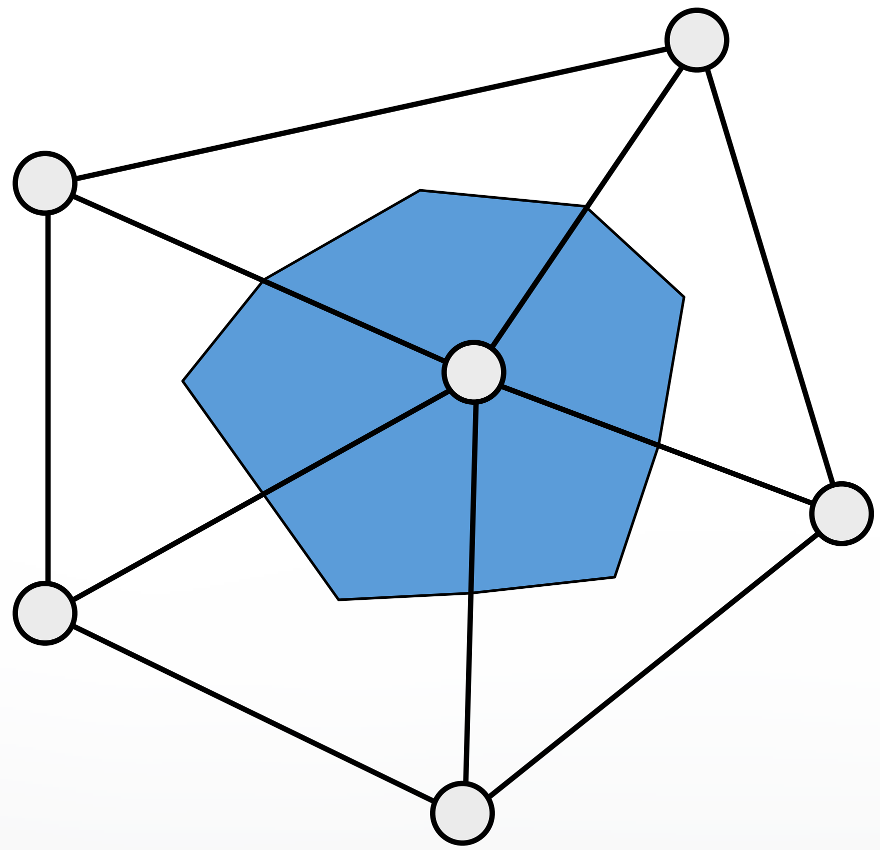
(circumcenters/midpoint)

better approximation

Barycentric Cells

Connect edge midpoints and triangle barycenters

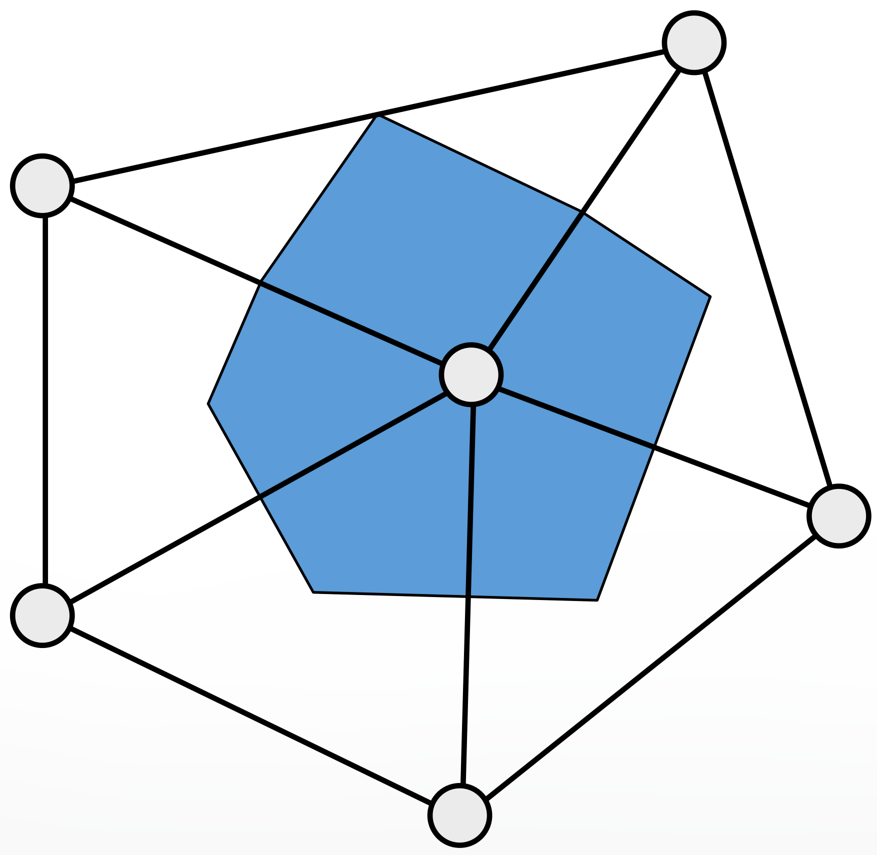
- Simple to compute
- Area is 1/3 o triangle areas
- Slightly wrong for obtuse triangles



Mixed Cells

Connect edge midpoints and

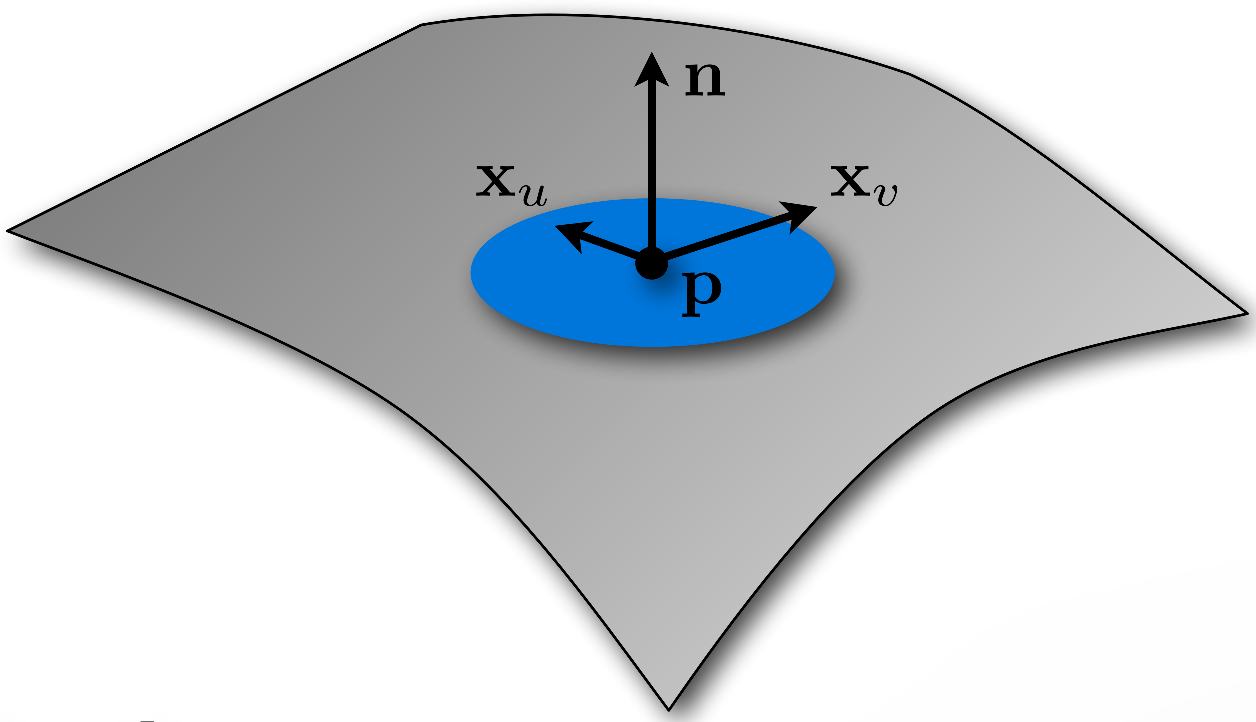
- Circumcenters for non-obtuse triangles
- Midpoint of opposite edge for obtuse triangles
- Better approximation, more complex to compute…



Normal Vectors

Continuous surface

$$\mathbf{x}(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}$$



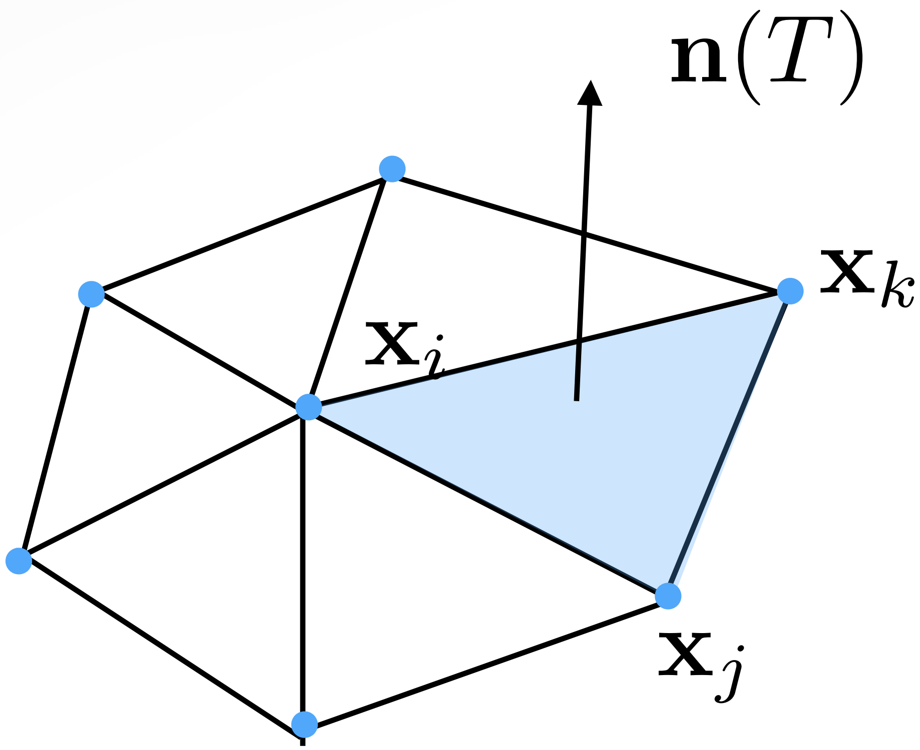
Normal vector

$$\mathbf{n} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\|\mathbf{x}_u \times \mathbf{x}_v\|}$$

Assume *regular* parameterization

$$\mathbf{x}_u \times \mathbf{x}_v \neq \mathbf{0}$$ normal exists

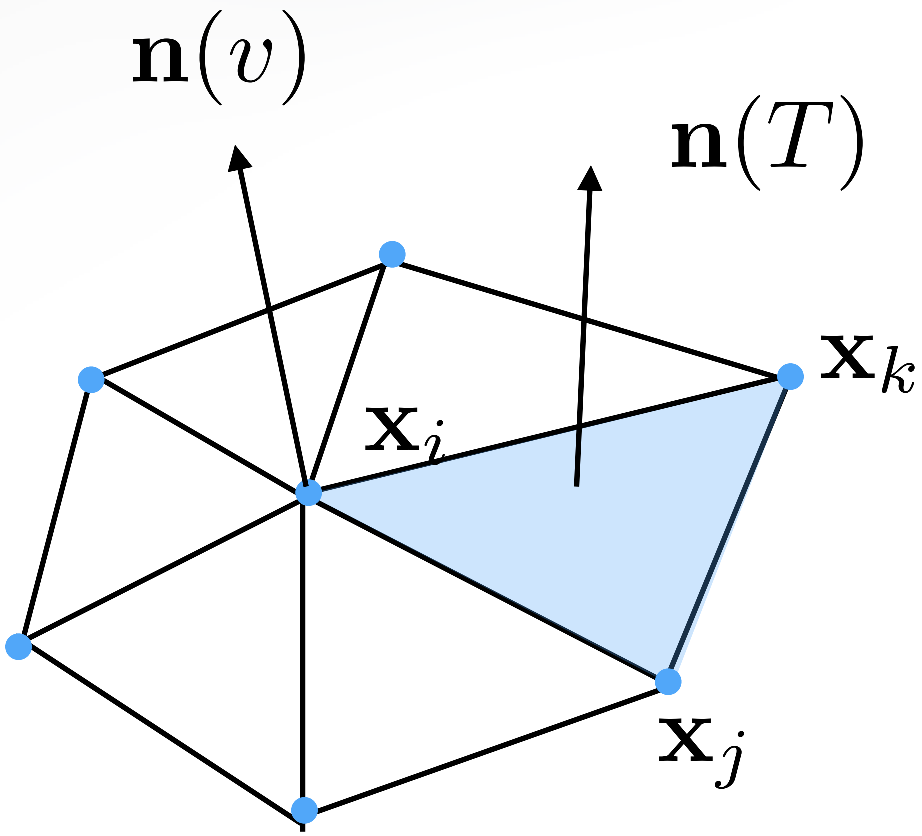
Discrete **Normal Vectors**



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

$$T = (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_j)$$

Discrete **Normal Vectors**

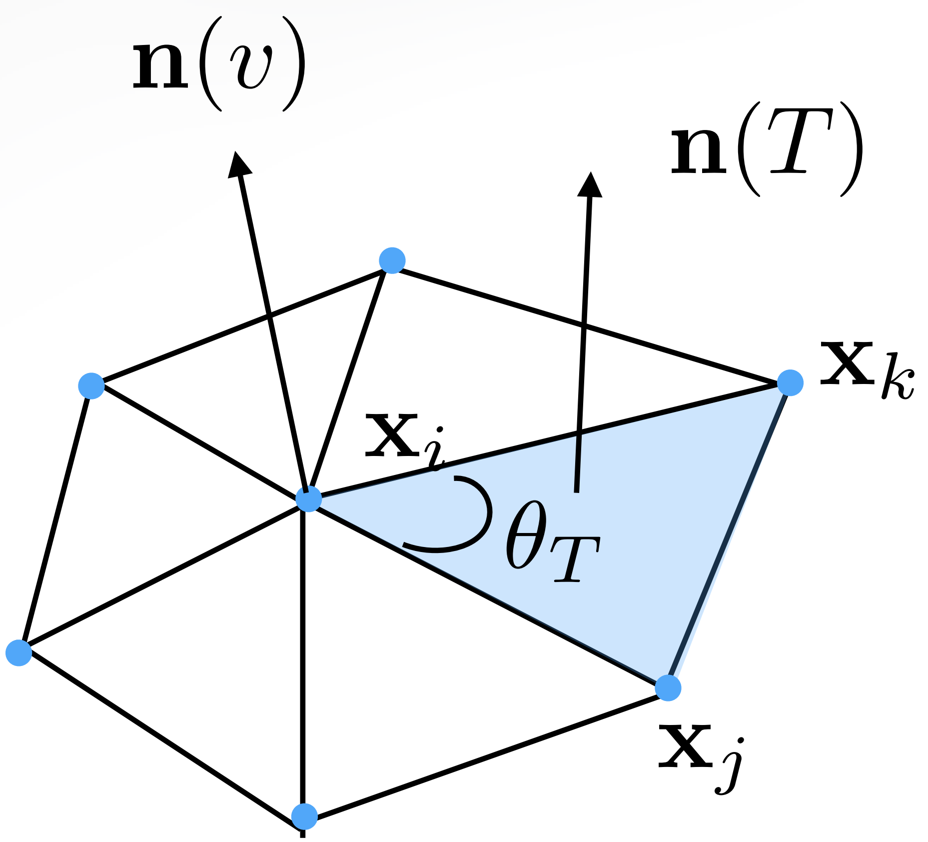


$$\mathbf{n}(T) = \frac{(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)}{\|(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)\|}$$

