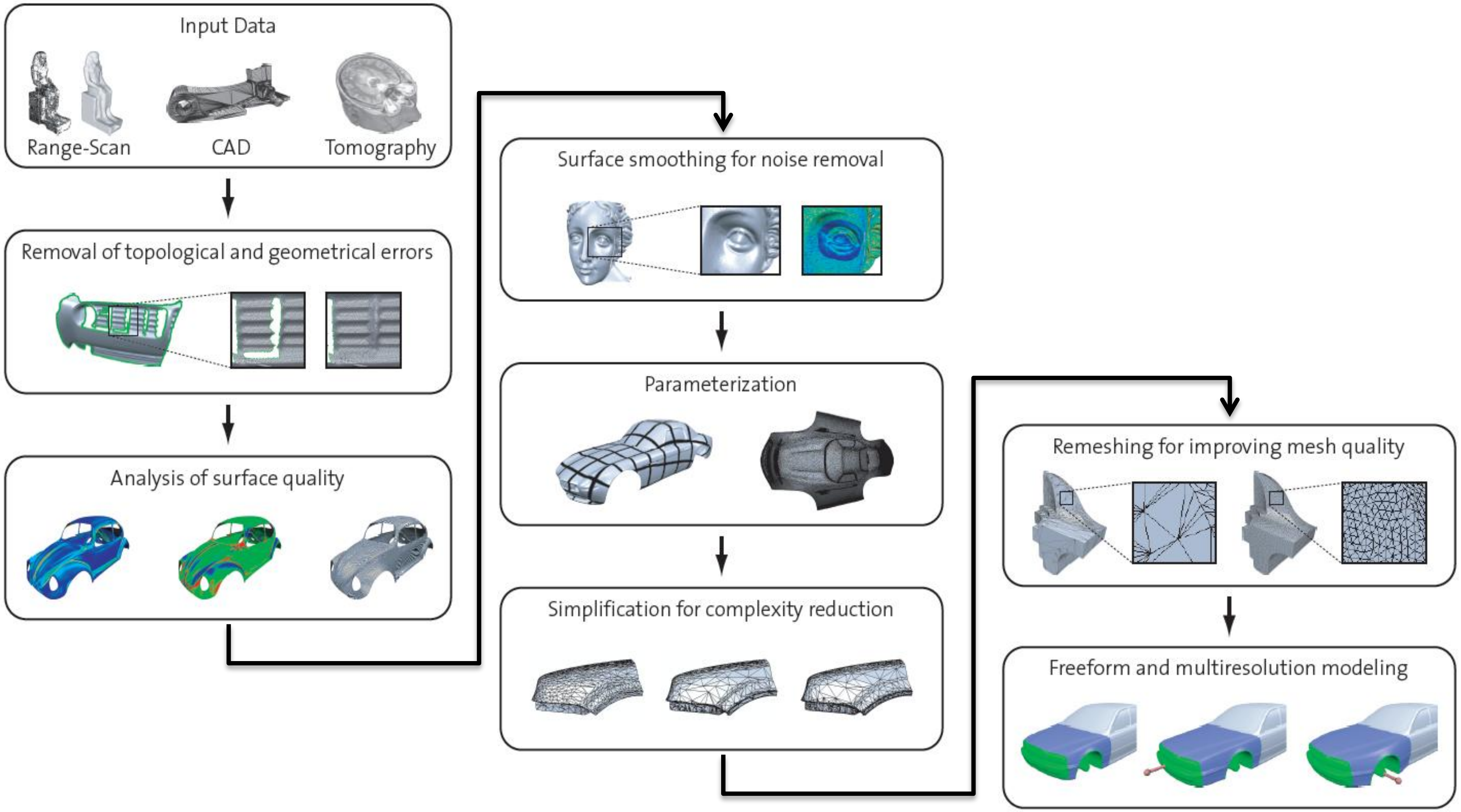
3D Shape Analysis
II. Basic notions

Hamid Laga
Global Edge Institute,
Tokyo Institute of Technology
hamid@img.cs.titech.ac.jp
http://www.img.cs.titech.ac.jp/~hamid/

Geometry processing pipeline

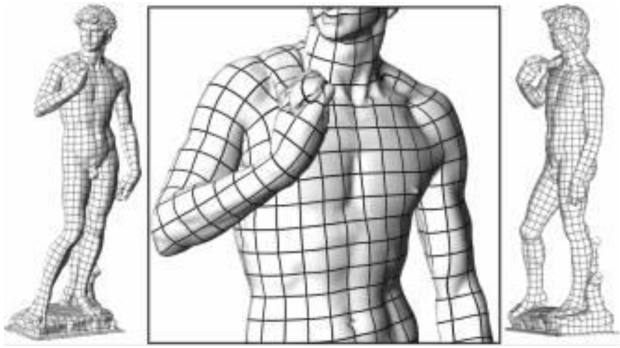
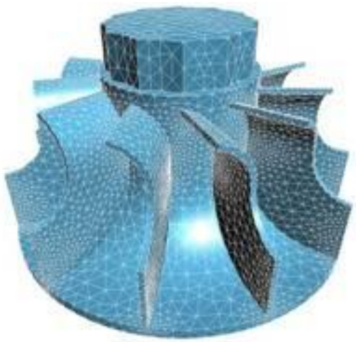
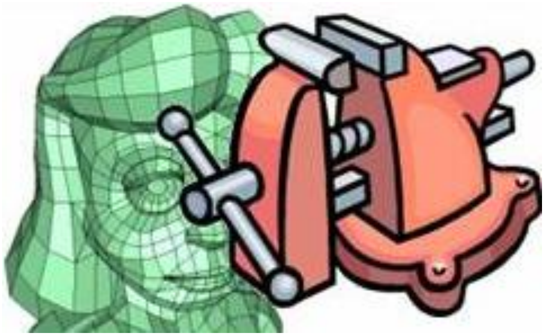


Part II: Basic notions

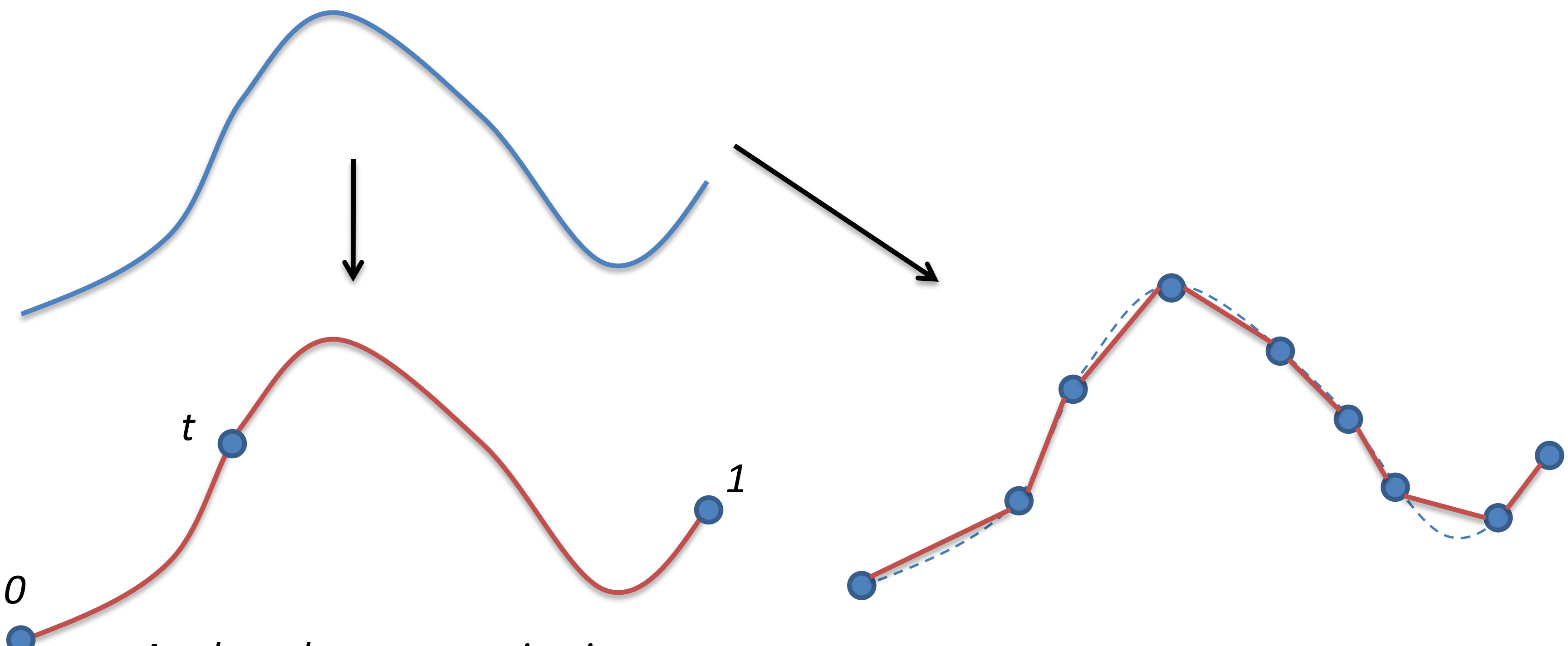
- 3D shape representations
  - Discrete representations
  - (Piecewise) Continuous representations
- Discrete differential geometry (DDG)
  - Differential properties (normals, curvatures, tensors)
  - Local surface analysis
- Application
  - Ridge-valley lines on meshes

Shape representations

- Computerized representation of 3D geometry
- Discrete representations
  - Triangular mesh
  - Polygon soup models

