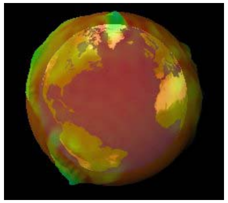
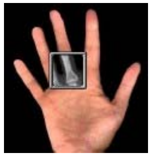


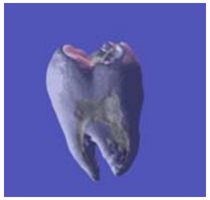
# Computational Visualization

1. Sources, characteristics, representation

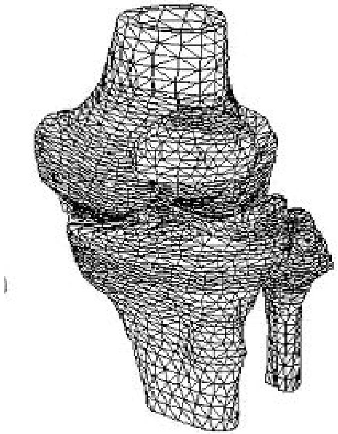
2. Mesh Processing

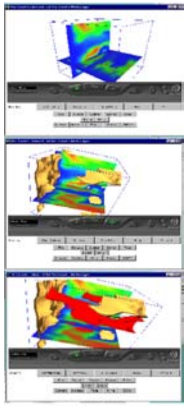


3. Contouring



4. Volume Rendering





5. Flow, Vector, Tensor Field Visualization
6. Application Case Studies

# Computational Visualization:

## Mesh Processing

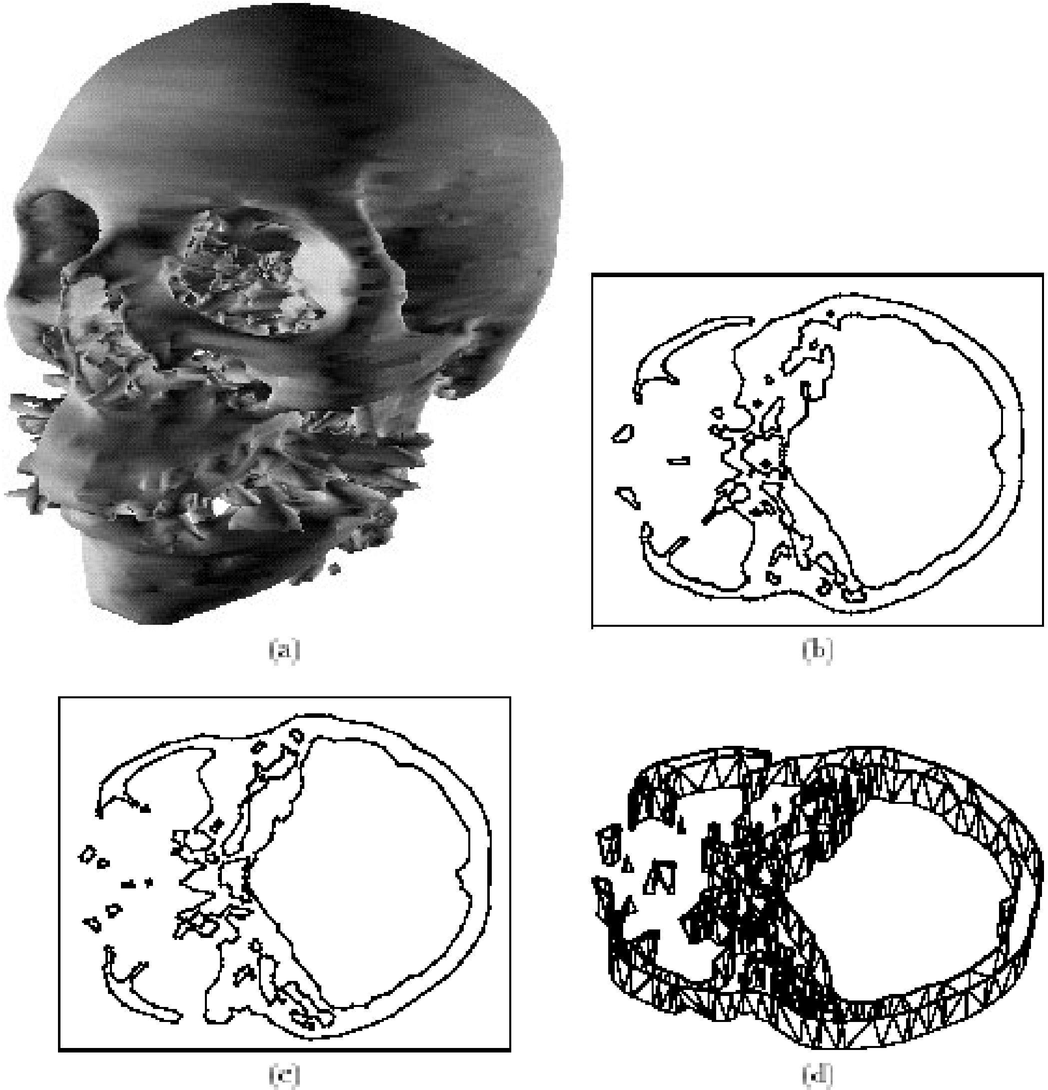
Lecture 2

# Outline

- Triangle and Tetrahedral Meshing
- Hexahedral Meshing
- Filtering (Anistropic Diffusion)

Center Computational Visualization

# Meshing I

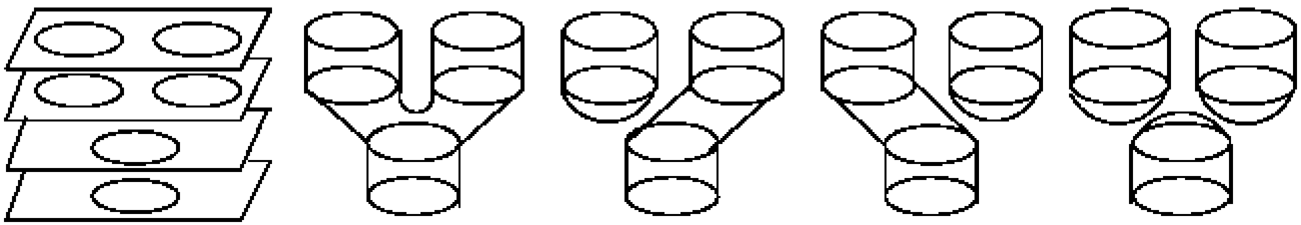



- To generate a boundary element triangular mesh from a set of cross-section polygonal slice data.