$$T = (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_j)$$

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

Discrete **Normal Vectors**



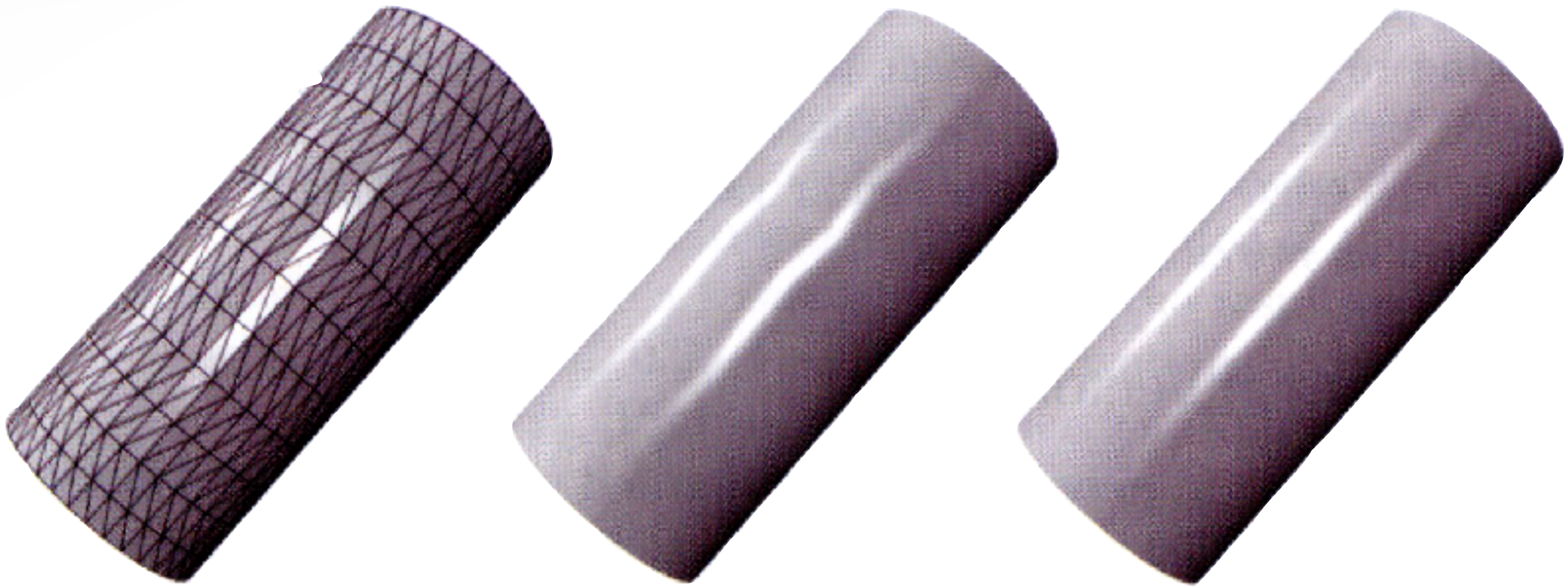
$$\mathbf{n}(T) = \frac{(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)}{\|(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)\|}$$

$$T = (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_j)$$

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

$\alpha_T = 1$ $\qquad$ $\alpha_T = |T|$ $\qquad$ $\alpha_T = \theta_T$

Discrete **Normal Vectors**



tessellated cylinder

$\alpha_T = 1$

$\alpha_T = |T|$

$\alpha_T = \theta_T$

Simple Curvature Discretization

mean curvature

Laplace-Beltrami $\Delta_{\mathcal{S}}\mathbf{x} = \mathrm{div}_{\mathcal{S}}\, \nabla_{\mathcal{S}}\mathbf{x} = -2H\mathbf{n}$

Simple Curvature Discretization

mean curvature

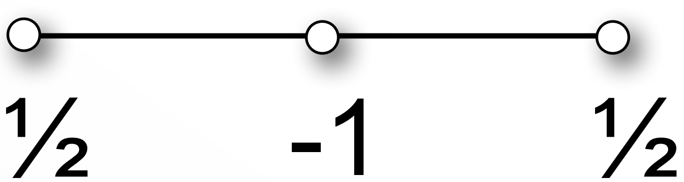
Laplace-Beltrami $$\Delta_{\mathcal{S}}\mathbf{x} = \operatorname{div}_{\mathcal{S}} \nabla_{\mathcal{S}}\mathbf{x} = -2H\mathbf{n}$$

How to discretize?

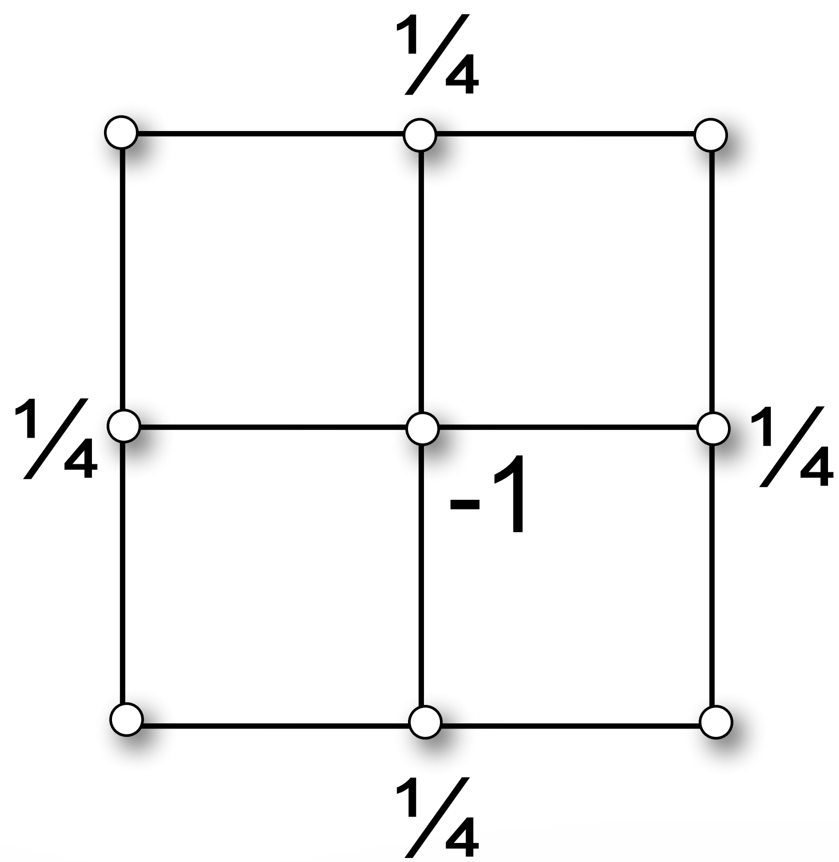
Laplace Operator on Meshes

Extend finite differences to meshes?

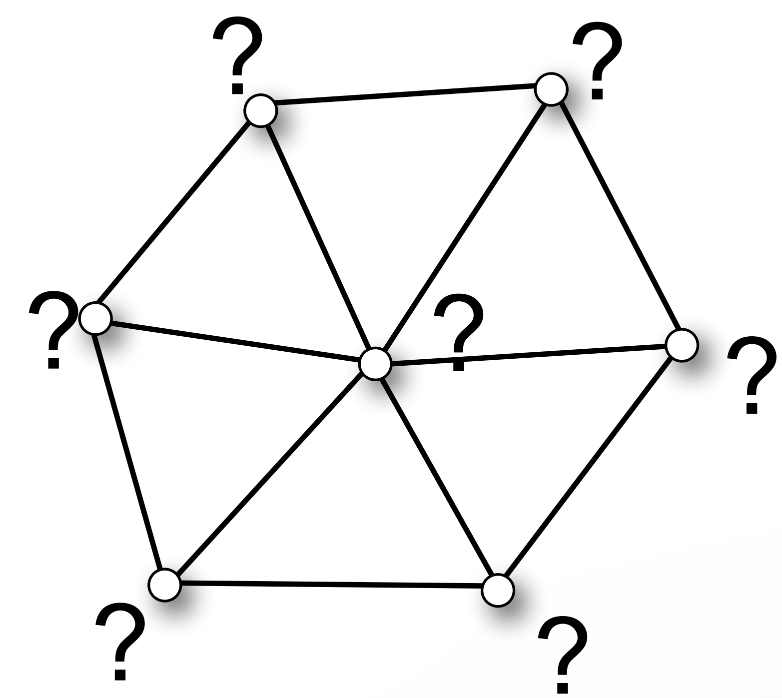
- What weights per vertex/edge?



1D grid



2D grid



2D/3D grid

Uniform Laplace

Uniform discretization

- What weights per vertex/edge?

Properties

- depends only on connectivity
- simple and efficient

Uniform Laplace

Uniform discretization

$$\Delta_{\text{uni}} \mathbf{x}_i \;:=\; \frac{1}{|\mathcal{N}_1(v_i)|} \sum_{v_j \in \mathcal{N}_1(v_i)} (\mathbf{x}_j - \mathbf{x}_i) \;\approx\; -2H\mathbf{n}$$

Properties

- depends only on connectivity
- simple and efficient
- bad approximation for irregular triangulations
    - can give non-zero H for planar meshes
    - tangential drift for mesh smoothing

Gradients

mean curvature

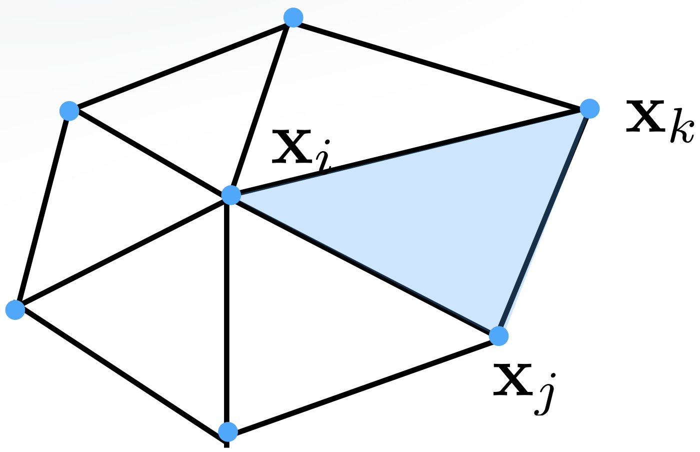
Laplace-Beltrami $\Delta_{\mathcal{S}} \mathbf{x} = \text{div}_{\mathcal{S}} \, \nabla_{\mathcal{S}} \mathbf{x} = -2H\mathbf{n}$

gradient operator

Discrete Gradient of a Function

- Defined on piecewise linear triangle
- Important for **parameterization** and **deformation**