1D Curve



Arc-length parameterization

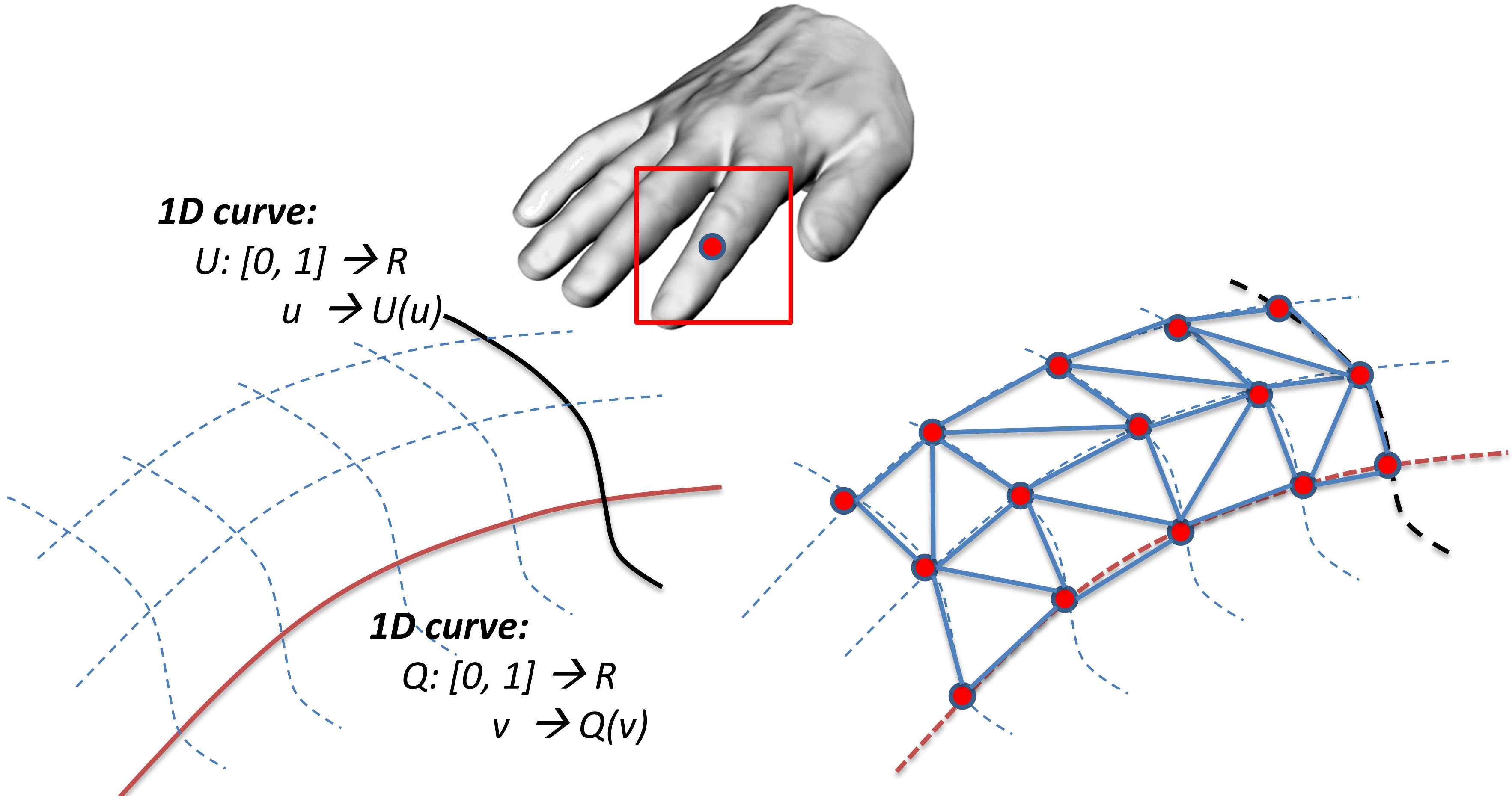
$S: [0, 1] \rightarrow R$

$t \rightarrow S(t) = (x(t), y(t))$

Continuous representation

Discrete representation

Surface in 3D space



Continuous representation of a 2D surface

S: [0, 1] x [0, 1] → R

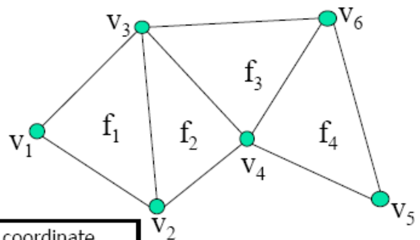
(u, v) → S(u, v)

Discrete representation

Vertices, edges, faces

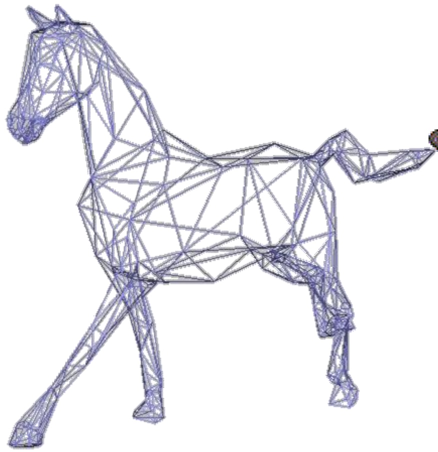
The simplest representation

- Polygon set:
  - List of vertices (3D coordinates)
  - List of faces



vertex	coordinate
v_1	(x_1,y_1,z_1)
v_2	(x_2,y_2,z_2)
v_3	(x_3,y_3,z_3)
v_4	(x_4,y_4,z_4)
v_5	(x_5,y_5,z_5)
v_6	(x_6,y_6,z_6)

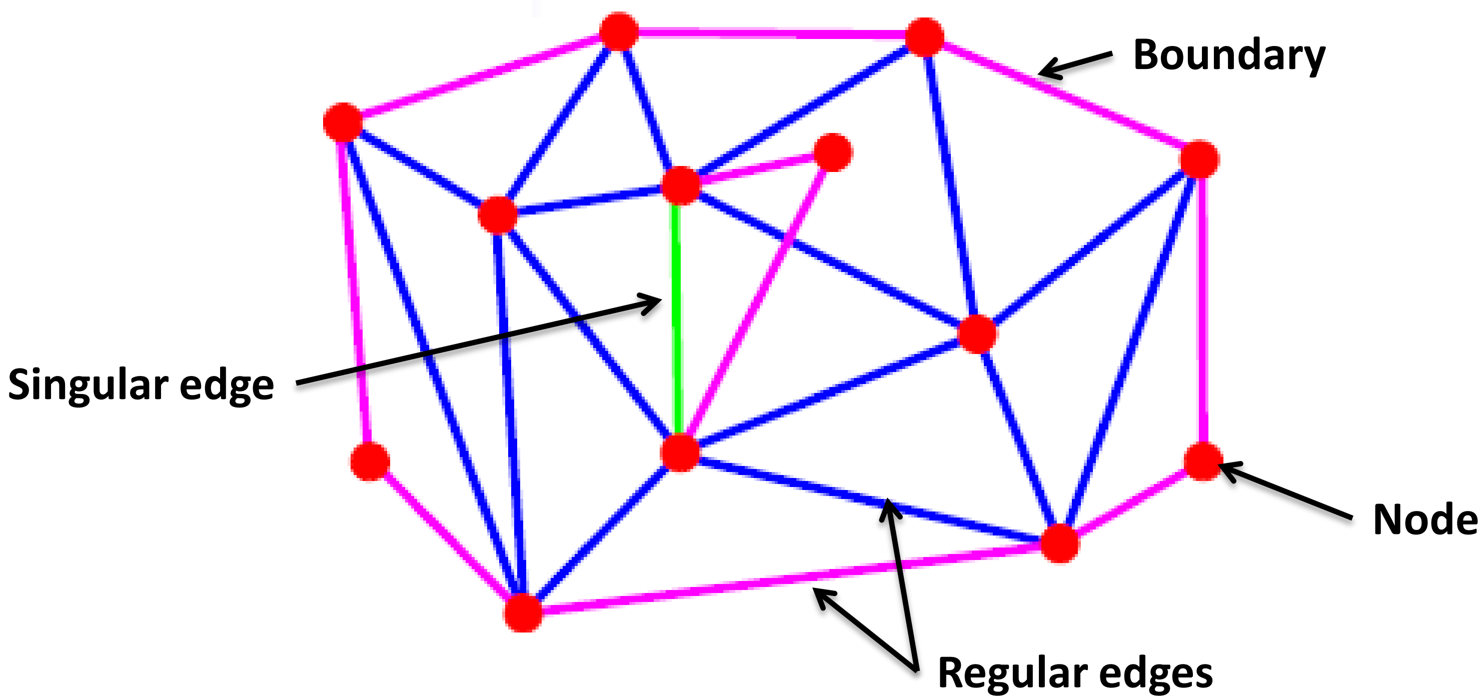
face	vertices (ccw)
f_1	(v_1, v_2, v_3)
f_2	(v_2, v_4, v_3)
f_3	(v_3, v_4, v_6)
f_4	(v_4, v_5, v_6)



Polygons (vertices + faces)

Triangular polygon representation

- Straight-line graph embedded in R^3



Manifold mesh

- No singular edges
  - Each edge has at maximum two adjacent faces
  - Faces intersect only in edges (no self intersections)