- Subproblems
  - The correspondence problem
  - The tiling problem
  - The branching problem

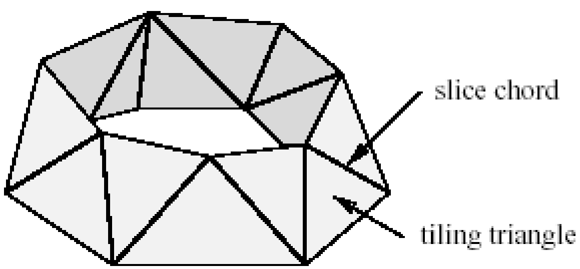
# Sub-problems

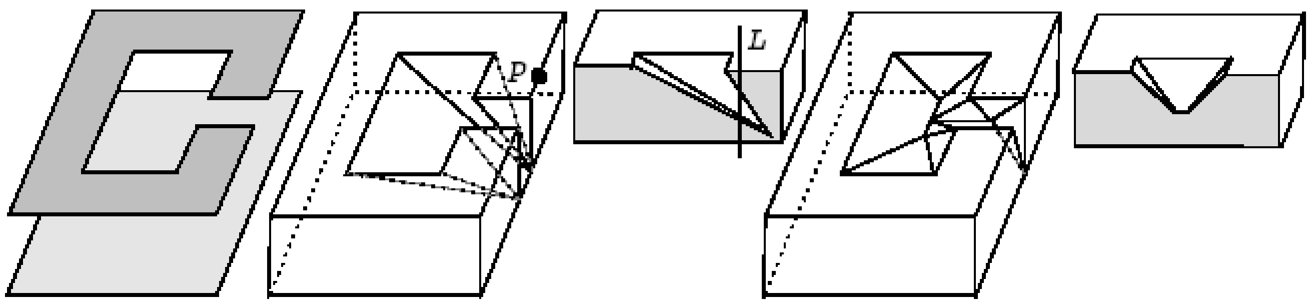
- Correspondence



(a) (b) (c) (d) (e)

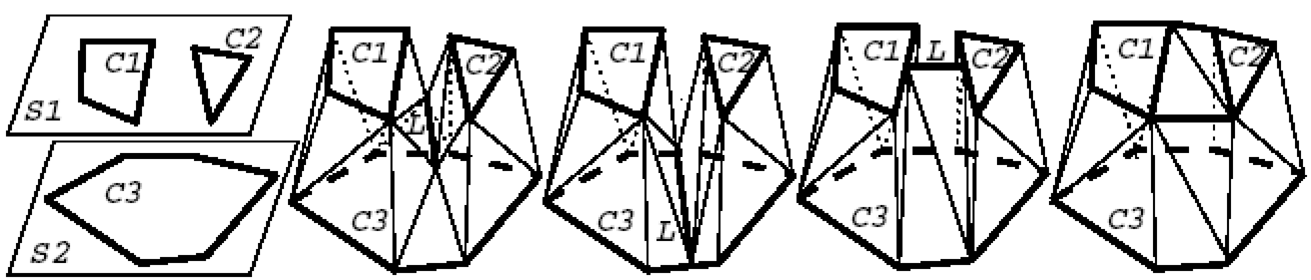
- Tiling




(a) (b) (c) (d) (e)

- Branching




(a) (b) (c) (d) (e)

# Incremental Construction

## Algorithm Steps

Step 1: Form closed contours from image slices.

Step 2: Create any required augmented contours.

Step 3: Find correspondences between contours.

Step 4: Form the tiling region of each vertex.

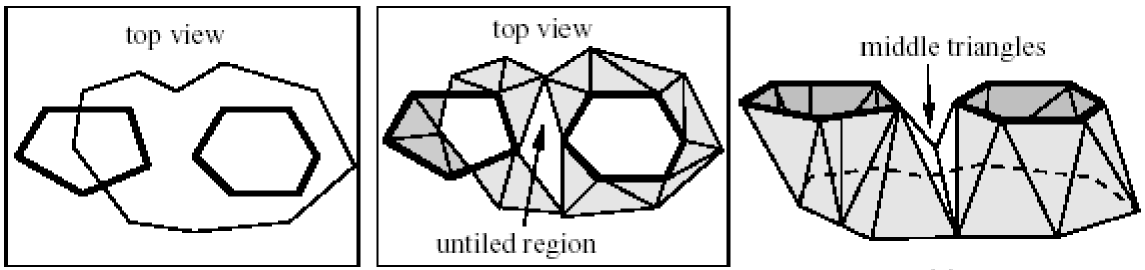
Step 5: Construct the tiling.

Step 6: Collect the boundaries of untiled regions.

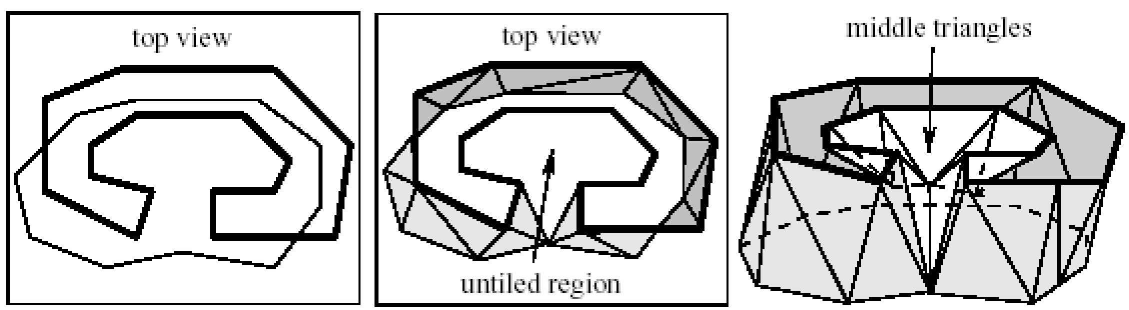
Step 7: Form triangles to cover untiled regions based on their edge Voronoi diagram (EVD).

# Algorithm Steps

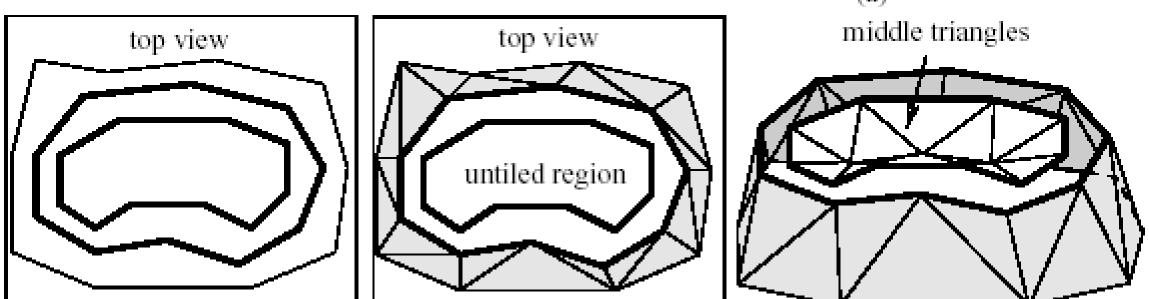
- A multi-pass tiling approach followed by the postprocessing of untiled regions



(a) (b) (c)