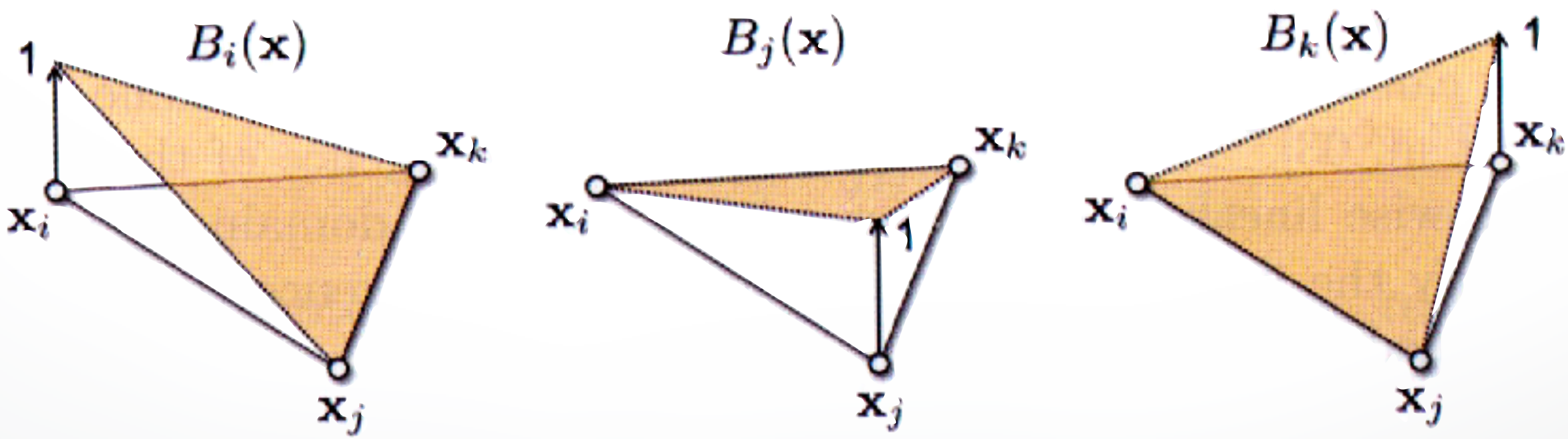
Gradients



triangle $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$

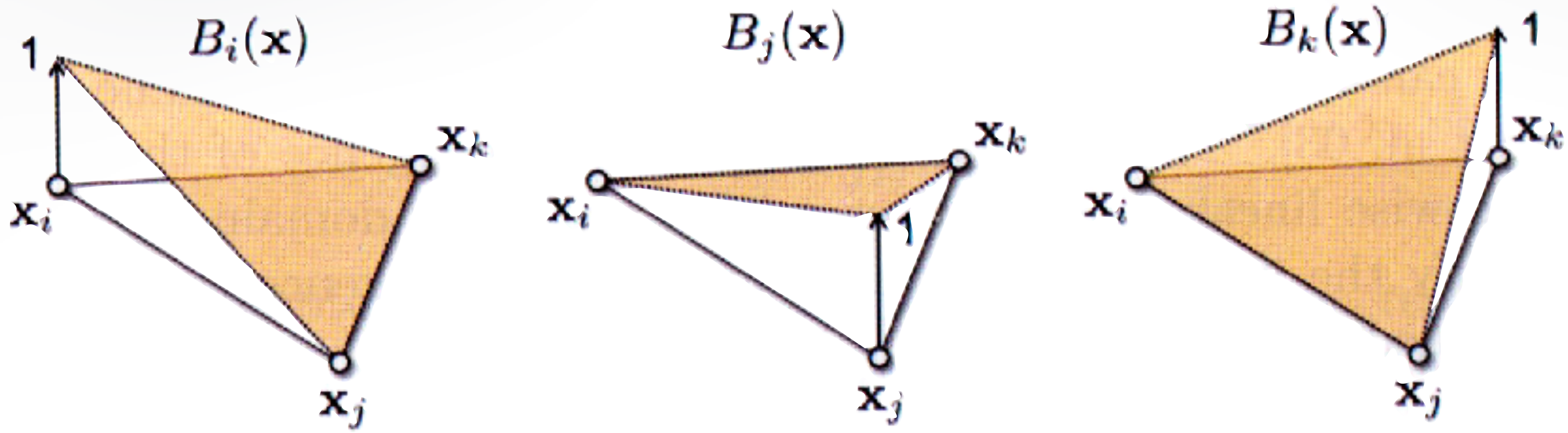
piecewise linear function $f(\mathbf{u}) = f_i B_i(\mathbf{u}) + f_j B_j(\mathbf{u}) + f_k B_k(\mathbf{u})$ $\quad \mathbf{u} = (u, v)$

$$f_i = f(\mathbf{x}_i)$$



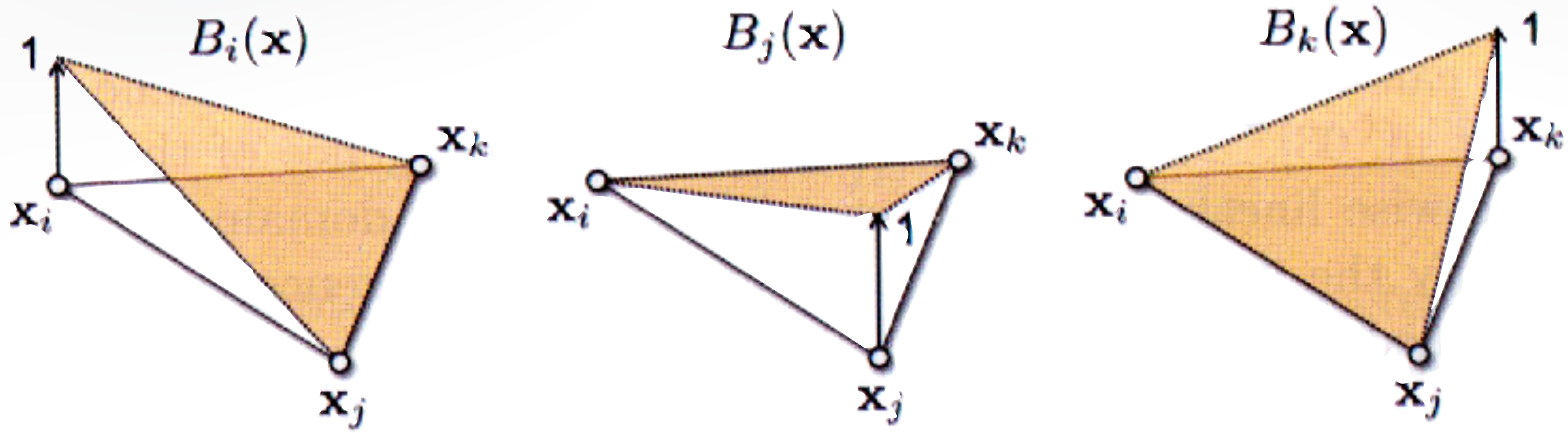
linear basis functions for barycentric interpolation on a triangle

Gradients



piecewise linear function $f(\mathbf{u}) = f_i B_i(\mathbf{u}) + f_j B_j(\mathbf{u}) + f_k B_k(\mathbf{u})$ $\mathbf{u} = (u, v)$

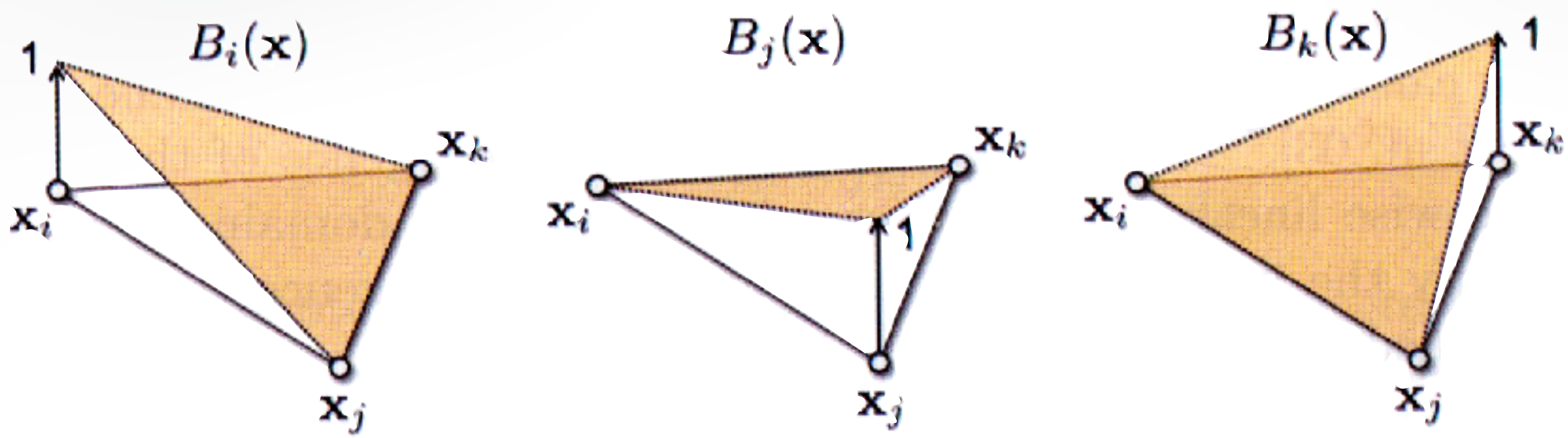
Gradients



piecewise linear function $f(\mathbf{u}) = f_i B_i(\mathbf{u}) + f_j B_j(\mathbf{u}) + f_k B_k(\mathbf{u})$ $\qquad \mathbf{u} = (u, v)$

gradient of linear function $\nabla f(\mathbf{u}) = f_i \nabla B_i(\mathbf{u}) + f_j \nabla B_j(\mathbf{u}) + f_k \nabla B_k(\mathbf{u})$

Gradients



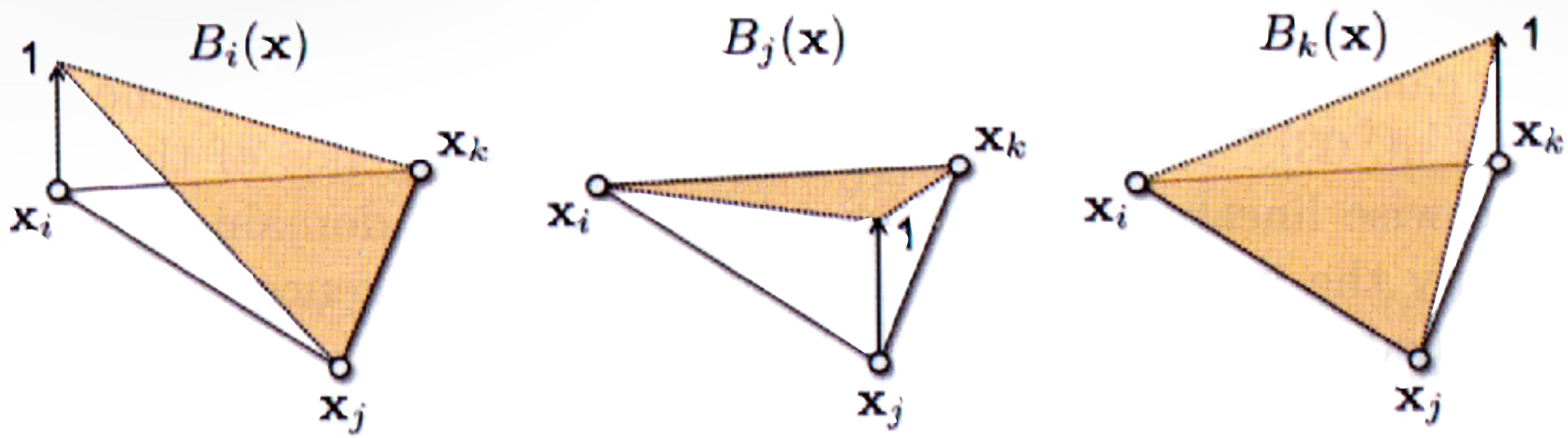
piecewise linear function $f(\mathbf{u}) = f_i B_i(\mathbf{u}) + f_j B_j(\mathbf{u}) + f_k B_k(\mathbf{u})$ $\mathbf{u} = (u, v)$

gradient of linear function $\nabla f(\mathbf{u}) = f_i \nabla B_i(\mathbf{u}) + f_j \nabla B_j(\mathbf{u}) + f_k \nabla B_k(\mathbf{u})$

partition of unity $B_i(\mathbf{u}) + B_j(\mathbf{u}) + B_k(\mathbf{u}) = 1$

gradients of basis $\nabla B_i(\mathbf{u}) + \nabla B_j(\mathbf{u}) + \nabla B_k(\mathbf{u}) = 0$

Gradients



piecewise linear function $f(\mathbf{u}) = f_i B_i(\mathbf{u}) + f_j B_j(\mathbf{u}) + f_k B_k(\mathbf{u})$ $\quad \mathbf{u} = (u, v)$

gradient of linear function $\nabla f(\mathbf{u}) = f_i \nabla B_i(\mathbf{u}) + f_j \nabla B_j(\mathbf{u}) + f_k \nabla B_k(\mathbf{u})$

partition of unity $B_i(\mathbf{u}) + B_j(\mathbf{u}) + B_k(\mathbf{u}) = 1$

gradients of basis $\nabla B_i(\mathbf{u}) + \nabla B_j(\mathbf{u}) + \nabla B_k(\mathbf{u}) = 0$

gradient of linear function $\nabla f(\mathbf{u}) = (f_j - f_i)\nabla B_j(\mathbf{u}) + (f_k - f_i)\nabla B_k(\mathbf{u})$

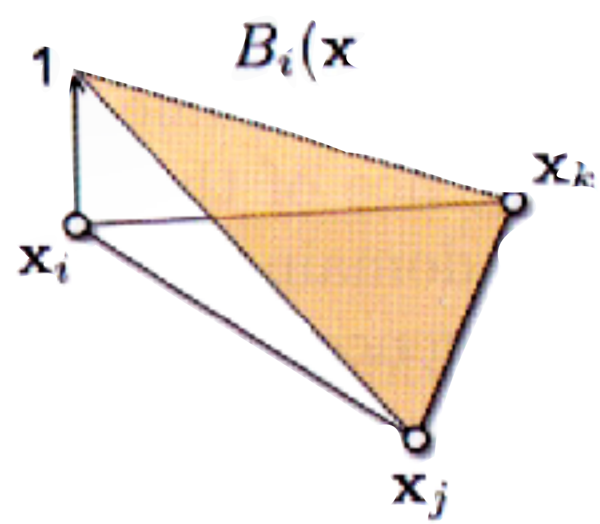
Gradients

gradient of linear function $\nabla f(\mathbf{u}) = (f_j - f_i)\nabla B_j(\mathbf{u}) + (f_k - f_i)\nabla B_k(\mathbf{u})$

Gradients

gradient of linear function $\nabla f(\mathbf{u}) = (f_j - f_i)\nabla B_j(\mathbf{u}) + (f_k - f_i)\nabla B_k(\mathbf{u})$