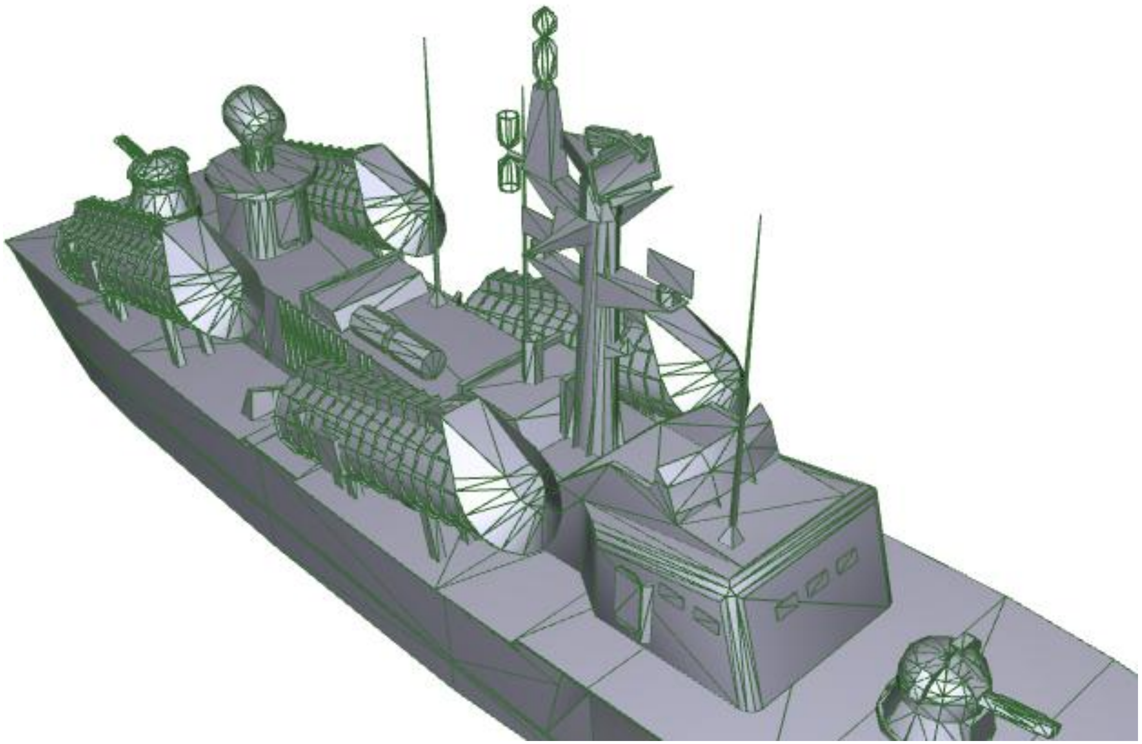
Non-Manifold

Closed Manifold

Open Manifold

Polygon soup model

– No restriction on how polygons are put together



Example II.1

- Mesh smoothing

```
OFF
346 690 0
0.13064 0.44224 0.34724
0.1238 0.3176 0.38448
0.07972 0.33812 0.35104
0.05464 0.4582 0.45288
0.1238 0.46884 0.35712
0.1124 0.46428 0.35712
.............

3 0 1 2
3 3 4 5
3 5 6 2
3 7 4 3
3 8 9 10
3 3 5 2
3 4 11 12
3 4 7 11
3 13 14 15
3 16 12 0
.............
```

V: Vertices
(X, Y, Z)

F: Faces
(V1, V2, V3)

Part II: Basic notions

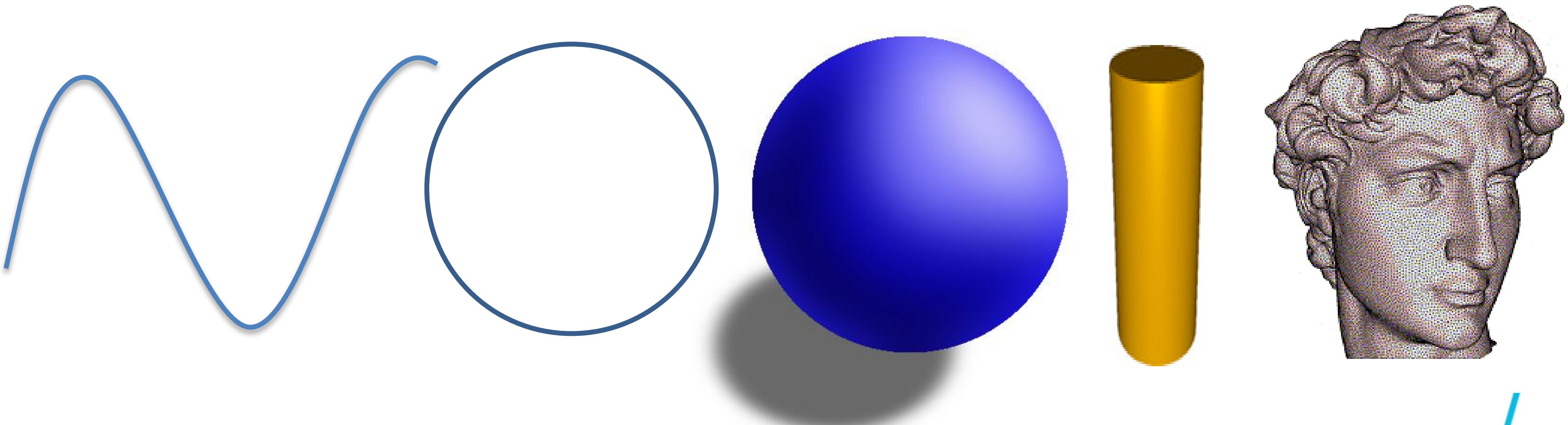
- 3D shape representations
  - Discrete representations
  - (Piecewise) Continuous representations

- Discrete differential geometry (DDG)
  - Differential properties
    (normals, curvatures, tensors)
  - Local surface analysis

- Multiscale, multiresolution analysis

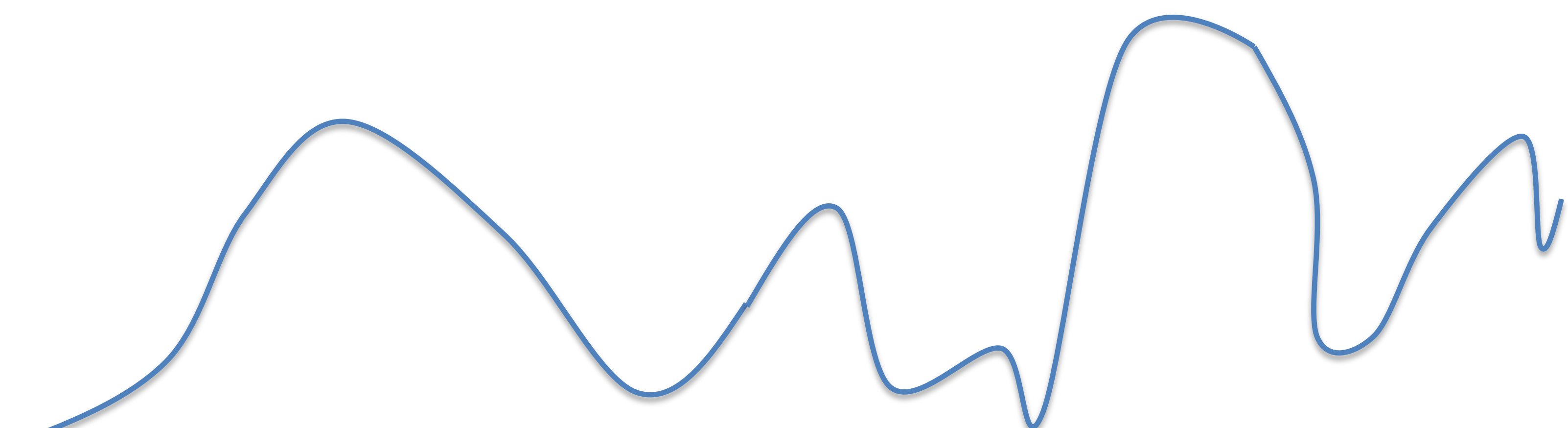
Continuous representation of shapes

- Basic geometric shapes
  - Can be easily represented with a single equation
- Complex shapes
  - Local approximation
    - Represent the surface locally with continuous functions



1D Curve: piecewise continuous

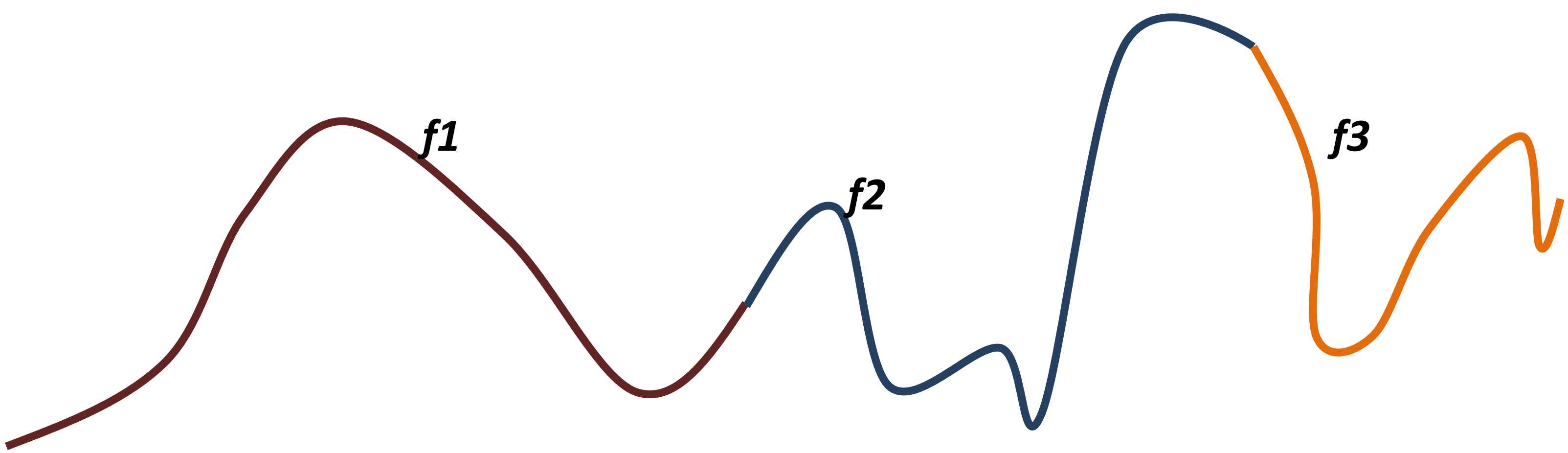
- Complex curves cannot be represented with a single function with sufficient accuracy