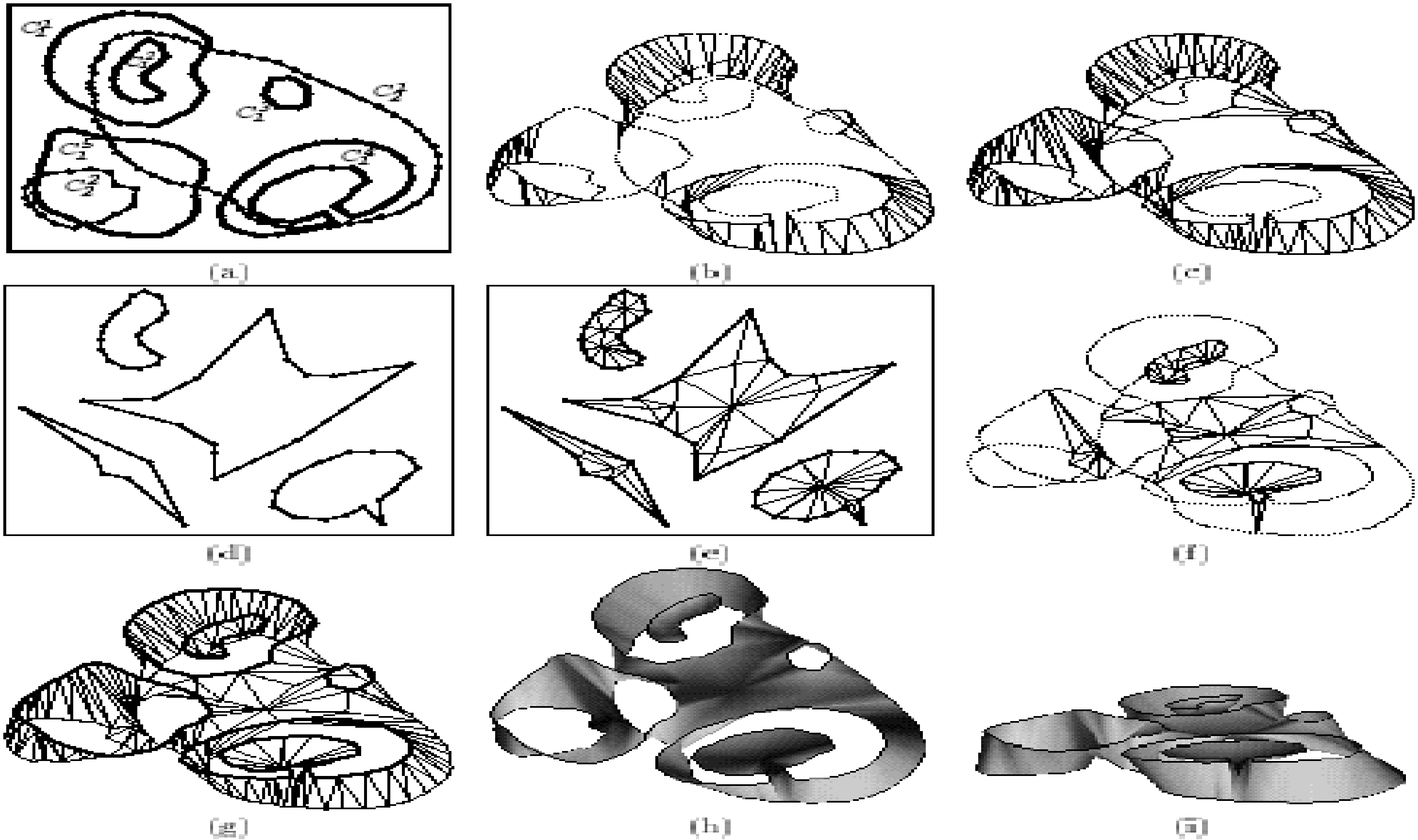


(a) (b) (c)



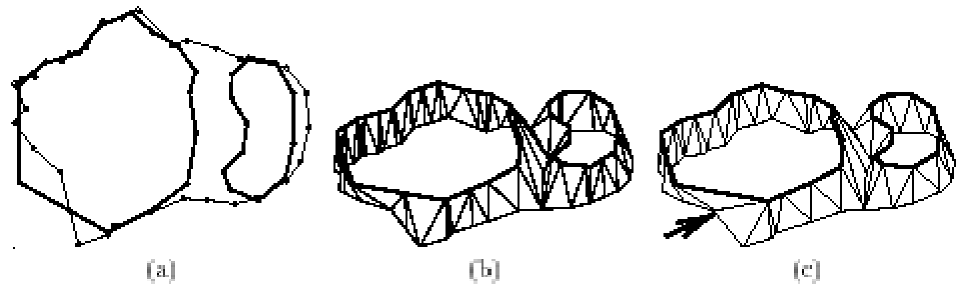
(a) (b) (c)