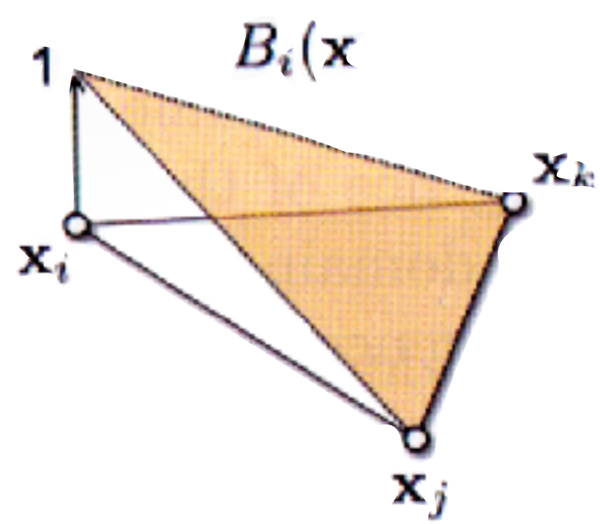
with appropriate normalization: $\nabla B_i(\mathbf{u}) = \dfrac{(\mathbf{x}_k - \mathbf{x}_j)^\perp}{2\, A_T}$



Gradients

gradient of linear function $\nabla f(\mathbf{u}) = (f_j - f_i)\nabla B_j(\mathbf{u}) + (f_k - f_i)\nabla B_k(\mathbf{u})$

with appropriate normalization: $\nabla B_i(\mathbf{u}) = \frac{(\mathbf{x}_k - \mathbf{x}_j)^\perp}{2\,A_T}$



$$\nabla f(\mathbf{u}) = (f_j - f_i)\frac{(\mathbf{x}_i - \mathbf{x}_k)^\perp}{2A_T} + (f_k - f_i)\frac{(\mathbf{x}_j - \mathbf{x}_i)^\perp}{2A_T}$$

$$f_i = f(\mathbf{x}_i)$$

discrete gradient of a piecewiese linear function within *T*

Discrete Laplace-Beltrami

gradient operator

mean curvature

Laplace-Beltrami $\Delta_{\mathcal{S}}\mathbf{x} = \operatorname{div}_{\mathcal{S}} \nabla_{\mathcal{S}}\mathbf{x} = -2H\mathbf{n}$

Discrete Laplace-Beltrami

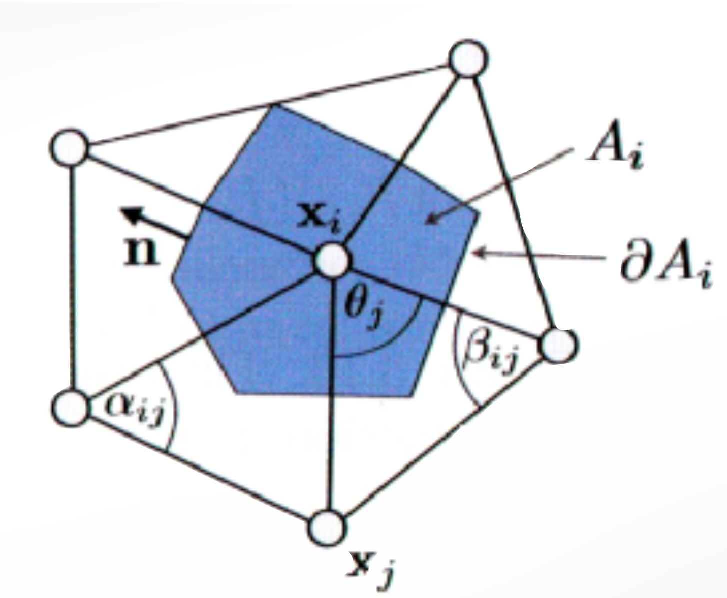
gradient operator

mean curvature

Laplace-Beltrami $\Delta_{\mathcal{S}} \mathbf{x} = \mathrm{div}_{\mathcal{S}} \nabla_{\mathcal{S}} \mathbf{x} = -2H\mathbf{n}$

divergence theorem

$$\int_{A_i} \mathrm{div}\, \mathbf{F}(\mathbf{u})\, \mathrm{d}A = \int_{\partial A_i} \mathbf{F}(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u})\, \mathrm{d}s$$



Discrete Laplace-Beltrami

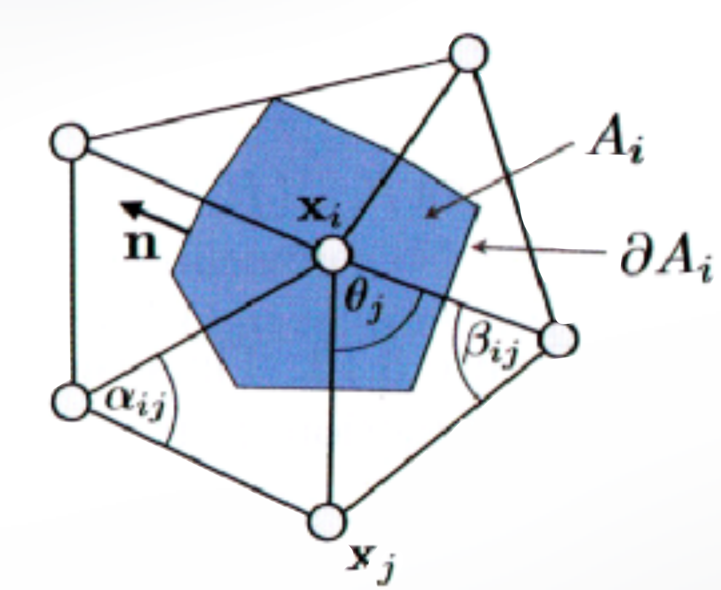
gradient operator

mean curvature

Laplace-Beltrami $\Delta_{\mathcal{S}} \mathbf{x} = \mathrm{div}_{\mathcal{S}} \nabla_{\mathcal{S}} \mathbf{x} = -2H\mathbf{n}$

divergence theorem

$$\int_{A_i} \mathrm{div}\, \mathbf{F}(\mathbf{u})\, \mathrm{d}A = \int_{\partial A_i} \mathbf{F}(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u})\, \mathrm{d}s$$



vector-valued function $\mathbf{F}$

local averaging domain $A_i = A(v_i)$

boundary ∂A_i

Discrete Laplace-Beltrami

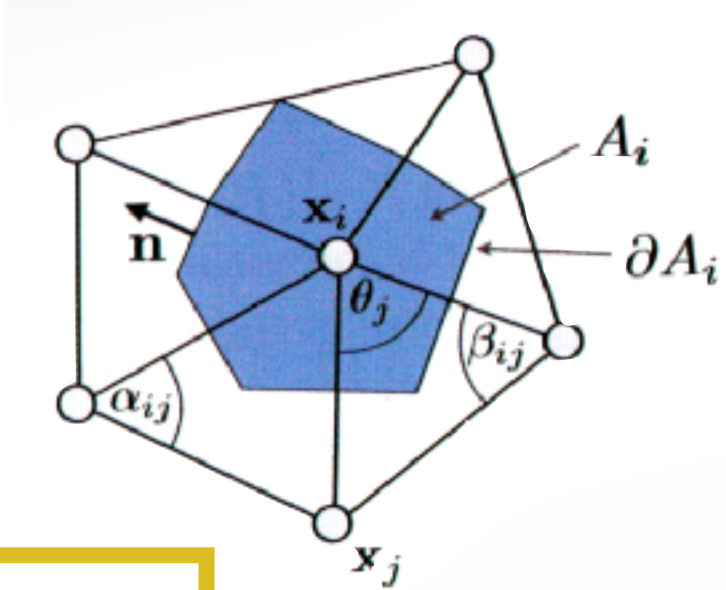
gradient operator

mean curvature

Laplace-Beltrami $\Delta_{\mathcal{S}}\mathbf{x} = \mathrm{div}_{\mathcal{S}}\, \nabla_{\mathcal{S}}\mathbf{x} = -2H\mathbf{n}$

divergence theorem

$$\int_{A_i} \mathrm{div}\,\mathbf{F}(\mathbf{u})\,\mathrm{d}A = \int_{\partial A_i} \mathbf{F}(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u})\,\mathrm{d}s$$



$$\int_{A_i} \Delta f(\mathbf{u})\,\mathrm{d}A = \int_{A_i} \mathrm{div}\nabla f(\mathbf{u})\,\mathrm{d}A = \int_{\partial A_i} \nabla f(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u})\,\mathrm{d}s$$

Discrete Laplace-Beltrami

average Laplace-Beltrami

$$\int_{A_i} \Delta f(\mathbf{u})\, \mathrm{d}A = \int_{A_i} \mathrm{div} \nabla f(\mathbf{u})\, \mathrm{d}A = \int_{\partial A_i} \nabla f(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u})\, \mathrm{d}s$$

Discrete Laplace-Beltrami

average Laplace-Beltrami

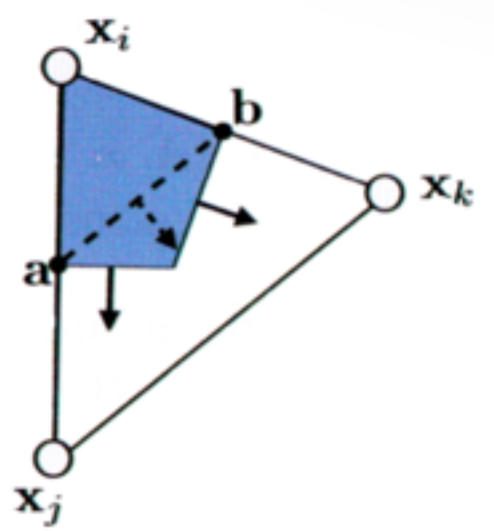
$$\int_{A_i} \Delta f(\mathbf{u})\, \mathrm{d}A = \int_{A_i} \mathrm{div} \nabla f(\mathbf{u})\, \mathrm{d}A = \int_{\partial A_i} \nabla f(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u})\, \mathrm{d}s$$

gradient is constant and local Voronoi passes through a,b:

$$\int_{\partial A_i \cap T} \nabla f(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u}) \mathrm{d}s = \nabla f(\mathbf{u}) \cdot (\mathbf{a} - \mathbf{b})^{\perp}$$

over triangle

$$= \frac{1}{2} \nabla f(\mathbf{u}) \cdot (\mathbf{x}_j - \mathbf{x}_k)^{\perp}$$



Discrete Laplace-Beltrami

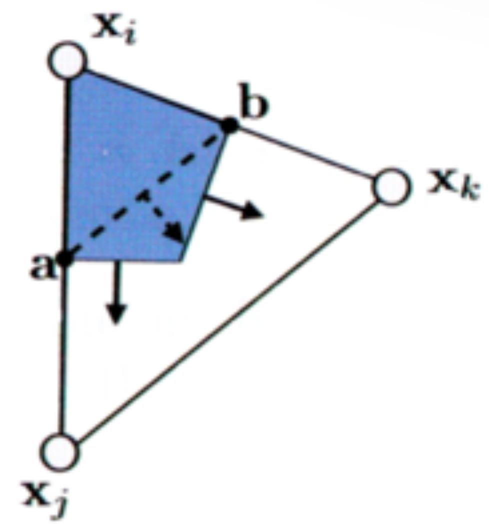
average Laplace-Beltrami

$$\int_{A_i} \Delta f(\mathbf{u})\, dA = \int_{A_i} \operatorname{div}\nabla f(\mathbf{u})\, dA = \int_{\partial A_i} \nabla f(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u})\, ds$$

gradient is constant and local Voronoi passes through a,b:

$$\int_{\partial A_i \cap T} \nabla f(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u}) ds = \nabla f(\mathbf{u}) \cdot (\mathbf{a} - \mathbf{b})^{\perp} = \frac{1}{2} \nabla f(\mathbf{u}) \cdot (\mathbf{x}_j - \mathbf{x}_k)^{\perp}$$

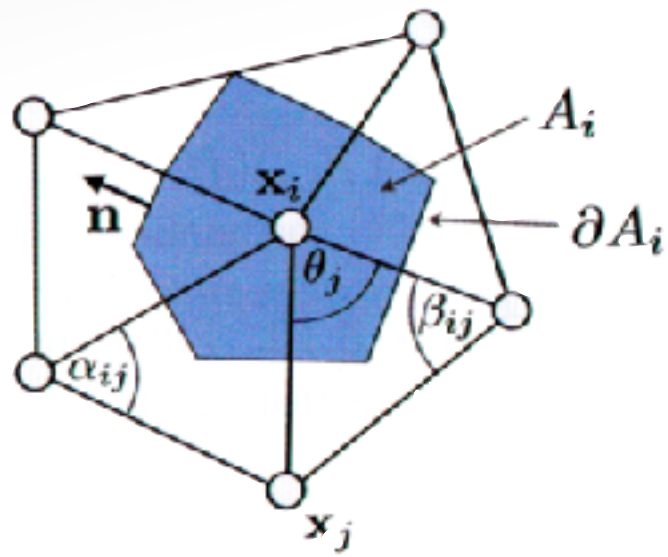
over triangle

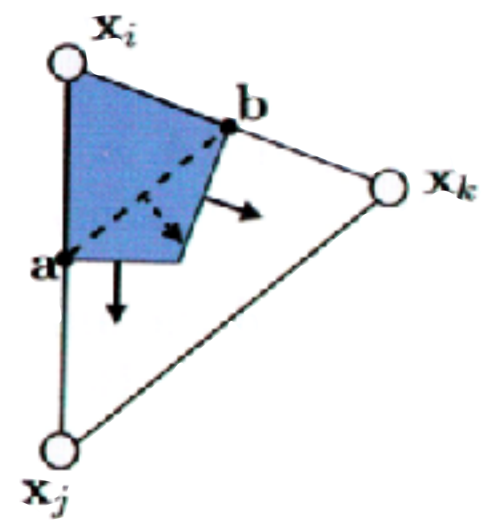


discrete gradient

$$\nabla f(\mathbf{u}) = (f_j - f_i)\frac{(\mathbf{x}_i - \mathbf{x}_k)^{\perp}}{2A_T} + (f_k - f_i)\frac{(\mathbf{x}_j - \mathbf{x}_i)^{\perp}}{2A_T}$$