Complex curve

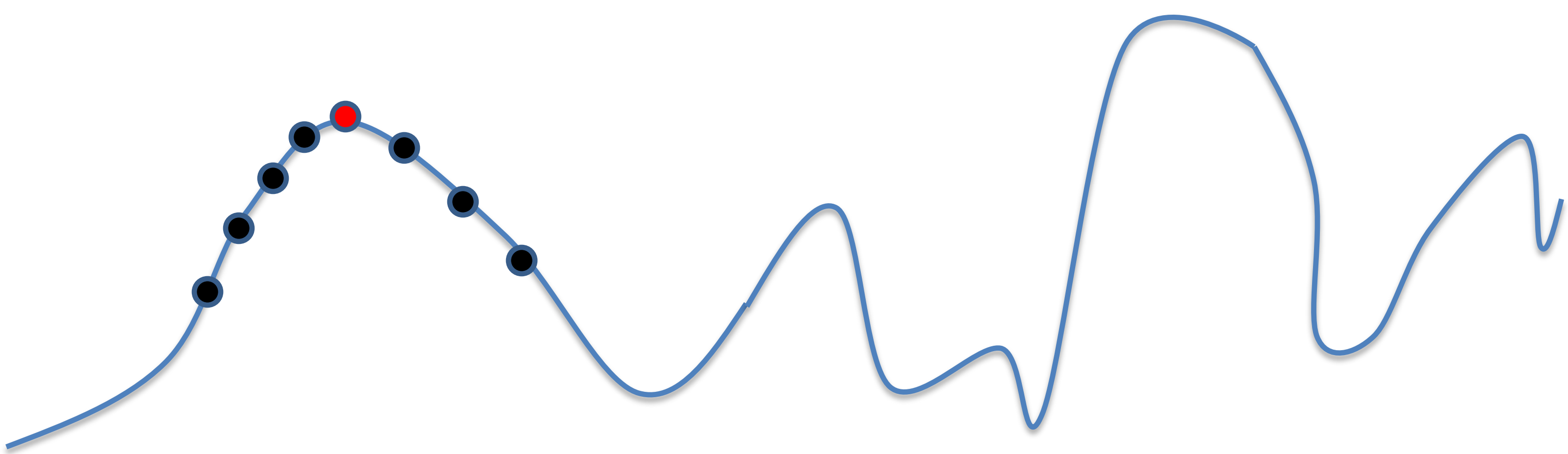
1D curve: piecewise continuous

- Complex curves cannot be represented with a single function with sufficient accuracy
  - Partition the curve into pieces
  - Represent each piece with a function (polynomial)



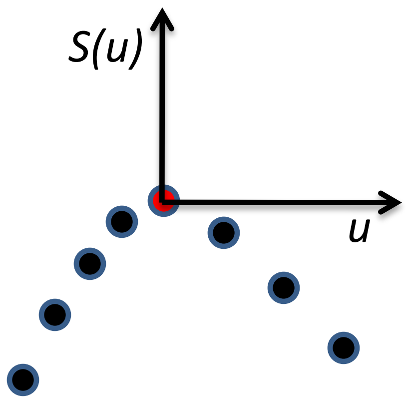
1D curve: piecewise continuous

- Complex curves cannot be represented with a single function with sufficient accuracy
  - Local approximation



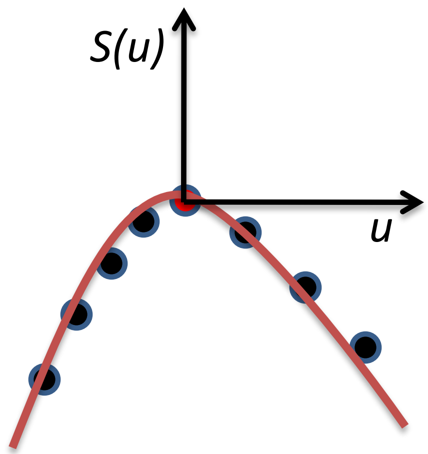
1D curve: piecewise continuous

- Complex curves cannot be represented with a single function with sufficient accuracy
  - Local approximation



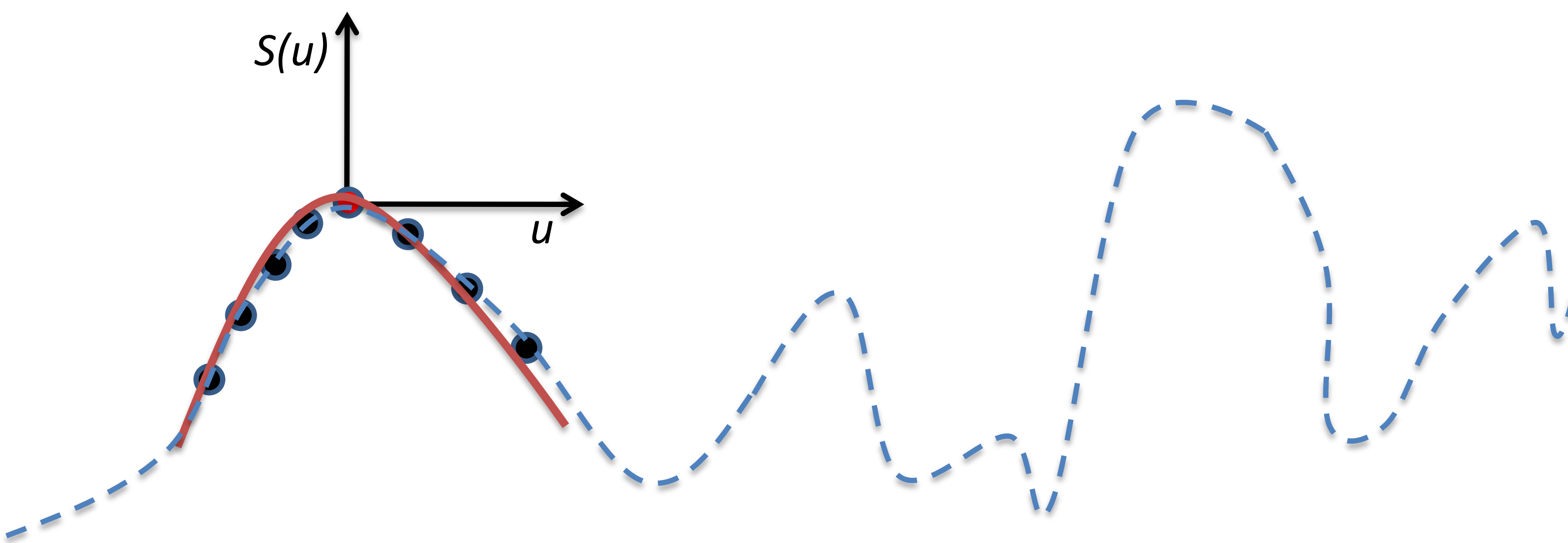
1D curve: piecewise continuous

- Complex curves cannot be represented with a single function with sufficient accuracy
  - Local approximation



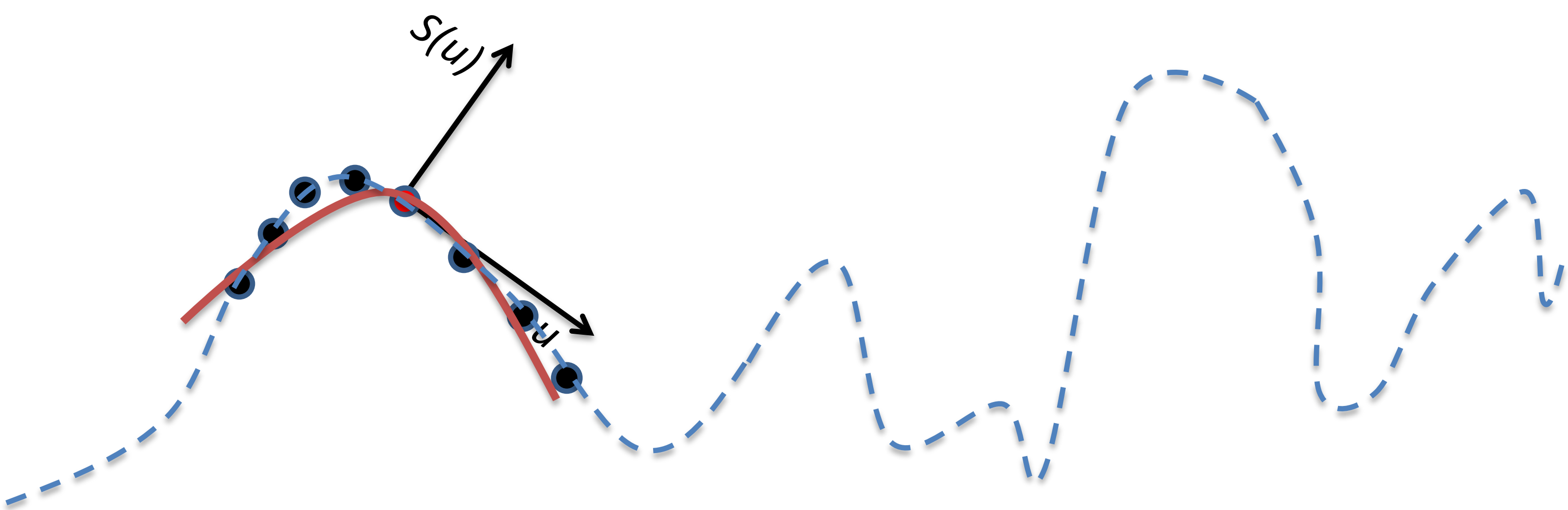
1D curve: piecewise continuous

- Complex curves cannot be represented with a single function with sufficient accuracy
  - Local approximation