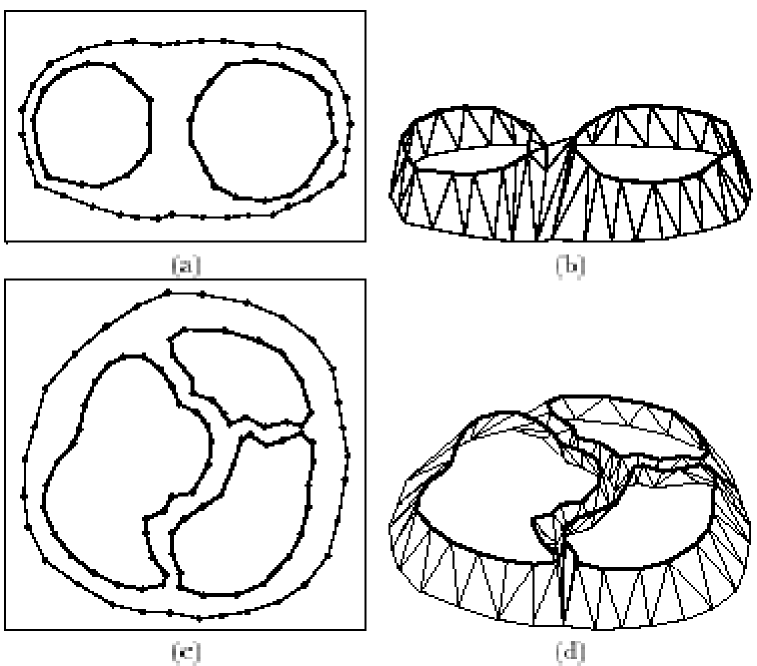
# Algorithm Steps on actual data

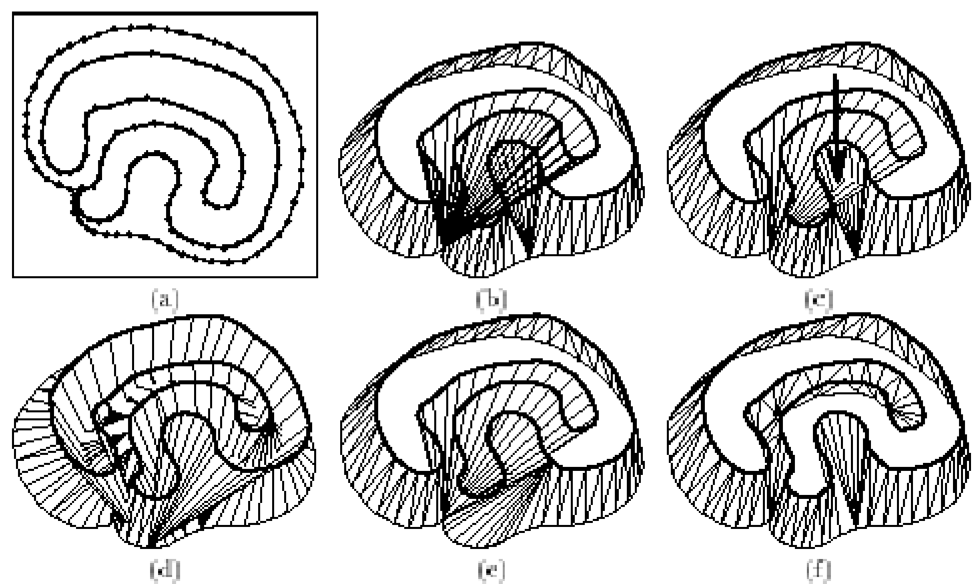


(a) (b) (c) (d) (e) (f) (g) (h) (i)

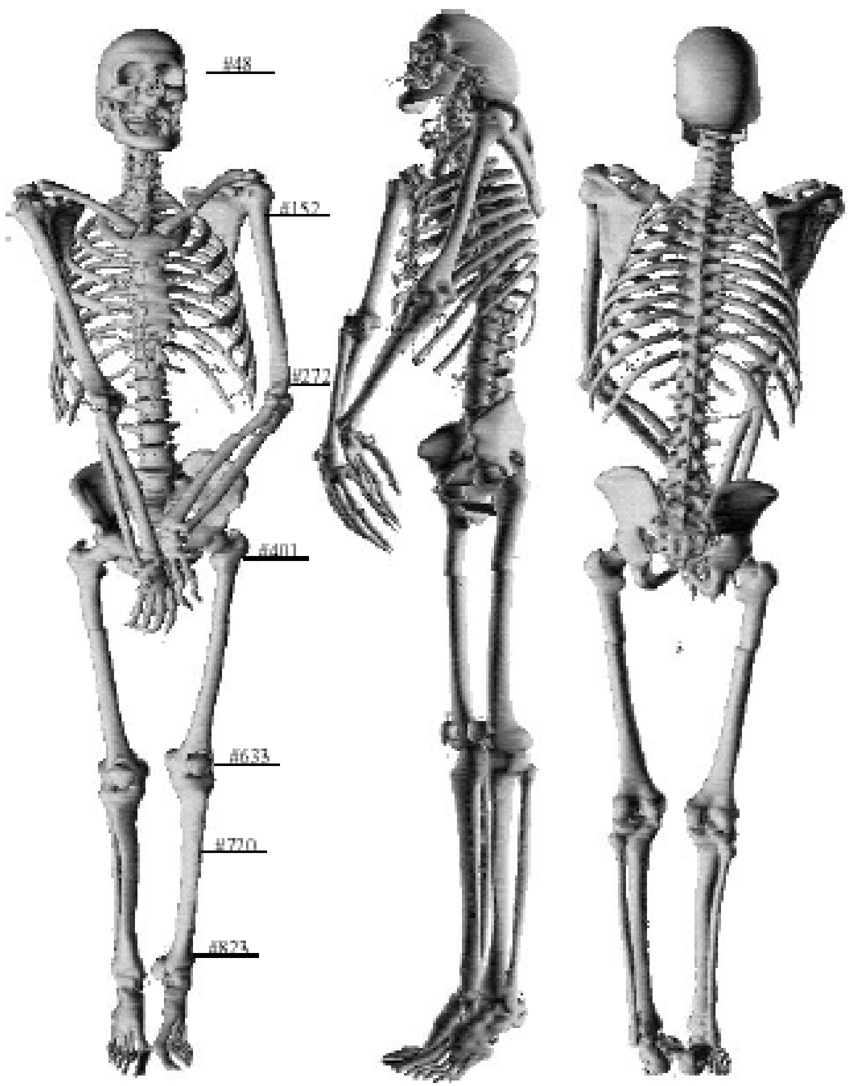
# Using the Edge Voronoi Diagram as Ridges