Discrete Laplace-Beltrami

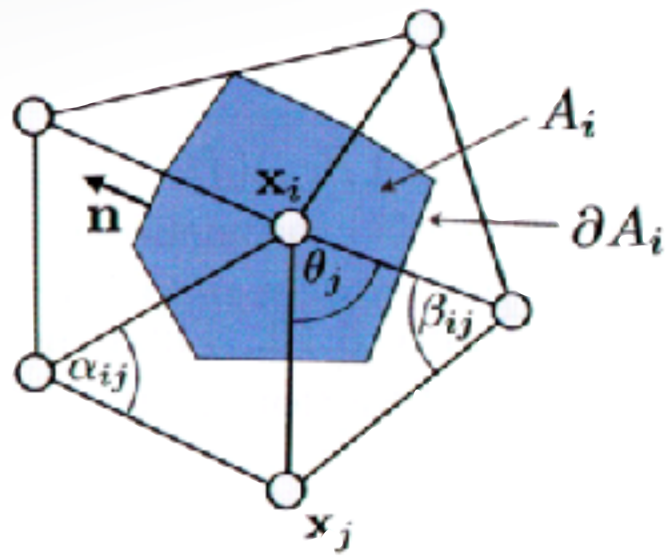


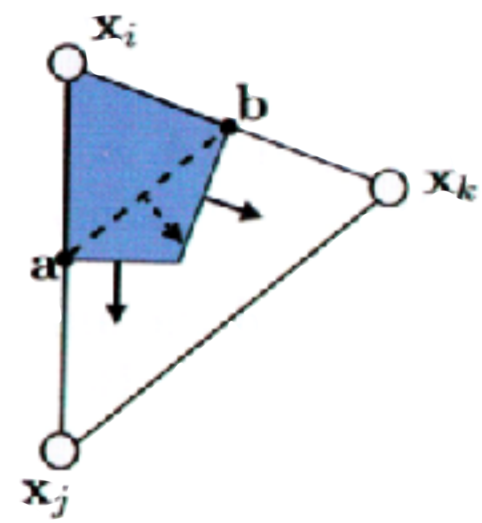


average Laplace-Beltrami within a triangle

$$\int_{\partial A_i \cap T} \nabla f(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u}) \mathrm{d}s = (f_j - f_i)\frac{(\mathbf{x}_i - \mathbf{x}_k)^{\perp} \cdot (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{4A_T} + (f_k - f_i)\frac{(\mathbf{x}_j - \mathbf{x}_i)^{\perp} \cdot (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{4A_T}$$

Discrete Laplace-Beltrami



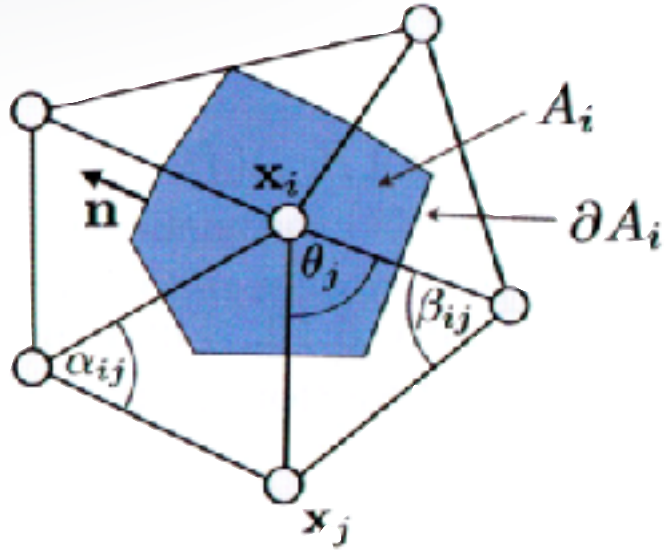


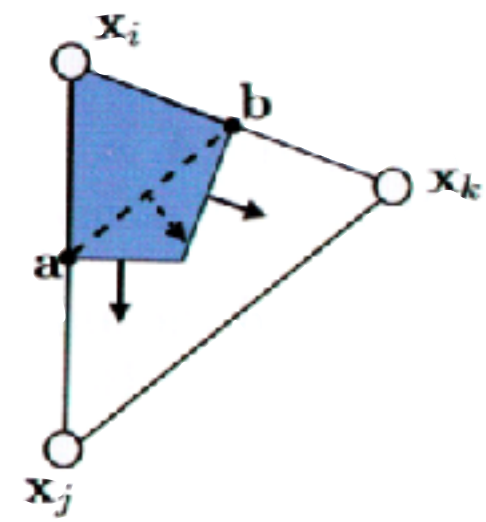
average Laplace-Beltrami within a triangle

$$\int_{\partial A_i \cap T} \nabla f(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u}) ds = (f_j - f_i) \frac{(\mathbf{x}_i - \mathbf{x}_k)^{\perp} \cdot (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{4A_T} + (f_k - f_i) \frac{(\mathbf{x}_j - \mathbf{x}_i)^{\perp} \cdot (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{4A_T}$$

$$\int_{\partial A_i \cap T} \nabla f(\mathbf{u}) \cdot \mathbf{n}(\mathbf{u}) ds = \frac{1}{2} \left(\cot \gamma_k (f_j - f_i) + \cot \gamma_j (f_k - f_i) \right)$$

Discrete Laplace-Beltrami

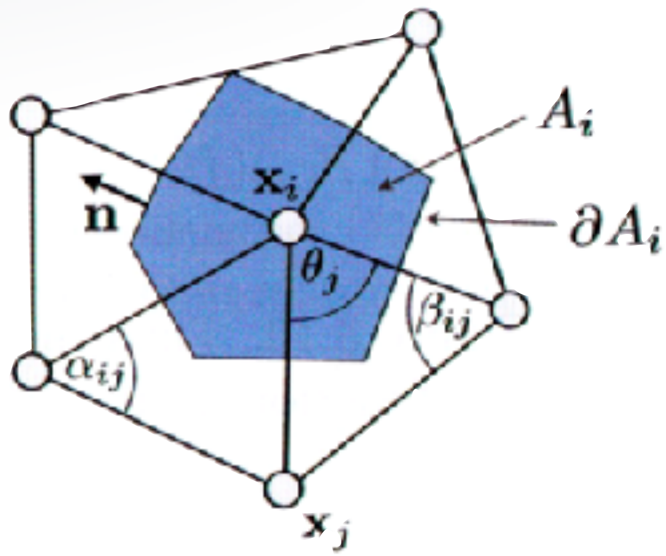


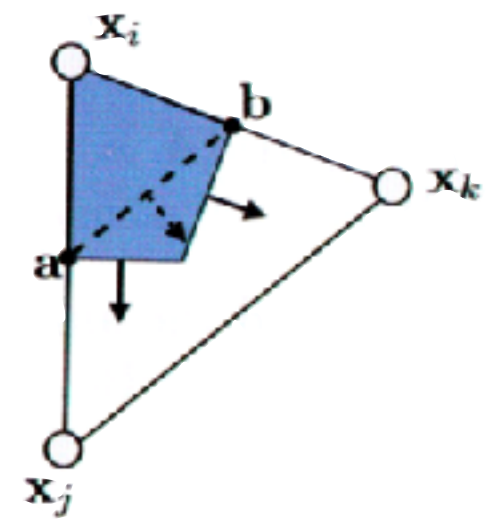


average Laplace-Beltrami over averaging region

$$\int_{A_i} \Delta f(\mathbf{u}) \mathrm{d}A = \frac{1}{2} \sum_{v_j \in \mathcal{N}_1(v_i)} (\cot \alpha_{i,j} + \cot \beta_{i,j})(f_j - f_i)$$

Discrete Laplace-Beltrami





average Laplace-Beltrami over averaging region

$$\int_{A_i} \Delta f(\mathbf{u}) \mathrm{d}A = \frac{1}{2} \sum_{v_j \in \mathcal{N}_1(v_i)} (\cot \alpha_{i,j} + \cot \beta_{i,j})(f_j - f_i)$$

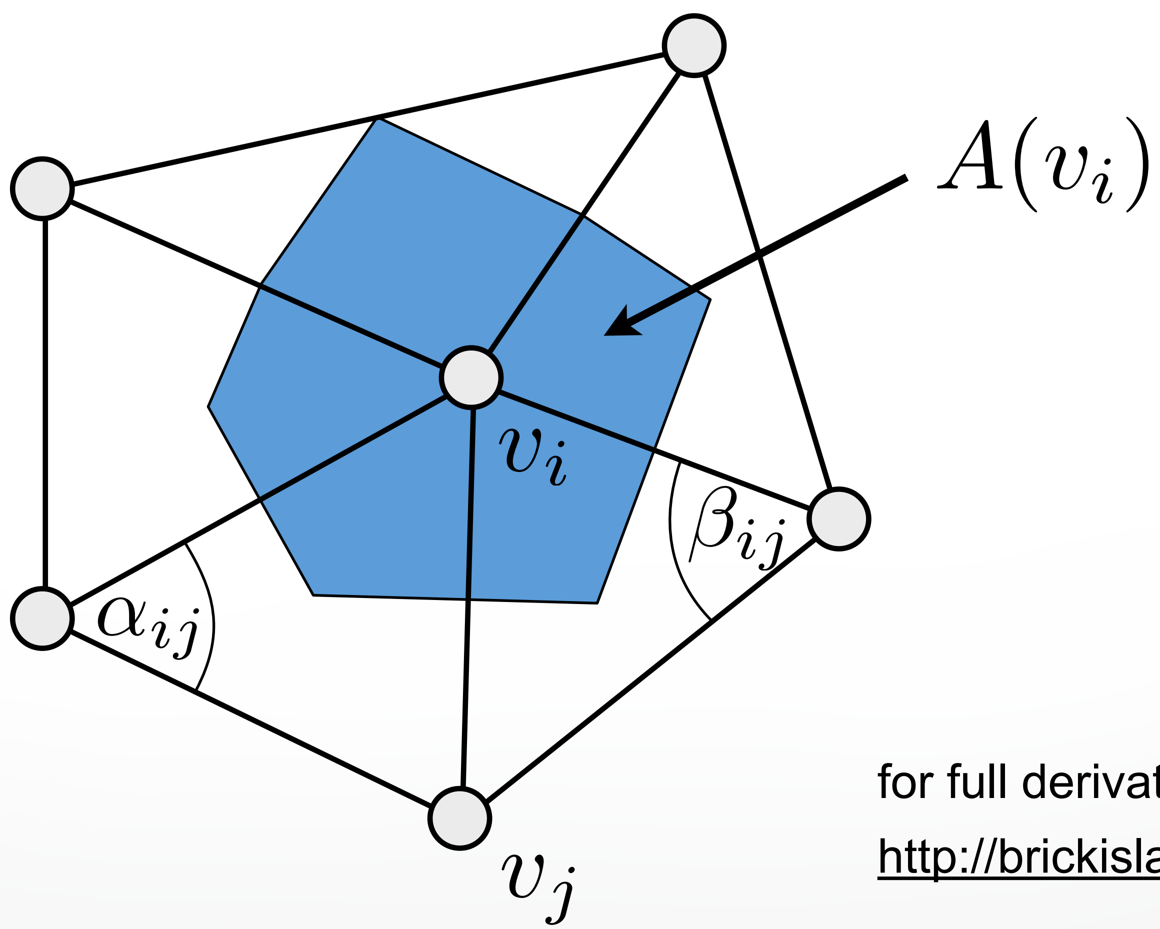
discrete Laplace-Beltrami

$$\Delta f(v_i) := \frac{1}{2A_i} \sum_{v_j \in \mathcal{N}_1(v_i)} (\cot \alpha_{i,j} + \cot \beta_{i,j})(f_j - f_i)$$

Discrete Laplace-Beltrami

Cotangent discretization

$$\Delta_{\mathcal{S}} f(v_i) \;:=\; \frac{1}{2A(v_i)} \sum_{v_j \in \mathcal{N}_1(v_i)} \left(\cot \alpha_{ij} + \cot \beta_{ij}\right)\left(f(v_j) - f(v_i)\right)$$



for full derivation, check out:
http://brickisland.net/cs177/

Discrete Laplace-Beltrami

Cotangent discretization

$$\Delta_{\mathcal{S}} f(v) := \frac{1}{2A(v)} \sum_{v_i \in \mathcal{N}_1(v)} (\cot \alpha_i + \cot \beta_i) \left(f(v_i) - f(v) \right)$$

Problems

- weights can become negative
- depends on triangulation

Still the most widely used discretization

Outline

- Discrete Differential Operators
- **Discrete Curvatures**
- Mesh Quality Measures

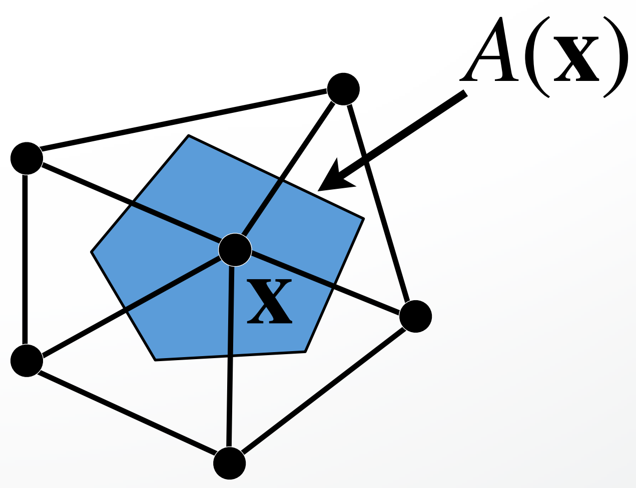
Discrete Curvatures

How to discretize curvature on a mesh?