1D curve: piecewise continuous

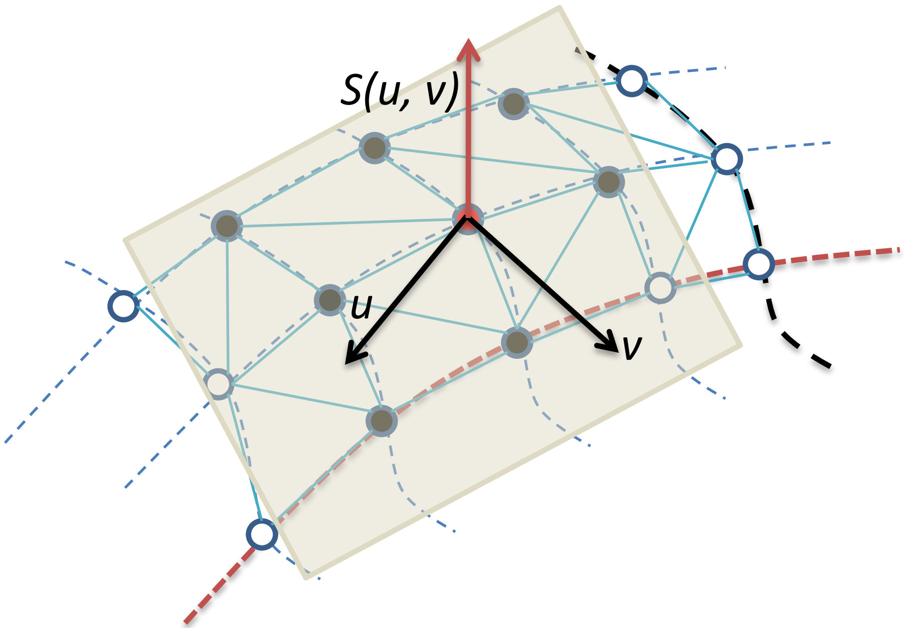
- Complex curves cannot be represented with a single function with sufficient accuracy
  - Local approximation



How about surfaces

- At each vertex
  - Build local coordinate system (u,v)
  - Collect N neighbor vertices
  - Fit a polynomial patch

$$S(u,v) = a_0 + a_1 u + a_2 v + a_3 u^2 + a_4 uv + a_5 v^2$$



TOKYO TECH Pursuing Excellence

From mesh to piecewise continuous

- Quadric polynomial fitting
  - Collect at least 6 vertices p_i, $i=1\ldots6$
  - Represent them in local coordinate frame
    - $p_i = (u_i, v_i, S(u_i, v_i))$
  - Fit the quadric by minimizing the Mean Least Square (MLS) error

$$S(u_1,v_1) = a_0 + a_1u_1 + a_2v_1 + a_3u_1^{\;2} + a_4u_1v_1 + a_5v_1^{\;2}$$

$$S(u_1,v_1) = a_0 + a_1u_2 + a_2v_2 + a_3u_2^{\;2} + a_4u_2v_2 + a_5v_2^{\;2}$$

$$\ldots$$

$$S(u_n,v_n) = a_0 + a_1u_n + a_2v_n + a_3u_n^{\;2} + a_4u_nv_n + a_5v_n^{\;2}$$

From mesh to piecewise continuous

- Quadric polynomial fitting

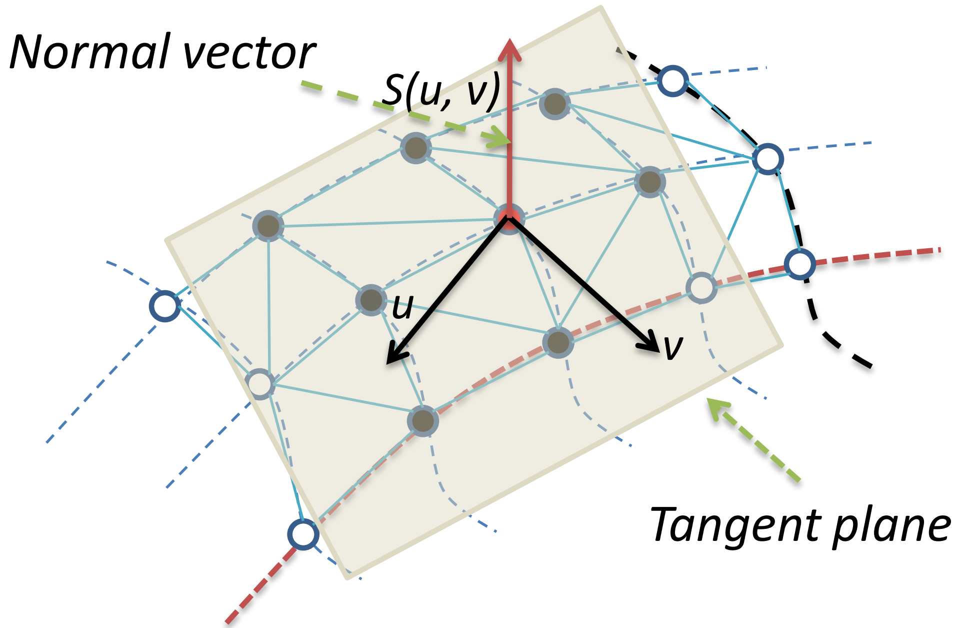
$$\underbrace{\begin{bmatrix} S(u_1,v_1) \\ S(u_2,v_2) \\ \cdots \\ S(u_n,v_n) \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} 1 & u_1 & v_1 & u_1^2 & u_1 v_1 & v_1^2 \\ 1 & u_2 & v_2 & u_2^2 & u_2 v_2 & v_2^2 \\ \cdots & & & & & \\ 1 & u_n & v_n & u_n^2 & u_n v_n & v_n^2 \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_0 \\ a_5 \end{bmatrix}}_{\mathbf{X}}$$

$$b = MX$$

$$X = \left(M^t M\right)^{-1} M^t b$$

Why do we need the piecewise continuous ?

- First derivatives
  - Tangent plane to the surface $$\frac{\partial S(u,v)}{\partial u}, \frac{\partial S(u,v)}{\partial v}$$
  - Normal to the surface $$\vec{n} = \frac{\partial S(u,v)}{\partial u} \times \frac{\partial S(u,v)}{\partial v}$$



Why do we need the piecewise continuous ?

- Second derivatives
  - Related to the surface curvature
    (we will see it soon)

Example II.2

- Polynomial fitting

Part II: Basic notions

- 3D shape representations
  - Discrete representations
  - (Piecewise) Continuous representations

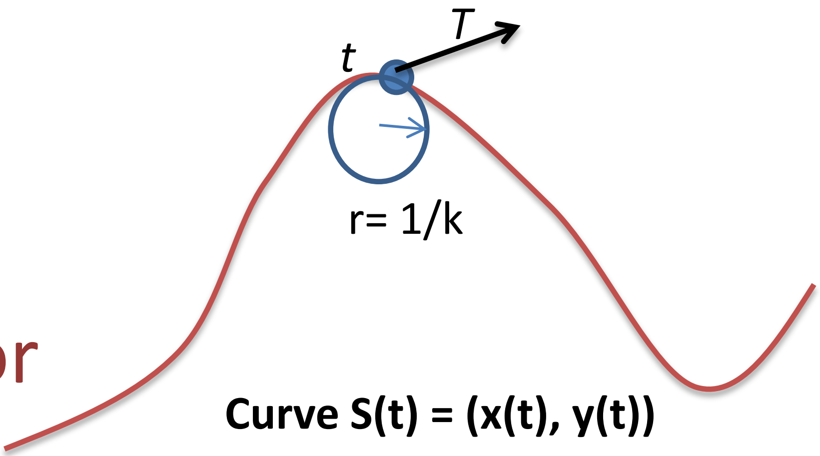
- Discrete differential geometry (DDG)
  - Differential properties (normals, curvatures, tensors)
  - Local surface analysis