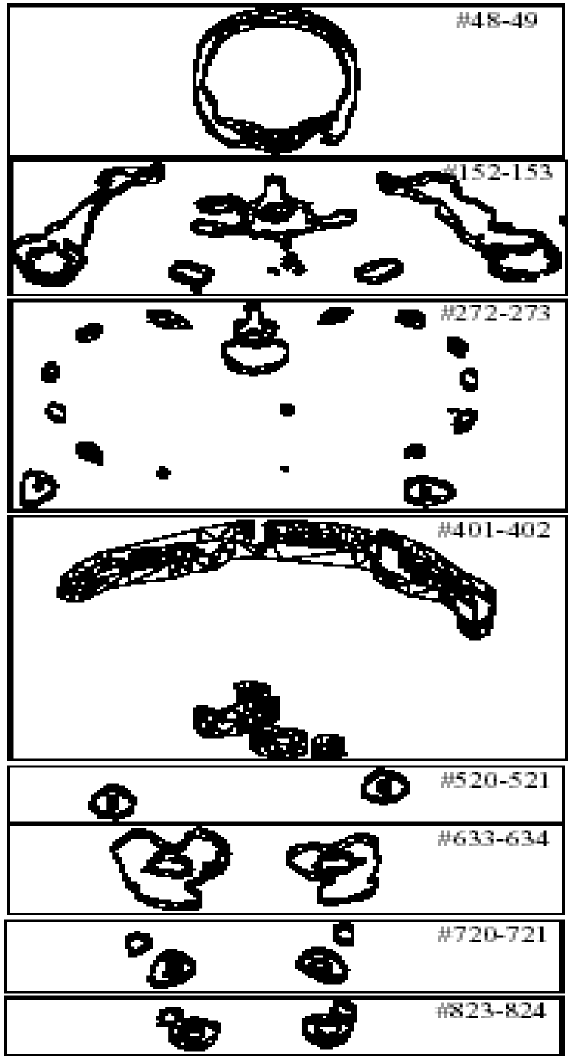


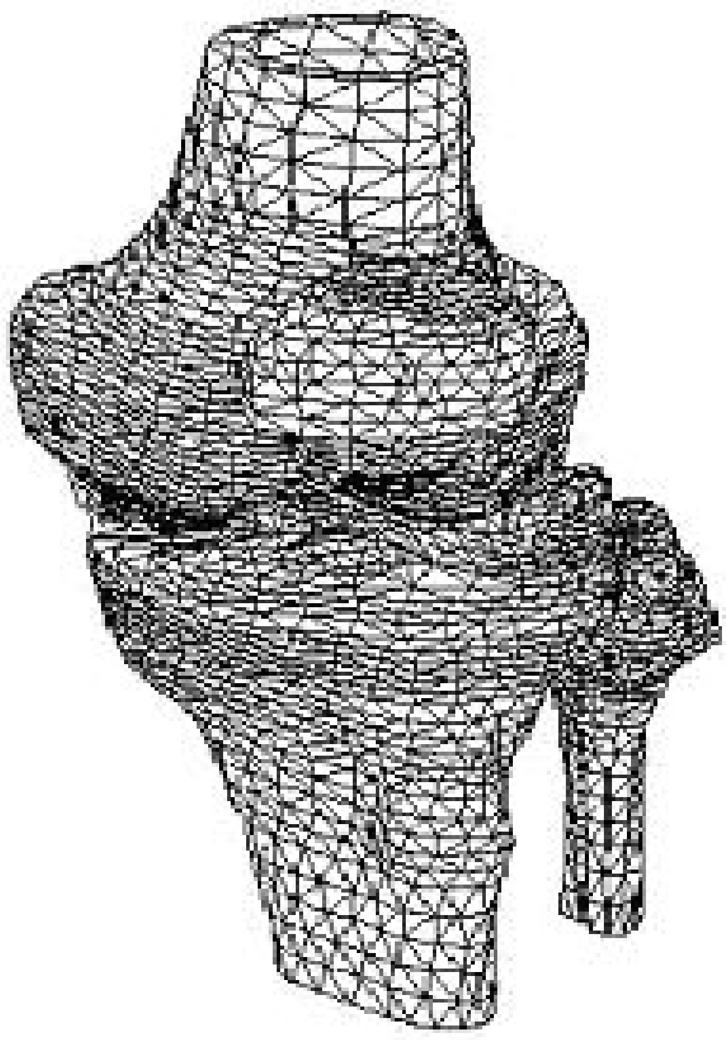



# Boundary Element Triangular Mesh



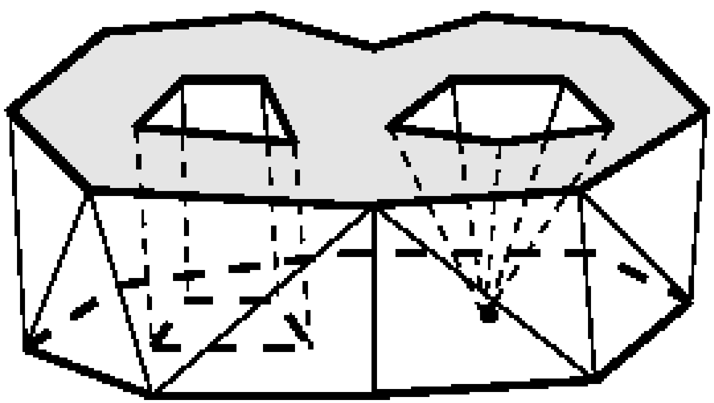
# Meshing II

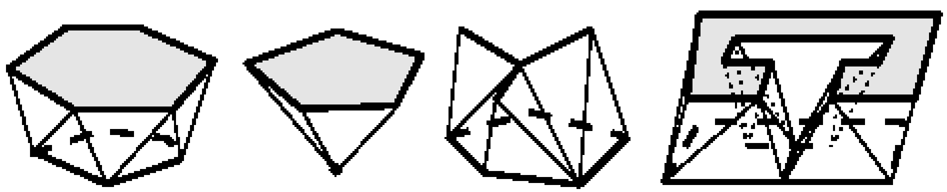


- To generate a 3D finite element tetrahedral mesh of the simplicial polyhedron obtained via the BEM construction of cross-section polygonal slice data.
- Subproblems
  - The shelling of tetrahedra to reduce polyhedron to prismatoids
  - The tetrahedralization of prismatoids

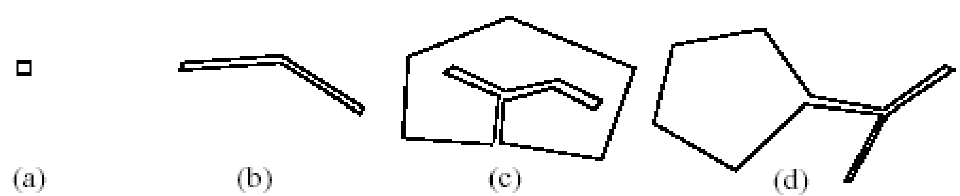
# What is prismatoid?

A prismatoid is a polyhedron having for bases two polygons in parallel planes, and for lateral faces triangles or trapezoids with one side lying in one base, and the opposite vertex or side lying in the other base, of the polyhedron.





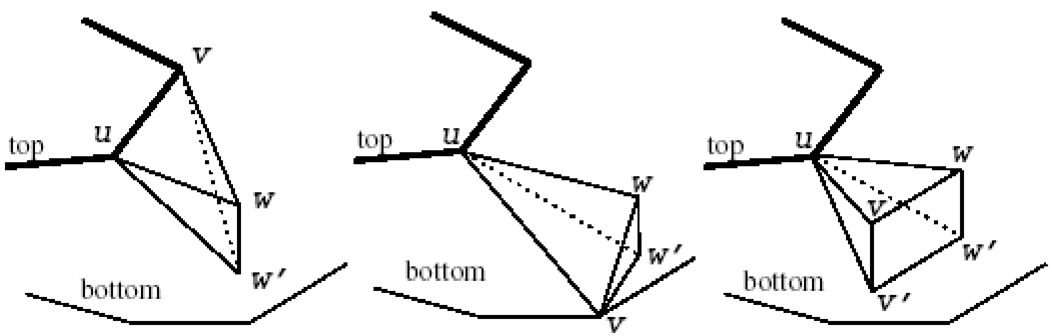
(a) (b) (c) (d)



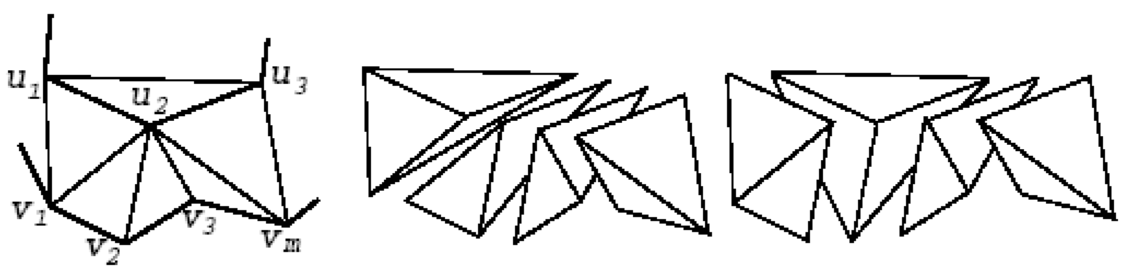
(a) (b) (c) (d)

# The Shelling Step

- Shell tetrahedra from the polyhedron, so the remaining part is a prismatoid or can be divided into prismatoids.