- Zero curvature within triangles
- Infinite curvature at edges / vertices
- Point-wise definition doesn't make sense

Approximate differential properties at point $\mathbf{x}$ as average over local neighborhood $A(\mathbf{x})$

- $\mathbf{x}$ is a mesh vertex
- $A(\mathbf{x})$ within one-ring neighborhood



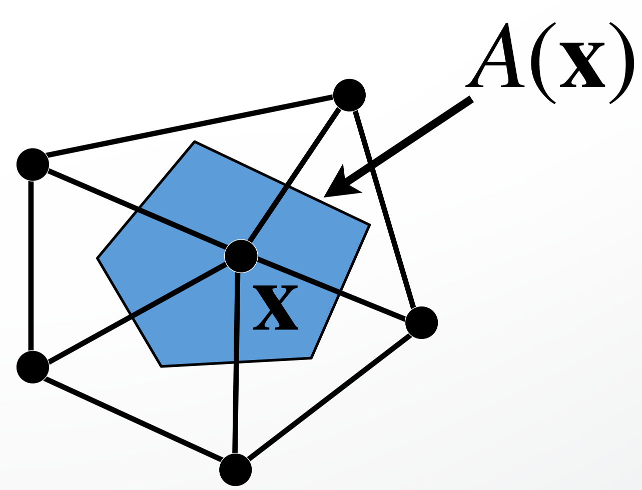
Discrete Curvatures

How to discretize curvature on a mesh?

- Zero curvature within triangles
- Infinite curvature at edges / vertices
- Point-wise definition doesn't make sense

Approximate differential properties at point $\mathbf{x}$ as average over local neighborhood $A(\mathbf{x})$

$$K(v) \approx \frac{1}{A(v)} \int_{A(v)} K(\mathbf{x}) \, \mathrm{d}A$$



Discrete Curvatures

Which curvatures to discretize?

- Discretize Laplace-Beltrami operator
- Laplace-Beltrami gives us mean curvature H
- Discretize Gaussian curvature K
- From H and K we can compute κ_1 and κ_2

mean curvature

Laplace-Beltrami $\Delta_{\mathcal{S}} \mathbf{x} = \mathrm{div}_{\mathcal{S}} \nabla_{\mathcal{S}} \mathbf{x} = -2H\mathbf{n}$

Discrete Gaussian Curvature

Gauss-Bonnet

$$\int K = 2\pi\chi \qquad\qquad \chi = 2 - 2g$$

Discrete Gauss Curvature

$$K = (2\pi - \sum_j \theta_j)/A$$

Verify via Euler-Poincaré

$$V - E + F = 2(1 - g)$$

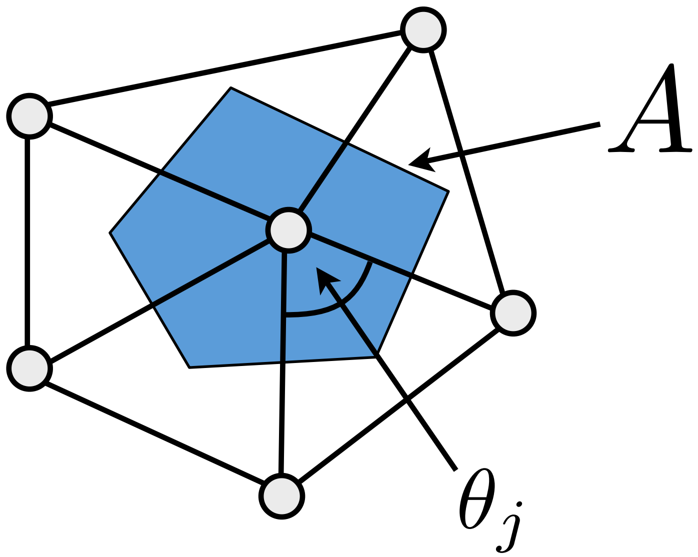
Discrete Curvatures

Mean curvature (absolute value)

$$H = \frac{1}{2} \|\Delta_{\mathcal{S}} \mathbf{x}\|$$

Gaussian curvature

$$K = (2\pi - \sum_j \theta_j)/A$$



Principal curvatures

$$\kappa_1 = H + \sqrt{H^2 - K} \qquad \kappa_2 = H - \sqrt{H^2 - K}$$

Outline

- Discrete Differential Operators
- Discrete Curvatures
- **Mesh Quality Measures**

Mesh Quality

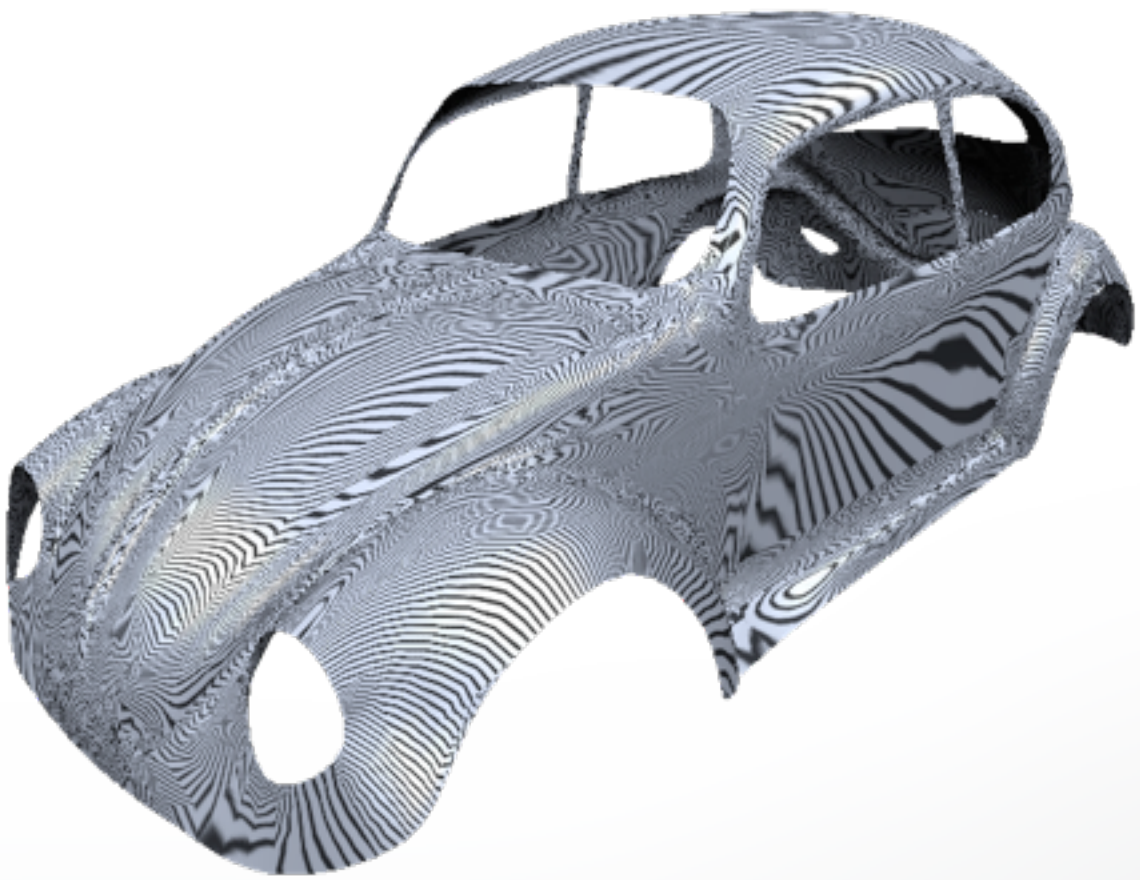
Visual inspection of “sensitive” attributes

- Specular shading

Mesh Quality

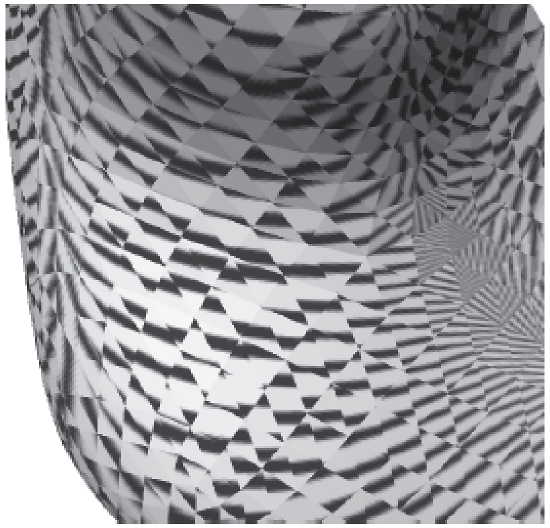
Visual inspection of "sensitive" attributes

- Specular shading
- Reflection lines

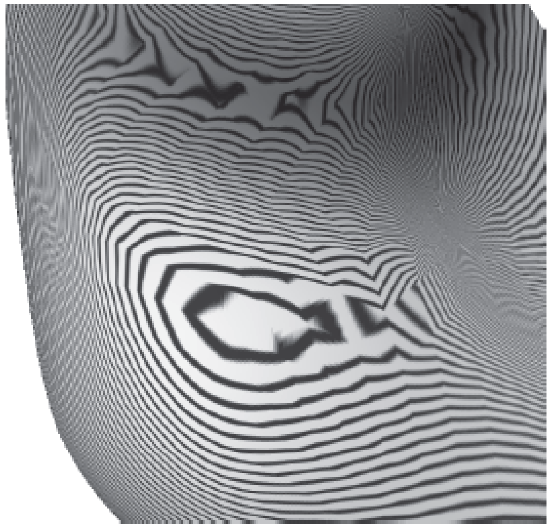


Visual inspection of “sensitive” attributes

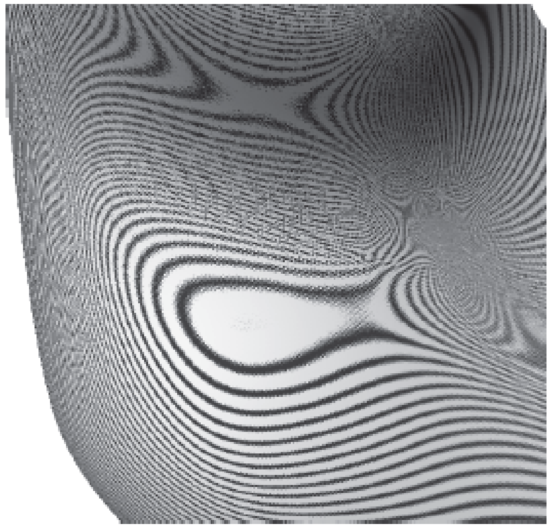
- Specular shading
- Reflection lines
  - differentiability one order lower than surface
  - can be efficiently computed using GPU