- Multiscale, multiresolution analysis

Curves

– Tangent vector

$$T = \frac{dS(t)}{dt} = \left[\frac{dx(t)}{dt}, \frac{dy(t)}{dt} \right]$$



– Unit length tangent vector

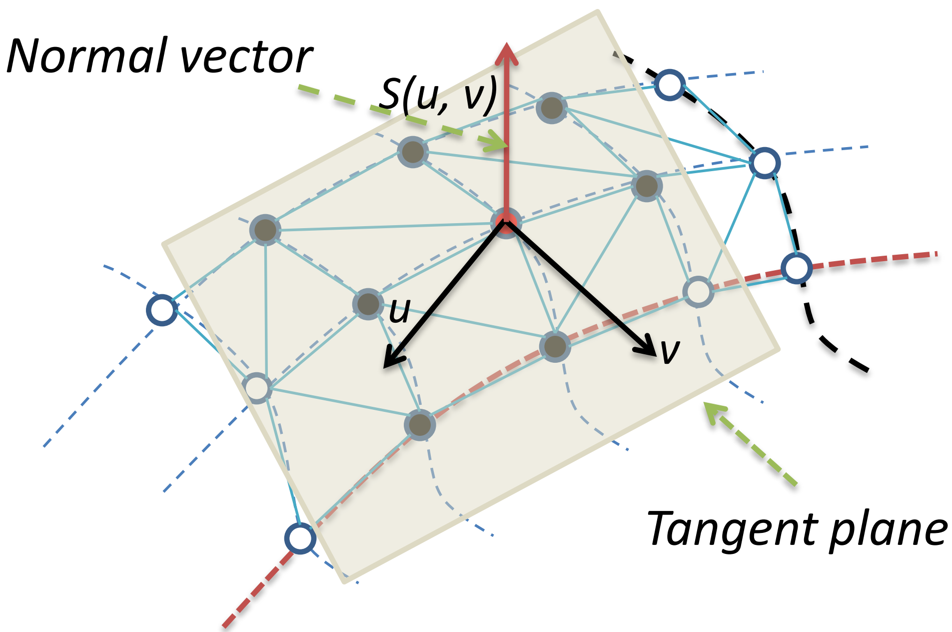
$$\vec{T} = \left[\frac{x'(t)}{\sqrt{x'(t)^2 + y'(t)^2}}, \frac{y'(t)}{\sqrt{x'(t)^2 + y'(t)^2}} \right]$$

– Curvature (measure of the curve bending)

$$k(t) = \frac{x'(t)y''(t) - y'(t)x''(t)}{\left(x'(t)^2 + y'(t)^2\right)^{3/2}}$$

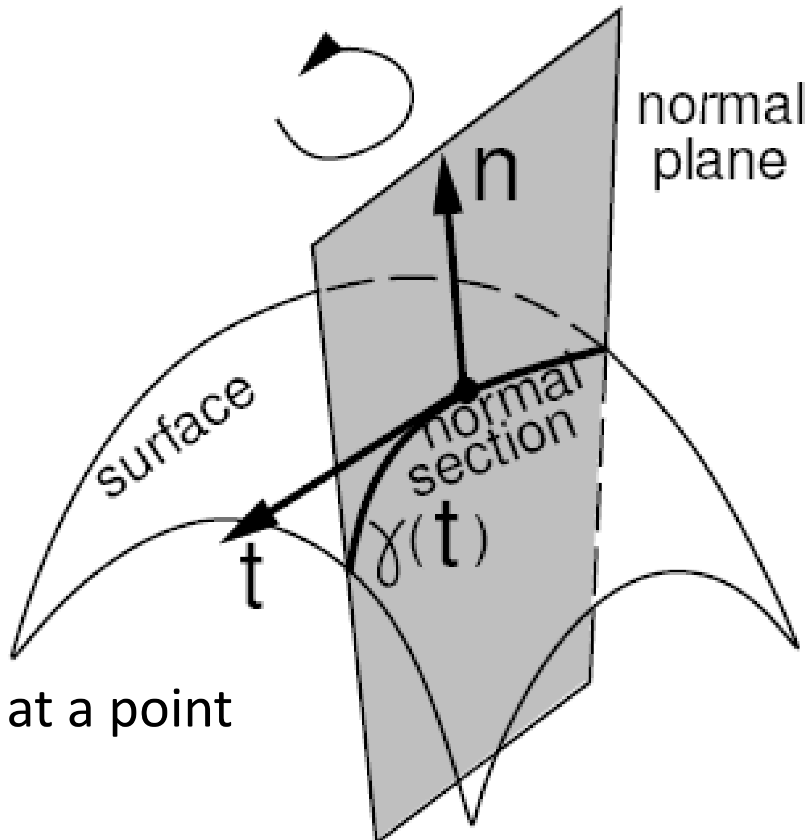
Why do we need the piecewise continuous ?

- First derivatives
  - Tangent plane to the surface $$\frac{\partial S(u,v)}{\partial u}, \frac{\partial S(u,v)}{\partial v}$$
  - Normal to the surface $$\vec{n} = \frac{\partial S(u,v)}{\partial u} \times \frac{\partial S(u,v)}{\partial v}$$



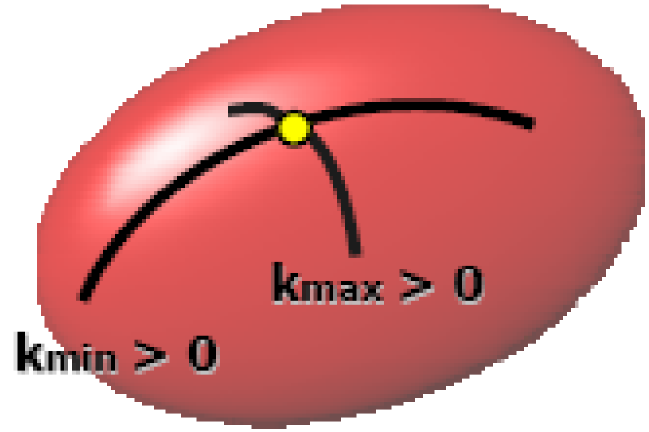
Curvatures

- Normal curvature
  - Curvature of the normal section
  - There are many normal sections
    → normal curvature is not unique
- Principal curvatures
  - Minimum curvature
    - Min of normal curvatures at a point
  - Maximum curvature
    - Maximum of normal curvatures at a point



Curvatures

- Principal curvatures (Kmin, Kmax)
  - Minimum (Kmin) and maximum of the normal curvatures at a point
- Principal directions
  - Two orthogonal tangential directions
  - Correspond to min / max curvatures