(a) (b) (c)




(a) (b) (c)

# Prismatoid → Tetrahedra

- To tetrahedralize a non-nested prismatoid without Steiner points.

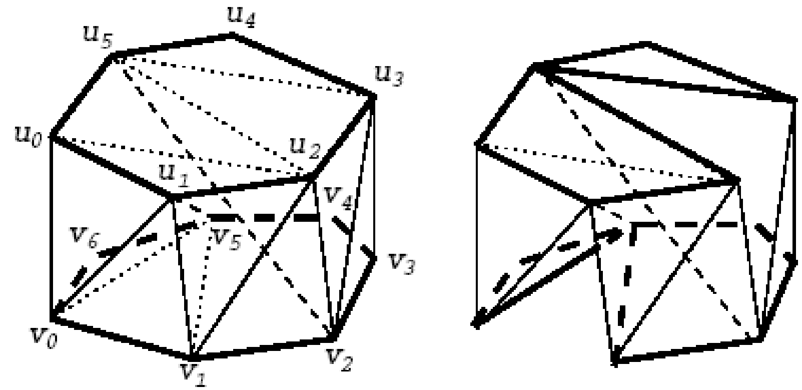
1. For each boundary triangle on both slices, calculate its metric.
2. Pick up the boundary triangle with the best metric and form one set of tetrahedra.
3. Update the advancing front and go to Step 1.
4. If the remaining part is non-tetrahedralizable, postprocess it.

Center Computational Visualization ccv

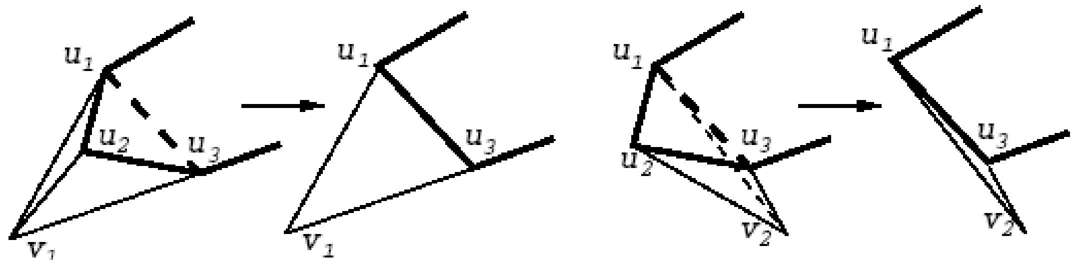
# Metric, Weight Factor, Grouping

- Metric = volume/(edge)$^3$
- Weight factor

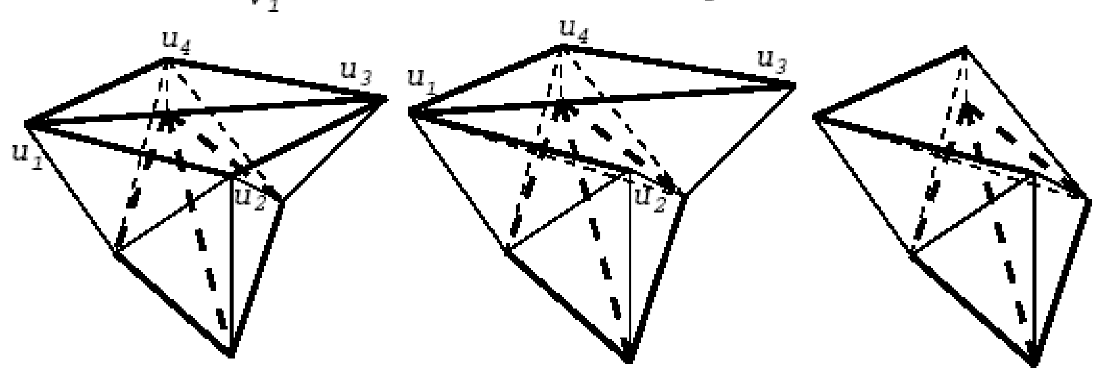
$$w = \begin{cases} 2(1 - \frac{d}{h}) & \text{if } d \le 0.5h \\ 1 & \text{if } 0.5h < d < h \\ \frac{h}{d} & \text{if } d \ge h \end{cases}$$




(a) (b)




- Grouping can avoid irregular remaining part

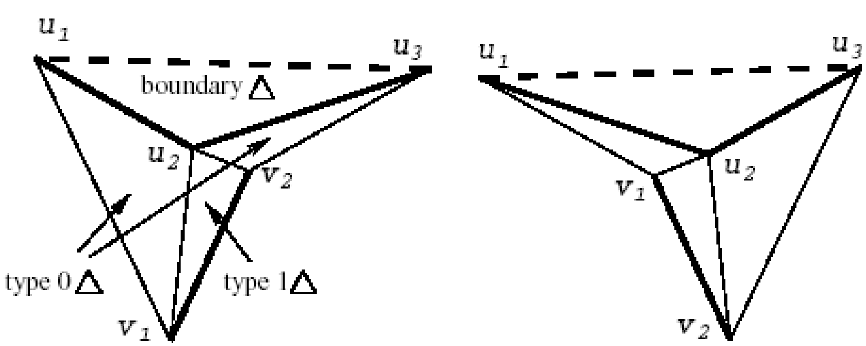



(a) (b) (c)

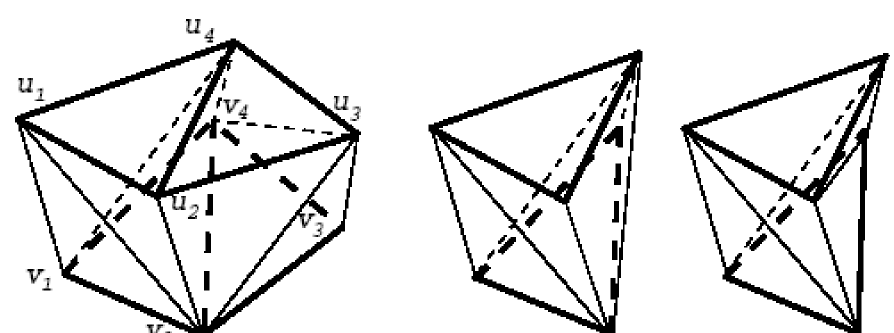
# Protection Rule

Lemma 1: Suppose a top boundary triangle $\Delta u_1u_2u_3$ is under the constraint that no more than one type 1 triangle is between the two type 0 triangles containing the contour segments $u_1u_2$ and $u_2\,u_3$. Furthermore, let the bottom vertices of the two type 0 triangles be $v_1$ and $v_2$. Our grouping operation cannot apply to $\Delta u_1u_2u_3$ to form a set of tetrahedra, if and only if all the following conditions are satisfied.