C^0



C^1

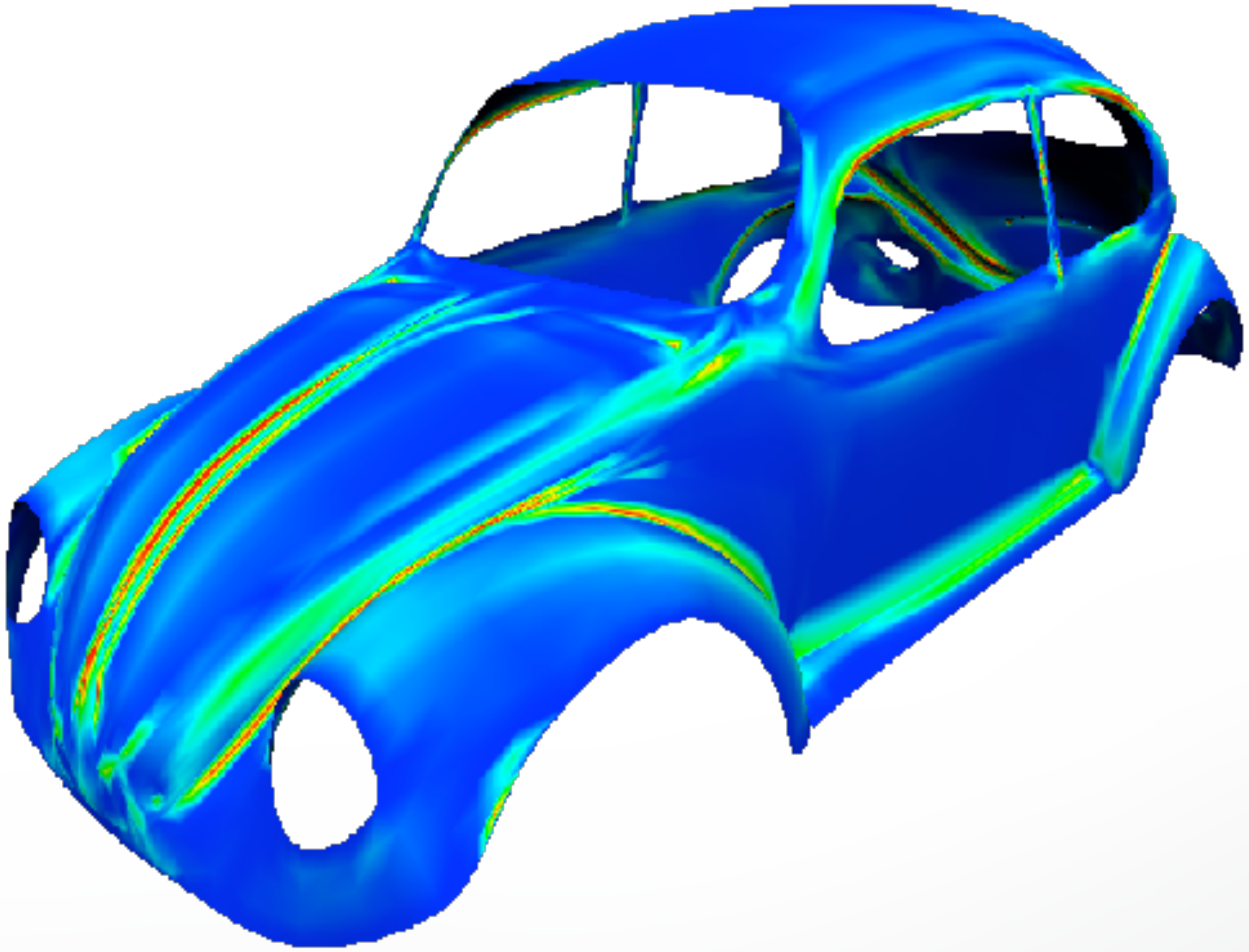


C^2

Mesh Quality

Visual inspection of “sensitive” attributes

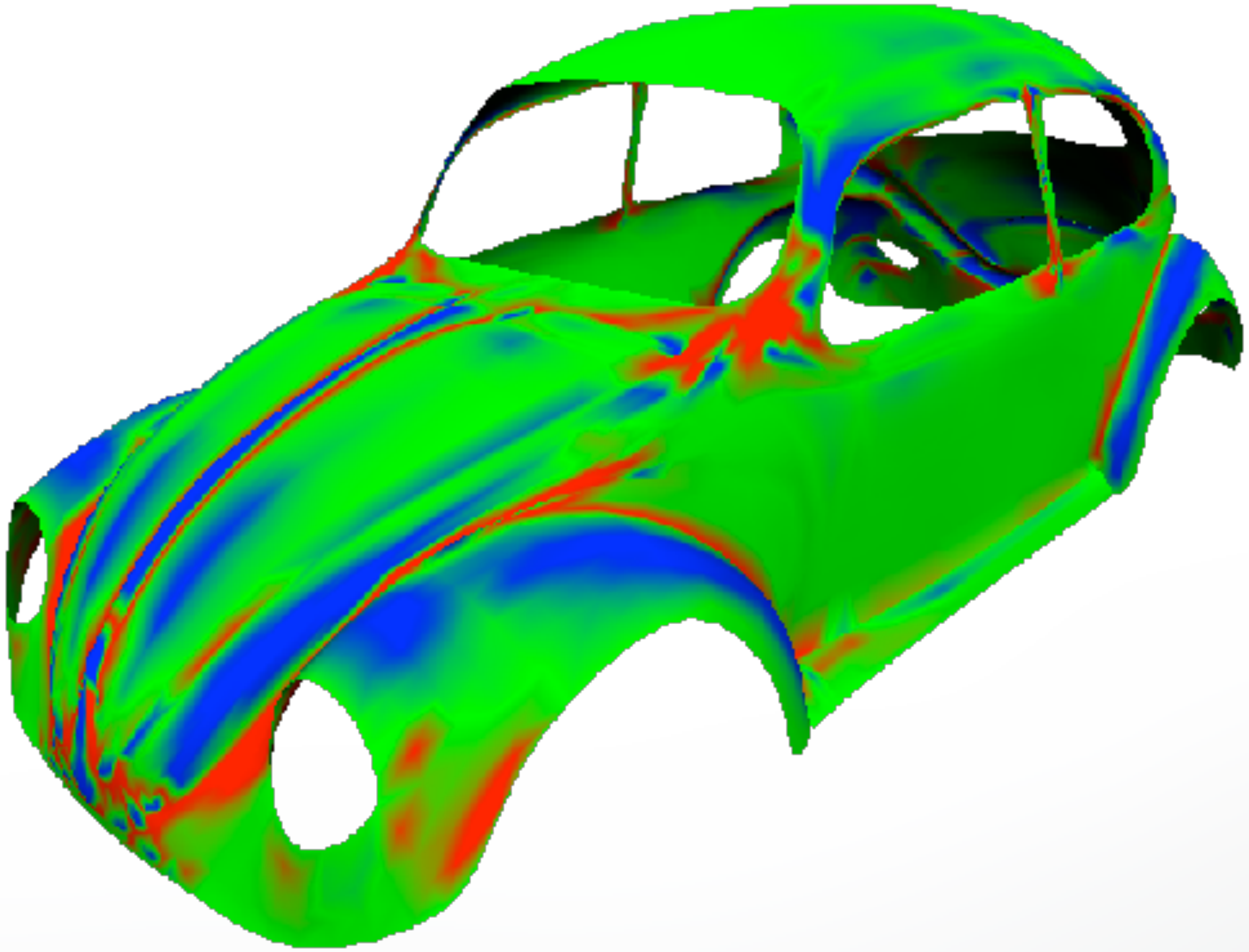
- Specular shading
- Reflection lines
- Curvature
  - Mean curvature



Mesh Quality

Visual inspection of “sensitive” attributes

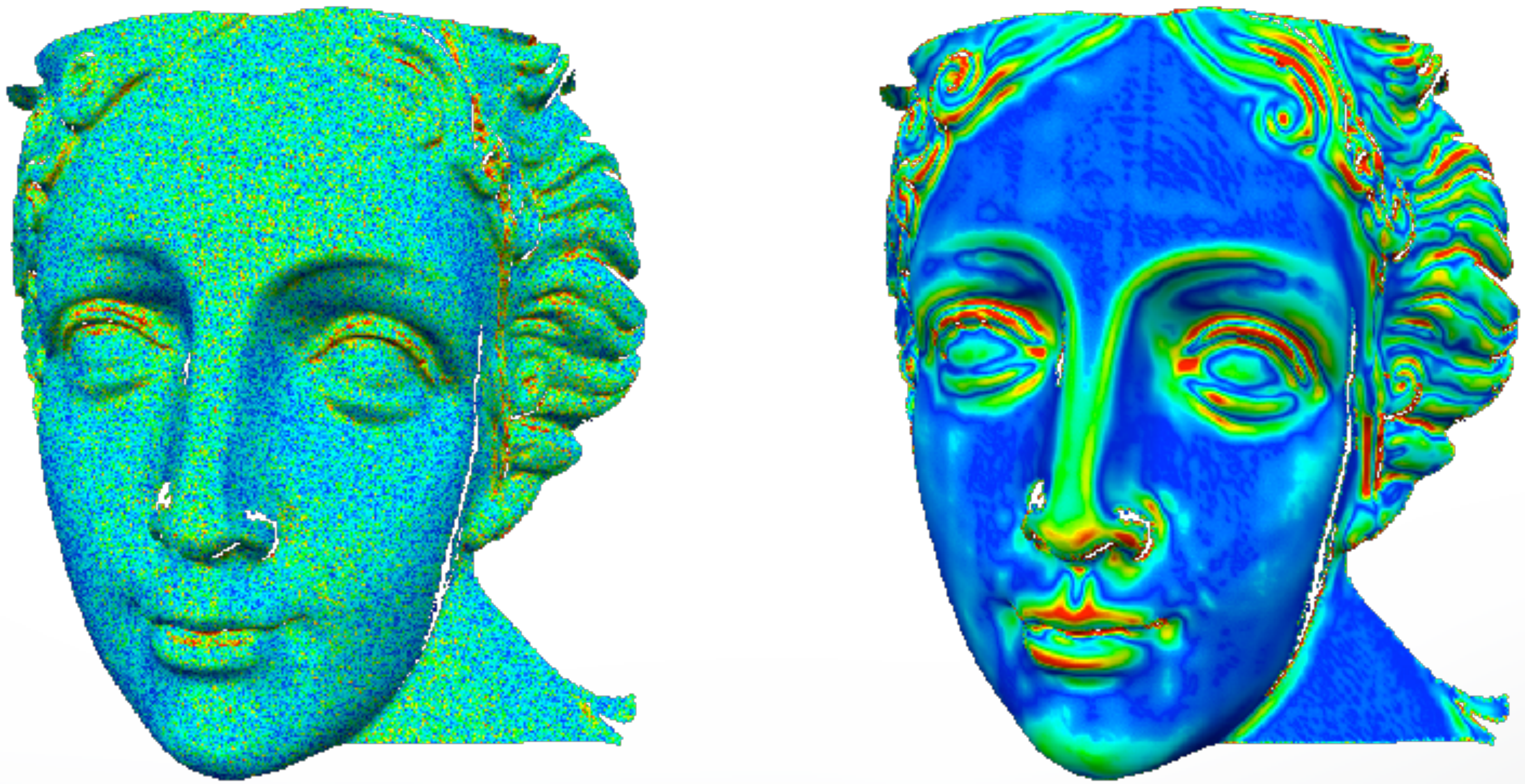
- Specular shading
- Reflection lines
- Curvature
  - Gauss curvature