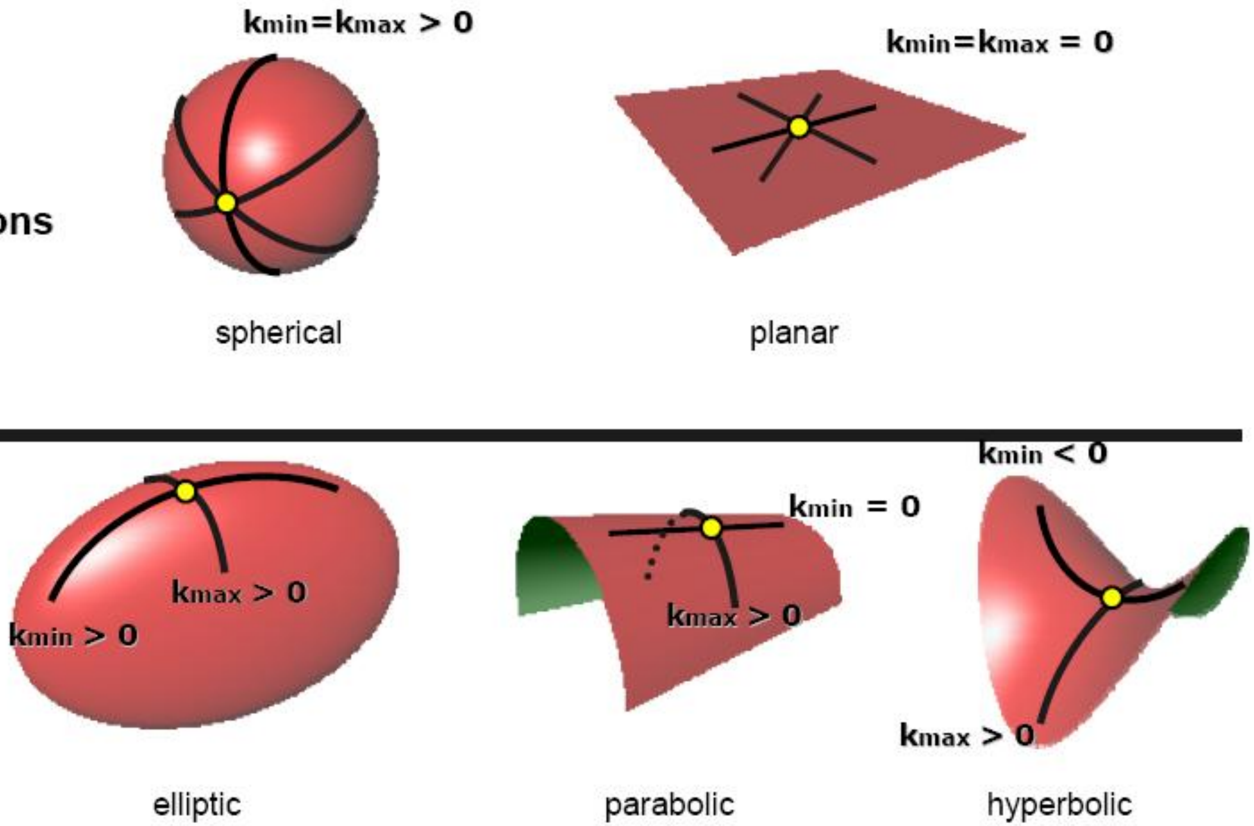
Curvature analysis

Isotropic

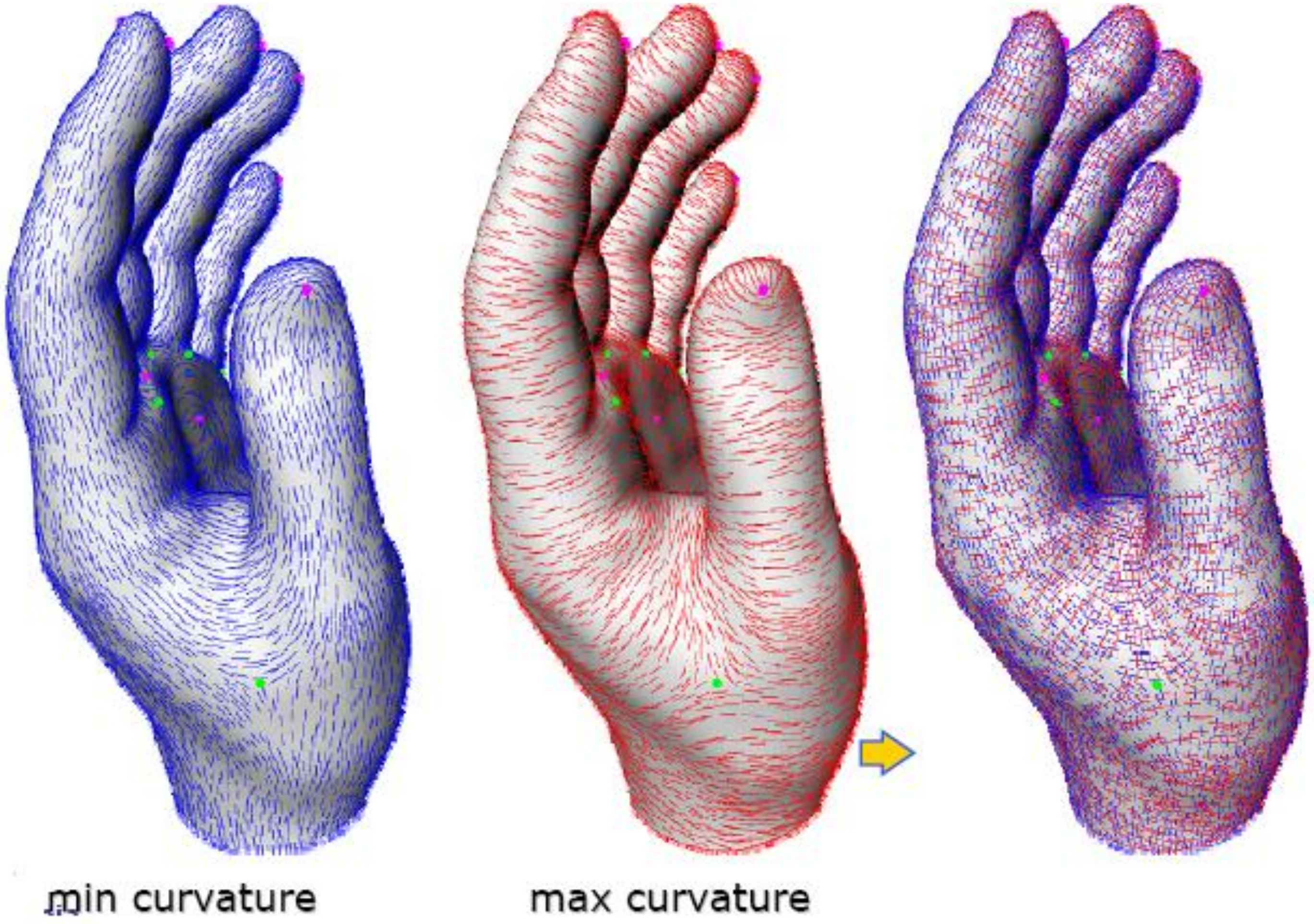
Equal in all directions

Anisotropic

2 distinct principal directions



Principal directions of curvature



Curvature analysis

- Curvature analysis for local shape understanding
  - Kmin = kmax > 0→ sphere
  - Kmin = Kmax = 0 → planar
  - Kmin > o, Kmax > 0 →elliptic
  - Kmin = 0, Kmax > 0 →parabolic (ex. cyllindric surface)
  - Kmin <0, Kmax > 0 →hyperbolic surface
- For global shape understanding
  - Analyze the distribution of the curvature (histogram)

Other curvatures

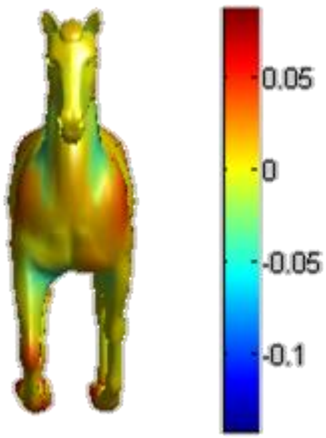
– Gaussian curvature

$$K = K_{\min} K_{\max}$$

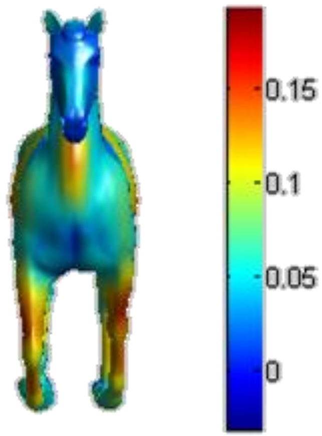
– Mean curvature

$$H = \frac{1}{2}(K_{\min} + K_{\max})$$

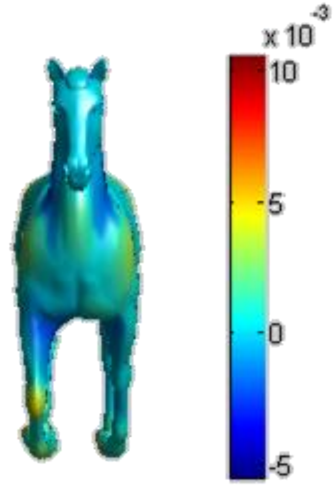
Examples



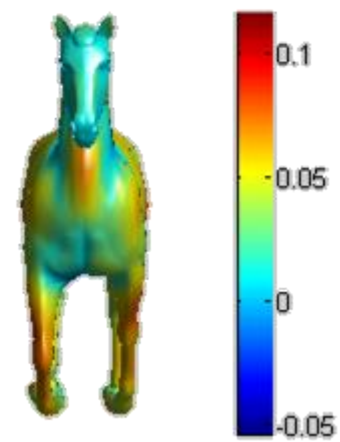
Min curvature



Max curvature

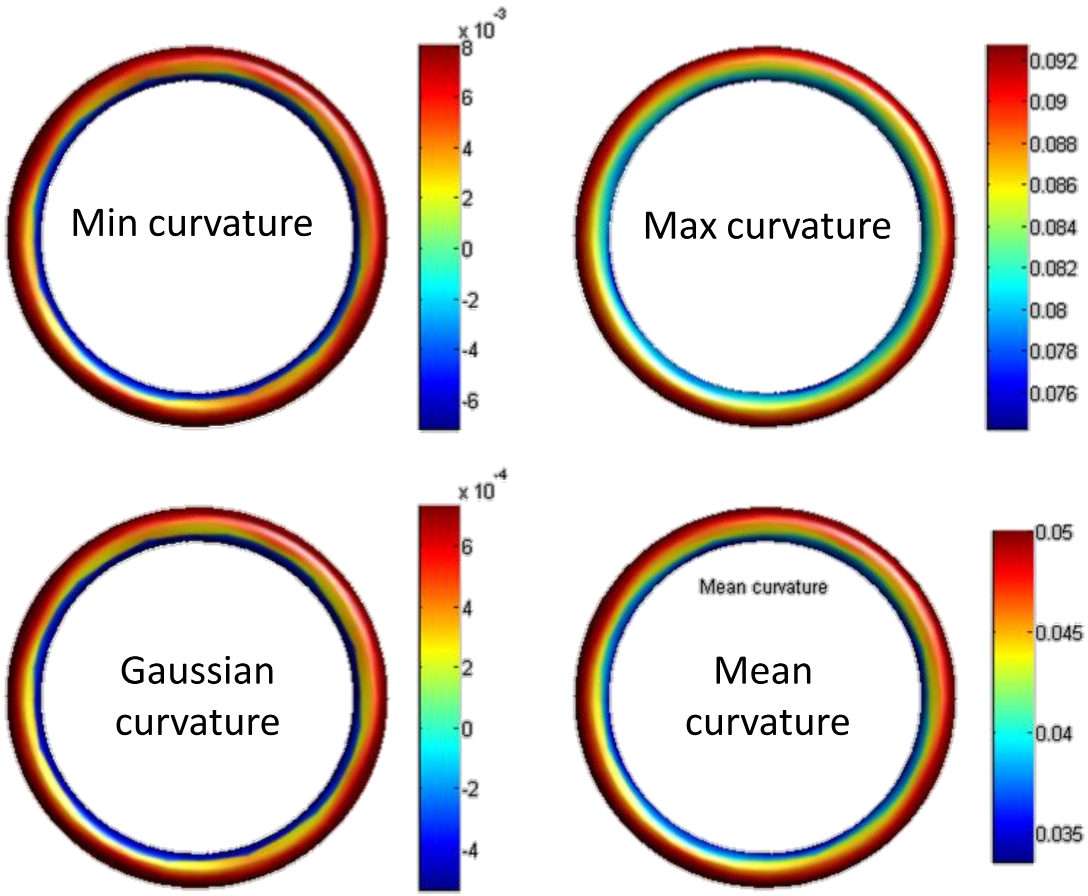


Gaussian curvature



Mean curvature

Examples



Min curvature
x 10^-3
8
6
4
2
0
-2
-4
-6
Max curvature
0.092
0.09
0.088
0.086
0.084
0.082
0.08
0.078
0.076
Gaussian curvature
x 10^-4
6
4
2
0
-2
-4
Mean curvature
Mean curvature
0.05
0.045
0.04
0.035

Normal estimation on mesh

- Option1
  - Average face normals around a vertex
  - Problem:
    - Does not reflect face “influence”

Normal estimation on mesh

- Option2
  - Weighted average face normals around a vertex
    - Use face area or angles at vertex

Normal estimation on mesh

– Option3
  – Estimate the tangent plane and take the normal to that
    - Center the data: vertex + its neighbors
    - Compute covariance matrix
    - Tangent plane spanned by the two largest eigenvectors of the covariance matrix
    - Normal is the eigenvector with smallest eigenvalue
      – What about the orientation ?