1. $v_1v_2$ is exactly one contour segment.
2. One of the slice chords $u_2v_1$ and $u_2v_2$ is reflex and the other is convex.
3. Both $u_1v_2$ and $u_3v_1$ are not inside the prismatoid.



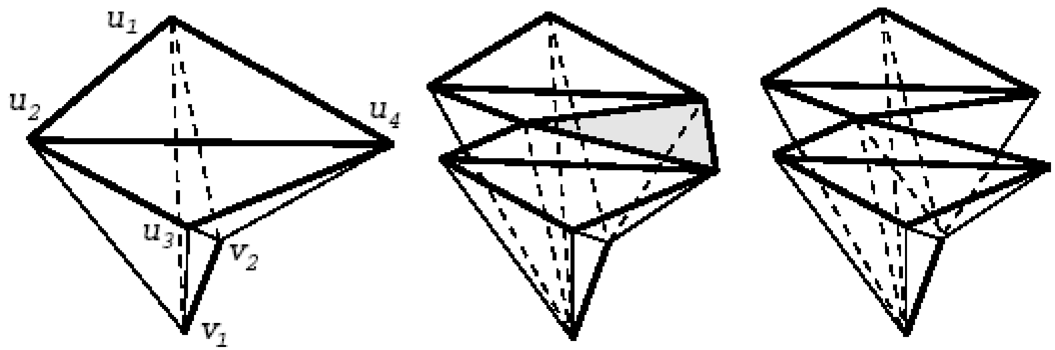
(a) (b)



(a) (b) (c)

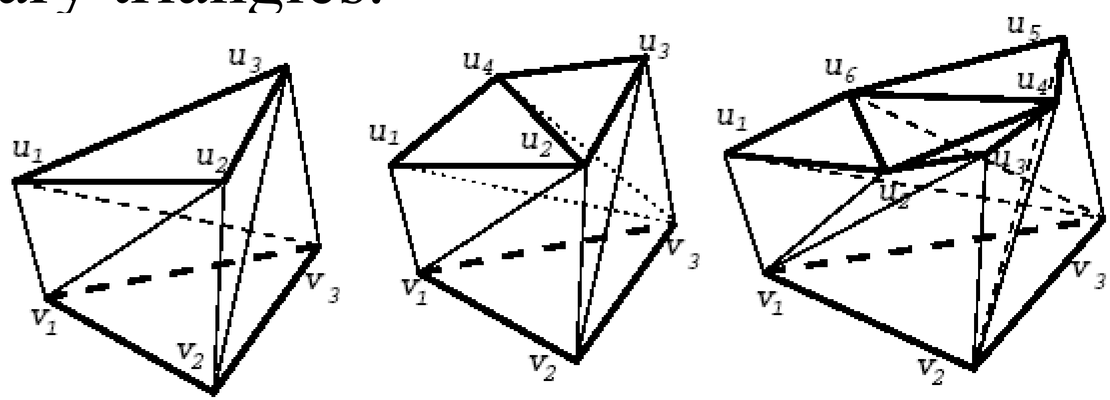
# Classification of Untetrahedralizable Prismatoids

1. Has two boundary triangles on the top face and one line segment on the bottom face.



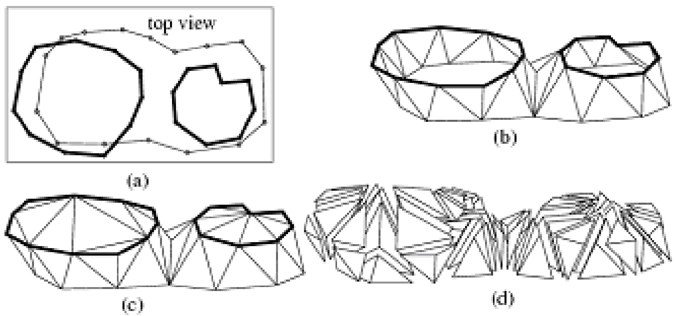

(a) (b) (c)

2. Has one bottom triangle which is treated as three boundary triangles.

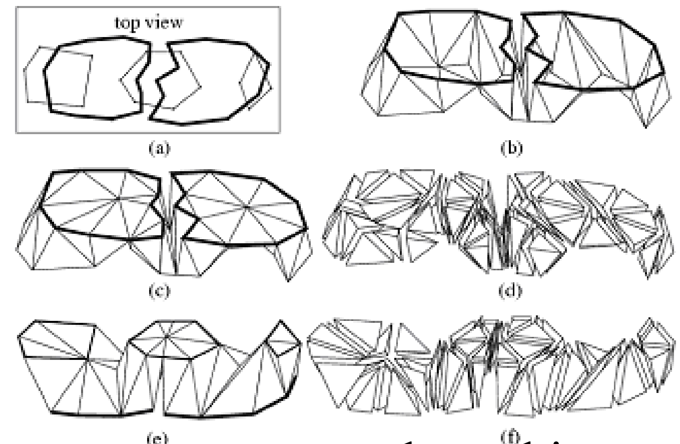



(a) (b) (c)