Mesh Quality Criteria

Smoothness

- Low geometric noise



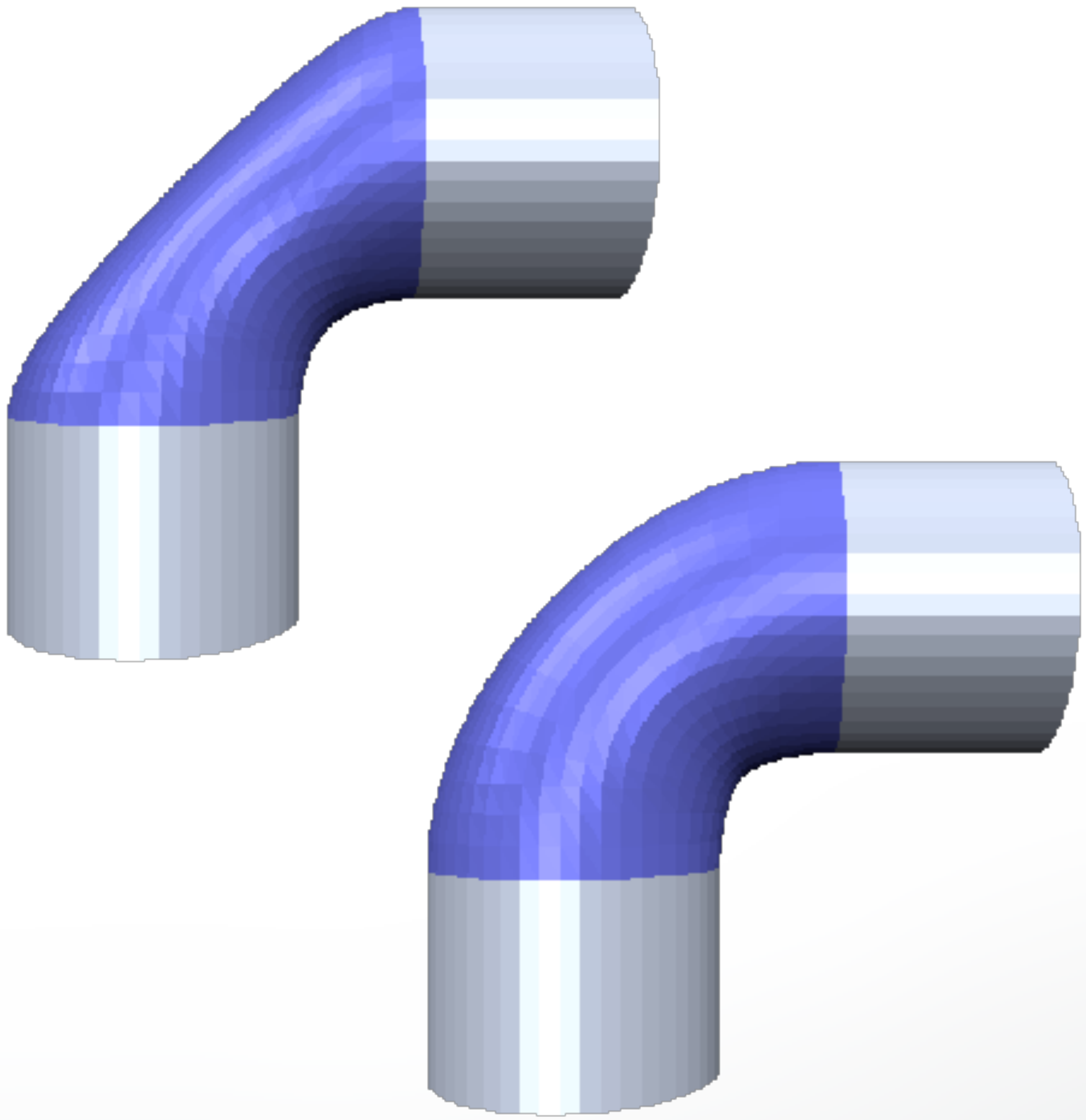
Mesh Quality Criteria

Smoothness

- Low geometric noise

Fairness

- Simplest shape



Mesh Quality Criteria

Smoothness

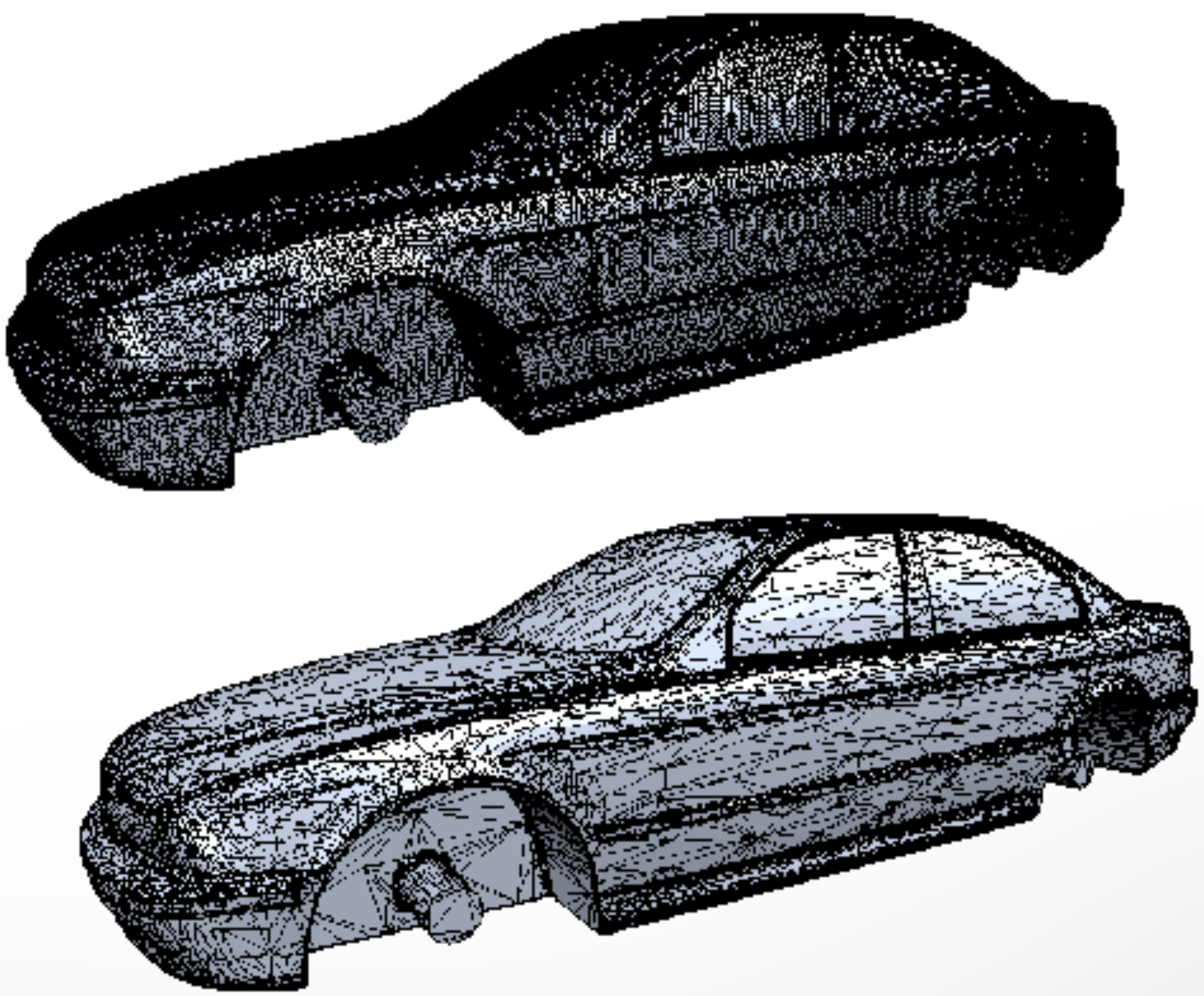
- Low geometric noise

Fairness

- Simplest shape

Adaptive tesselation

- Low complexity



Mesh Quality Criteria

Smoothness

- Low geometric noise

Fairness

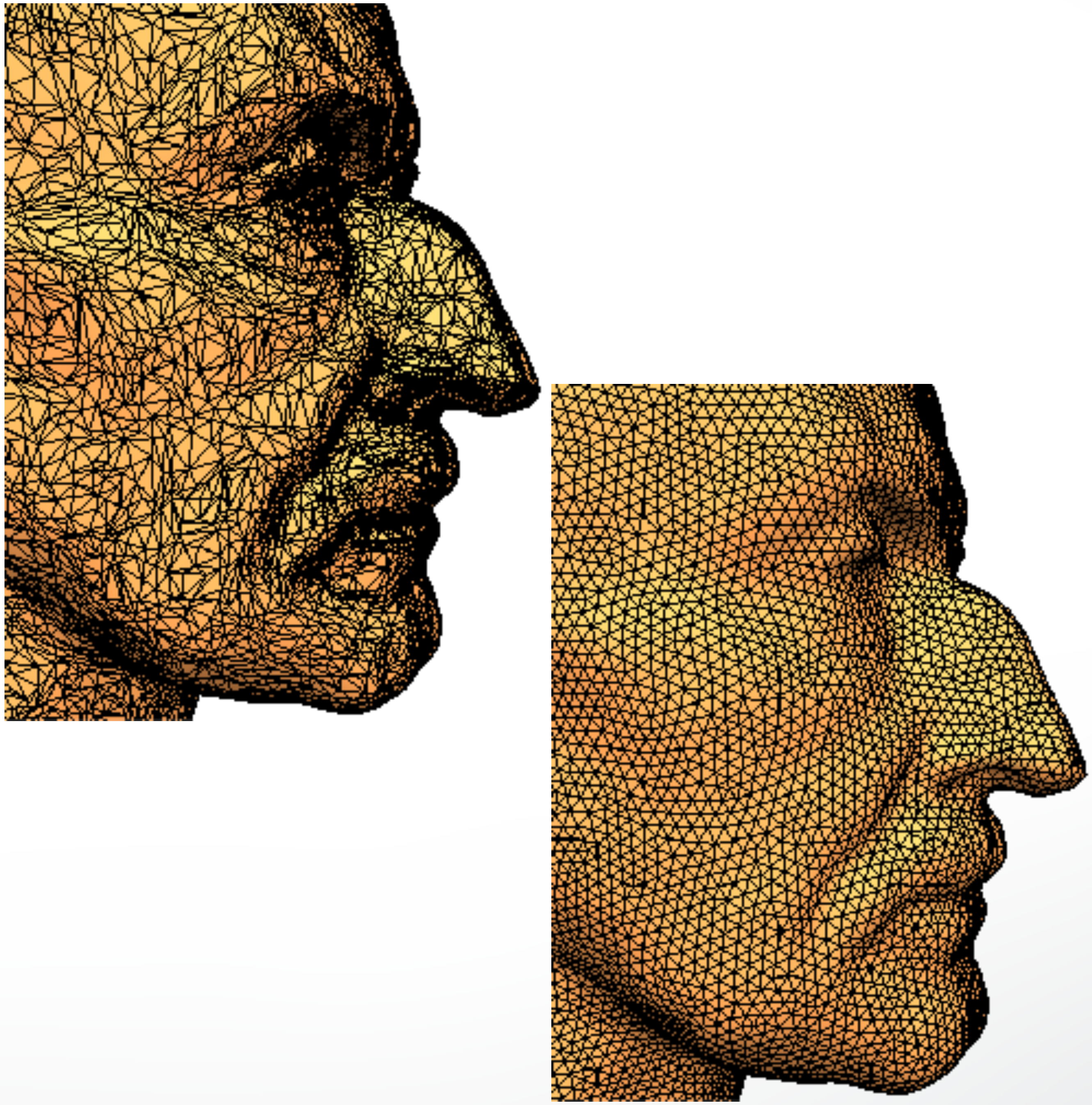
- Simplest shape

Adaptive tesselation

- Low complexity

Triangle shape

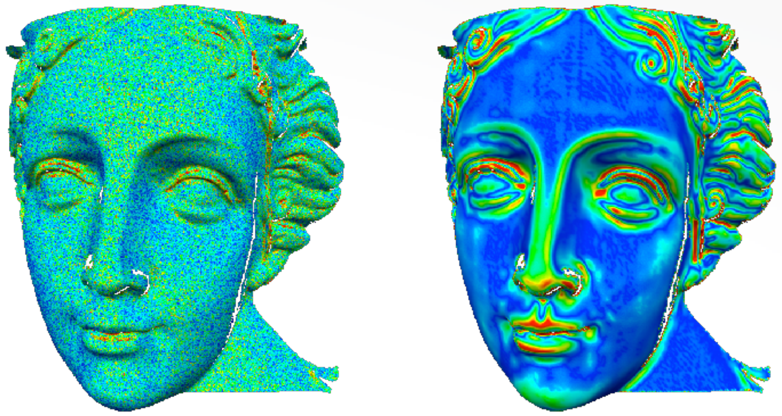
- Numerical Robustness

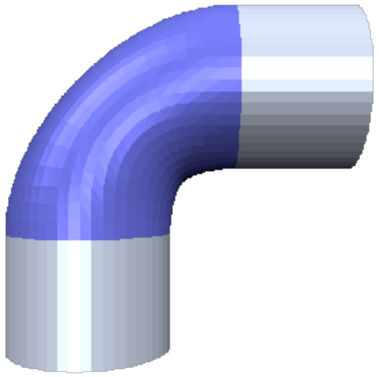


Mesh Optimization

Smoothness

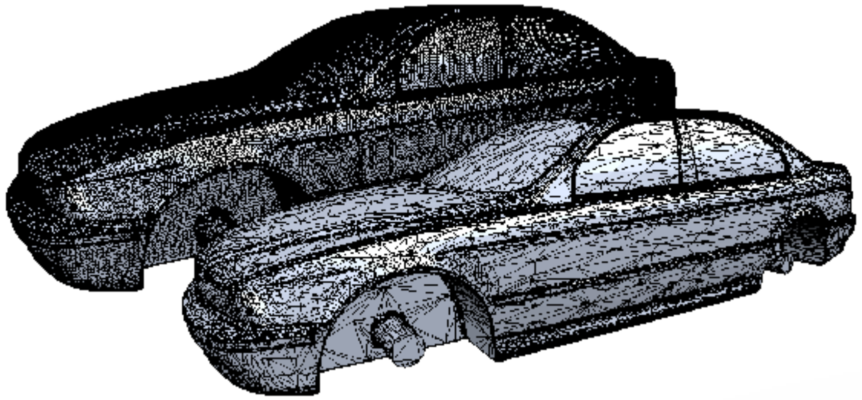
- Smoothing





Fairness

- Fairing

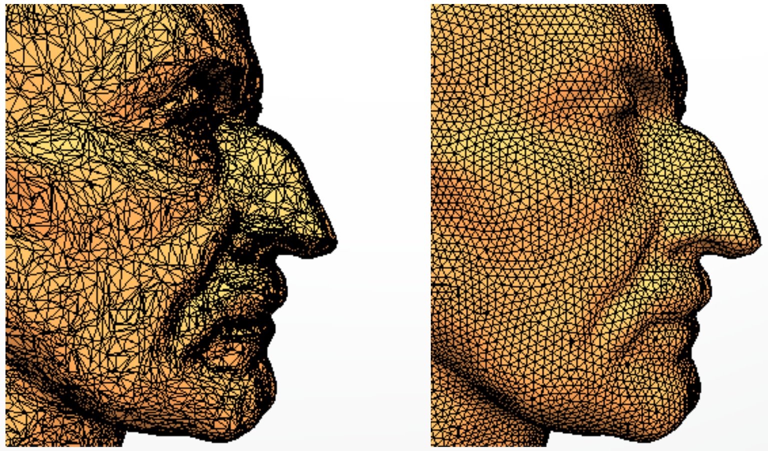


Adaptive tesselation

- Decimation

Triangle shape

- Remeshing



Summary

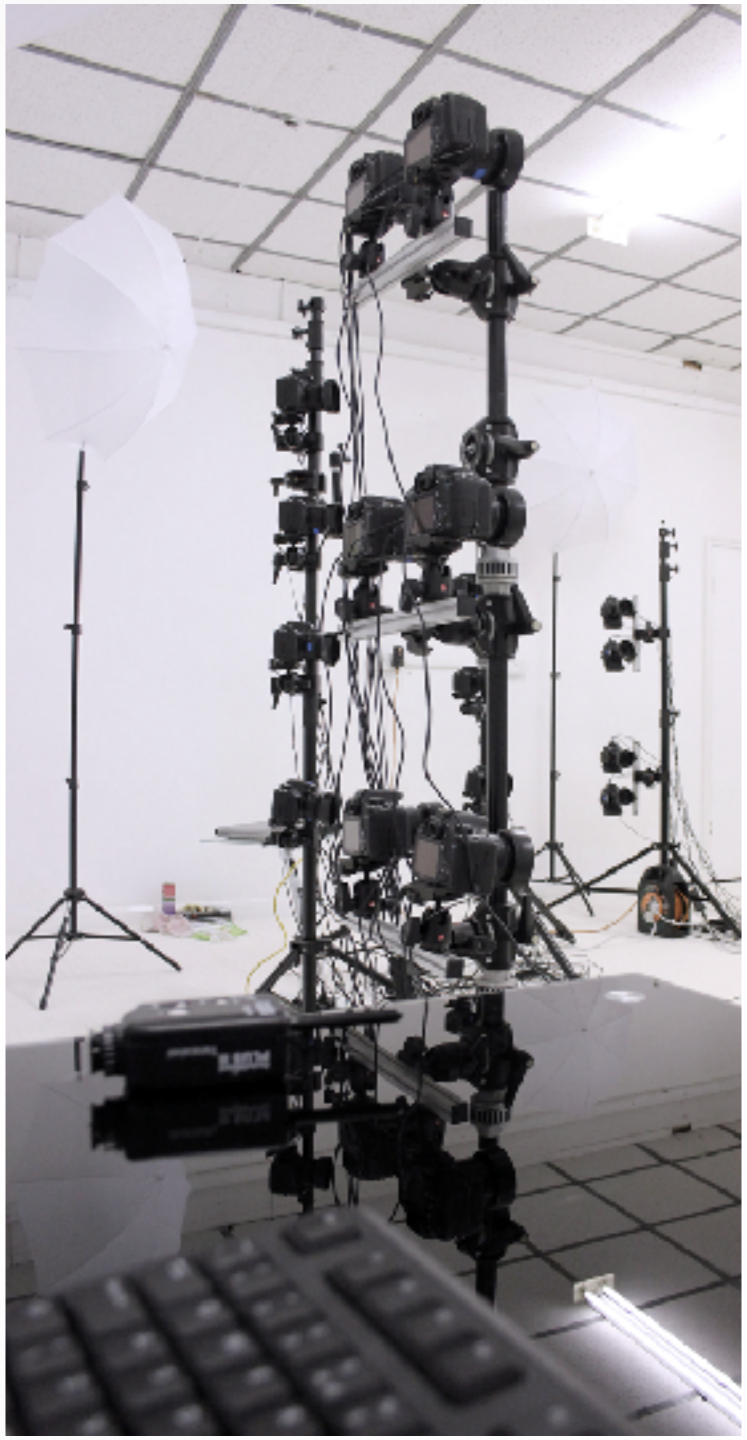
Invariants as overarching theme

- shape does not depend on Euclidean motions (no stretch)
  - **metric** & **curvatures**
- smooth continuous notions to discrete notions
  - generally only as **averages**
- different ways to derive same equations
  - DEC: discrete exterior calculus, FEM, abstract measure theory.

Literature

- Book: Chapter 3
- Taubin: A signal processing approach to fair surface design, SIGGRAPH 1996
- Desbrun et al. : Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow, SIGGRAPH 1999
- Meyer et al.: Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, VisMath 2002
- Wardetzky et al.: Discrete Laplace Operators: No free lunch, SGP 2007

Next Time



3D Scanning

http://cs621.hao-li.com

Thanks!