Practical (efficient) curvature estimation

- Quadratic approximation $S(u,v) = au^2 + buv + cv^2$
- Principal curvatures kmin and kmax are real roots of:

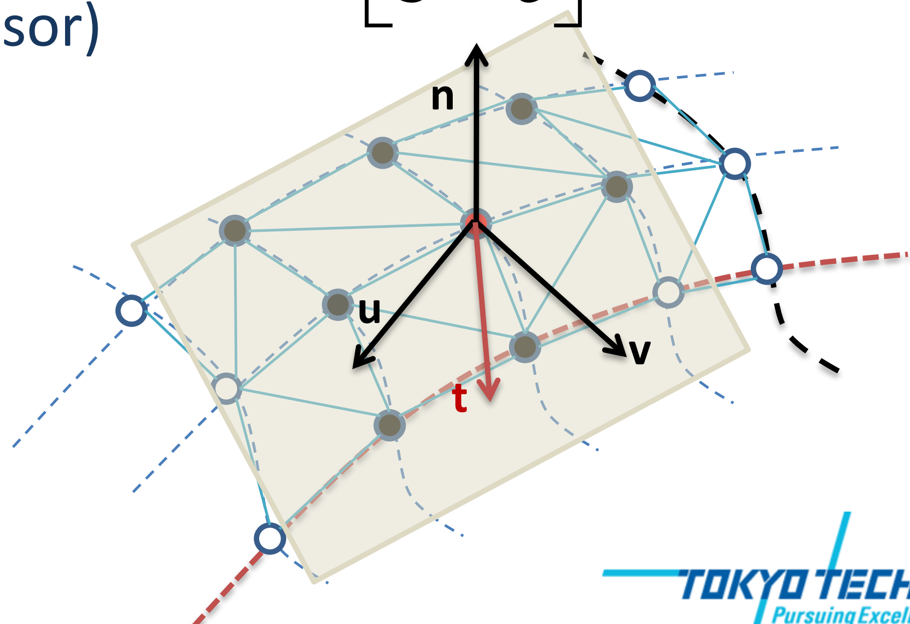
$$k^2 - (a-c)k + ac - b^2 = 0$$

- Mean curvature $H = \frac{1}{2}(K_{\min} + K_{\max})$
- Gaussian curvature $K = K_{\min} K_{\max}$

Curvature tensors

- Quadratic approximation $S(u,v) = au^2 + buv + cv^2$
- Given: **t**: a unit tangential direction of coord. (u, v)
- We want: the normal curvature kt in direction t ?
- The second fundamental form $II = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$
  - (Called also curvature tensor)
- Normal curvature

$$k_t = [u \quad v] II \begin{bmatrix} u \\ v \end{bmatrix}$$



TOKYO TECH Pursuing Excellence

Curvature tensors

- Exact tensor formula

$$II = \begin{bmatrix} \frac{\partial \vec{n}}{\partial u}\vec{u} & \frac{\partial \vec{n}}{\partial v}\vec{u} \\ \frac{\partial \vec{n}}{\partial u}\vec{v} & \frac{\partial \vec{n}}{\partial v}\vec{v} \end{bmatrix}$$

- Principal curvatures (kmin, kmax):
  - Eigenvalues of II
- Principal directions
  - Eigenvectors

Application of curvature analysis

– Ridge and valley detection

TOKYO TECH Pursuing Excellence

Application of curvature analysis

– Ridge and valley detection

– Given

- Principal curvatures k_{min}, k_{max}
- Their associated principal directions t_{min}, t_{max}
- The curvature derivatives along the principal directions

$$e_{max} = \partial k_{max} / \partial \mathbf{t}_{max} \quad e_{min} = \partial k_{min} / \partial \mathbf{t}_{min}$$

$$\begin{aligned} e_{max} &= 0, \quad \partial e_{max} / \partial \mathbf{t}_{max} < 0, \quad k_{max} > |k_{min}|, \quad &\text{(ridges)}, \\ e_{min} &= 0, \quad \partial e_{min} / \partial \mathbf{t}_{min} > 0, \quad k_{min} < -|k_{max}| \quad &\text{(valleys)}. \end{aligned}$$

TOKYO TECH Pursuing Excellence

Summary

- Triangular mesh representation
  - The most flexible
  - Can be used for rendering, geometry processing, shape analysis
- Differential properties
  - Curvature analysis very useful for local (and global) shape understanding

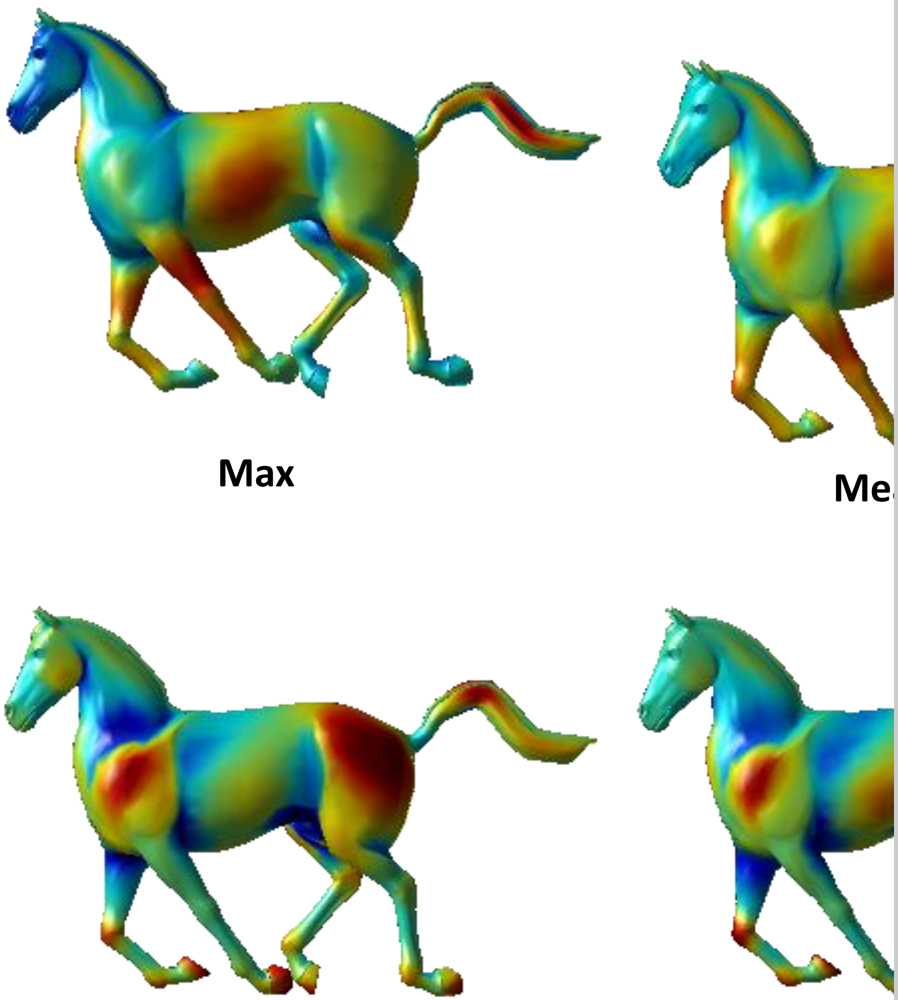
References

- Efficient differential properties estimation
  - Szymon Rusinkiewicz.
    Estimating Curvatures and Their Derivatives on Triangle Meshes
  - Pierre Alliez et al.
    Anisotropic Polygonal Remeshing. In ACM Transactions on Graphics, 2003.
  - Gabriel Taubin.
    Estimating the tensor of curvature of a surface from a polyhedral approximation.
- Applications
  - Yutaka Ohtake , Alexander Belyaev, Hans-Peter Seidel.
    Ridge-Valley Lines on Meshes via Implicit Surface Fitting. ACM Trans. Graphics 2004
- Similar courses
  - Craig Gotsman course on DGP: http://www.cs.technion.ac.il/~cs236329
  - Alla Sheffer course on Geometric Modeling http://www.ugrad.cs.ubc.ca/~cs424/
  - Siggraph 2007 and Eurographics 2006 course on Geometry Processing using polygonal meshes: http://www.agg.ethz.ch/publications/course_notes

Mesh processing libraries

- CGAL:
  - Computational Geometry Algorithms Library (Linux, Windows) http://www.cgal.org
- OpenMesh
  - Efficient Half-Edge structures for polygonal meshes http://www.openmesh.org
- MeshMaker
  - http://www.cs.ubc.ca/~sheffa/dgp/software/MeshMaker5.2.zip
- Graphite:
  - Powerful but no documentation: http://alice.loria.fr/software/graphite/

Backup slides



Max

Me

Min

Gaus