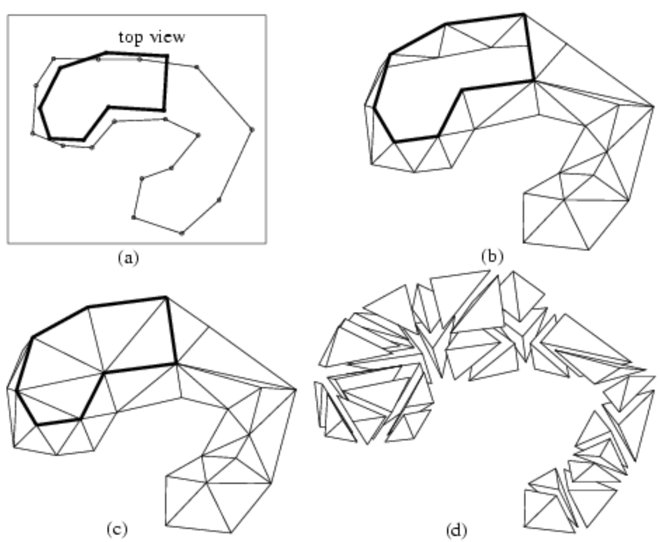
# Multiple Tetrahedralizable Cases



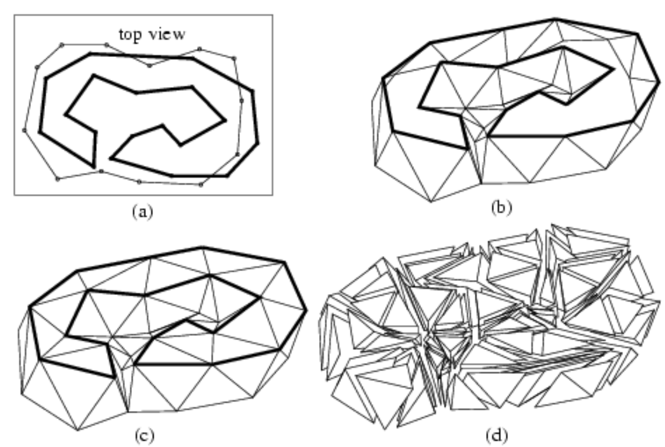

One-to-many branching




many-to-many branching

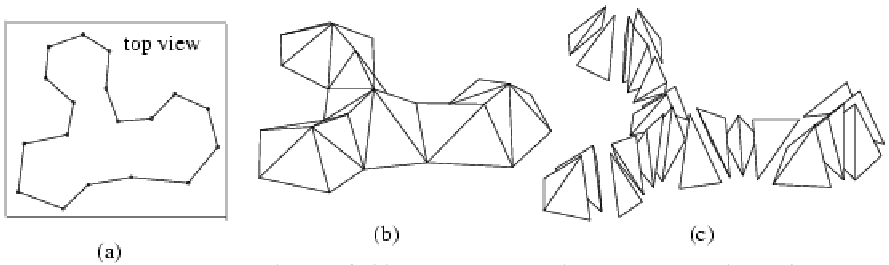



Dissimilar region (the right bottom portion of the bottom contour)

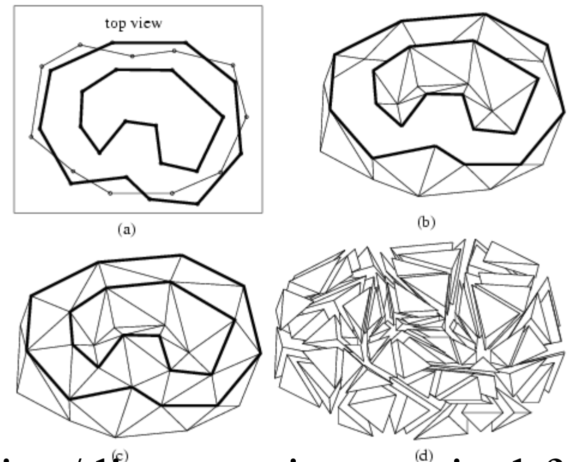



Dissimilar region (the inner portion of the top contour)

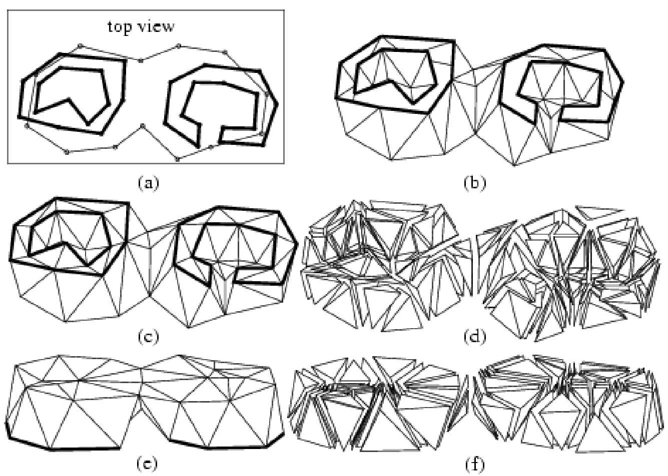
# Multiple Tetrahedralizable Cases



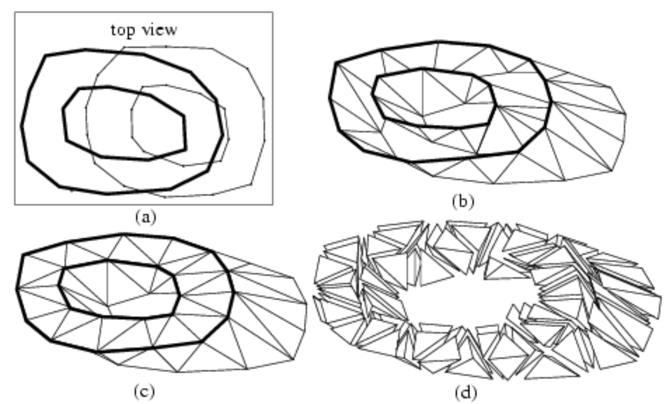

Appearing/disappearing vertical feature of a solid interior




Appearing/disappearing vertical feature (the top inner contour) of a void interior

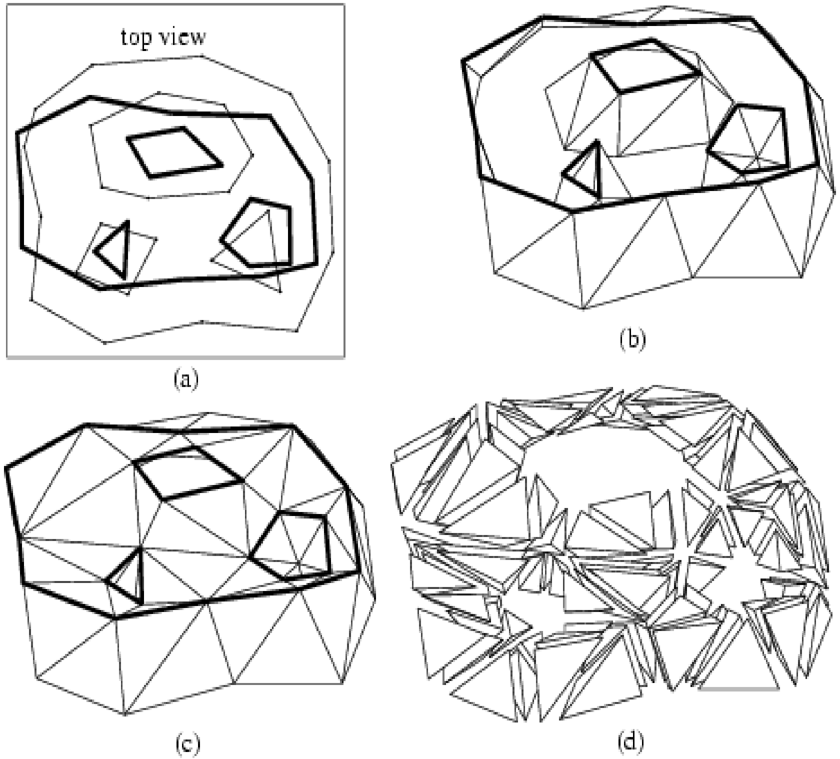



A branching, a dissimilar portion (the inner portion of the top right contour), and an appearing/disappearing vertical feature (the inner contour at the left of the top slice)

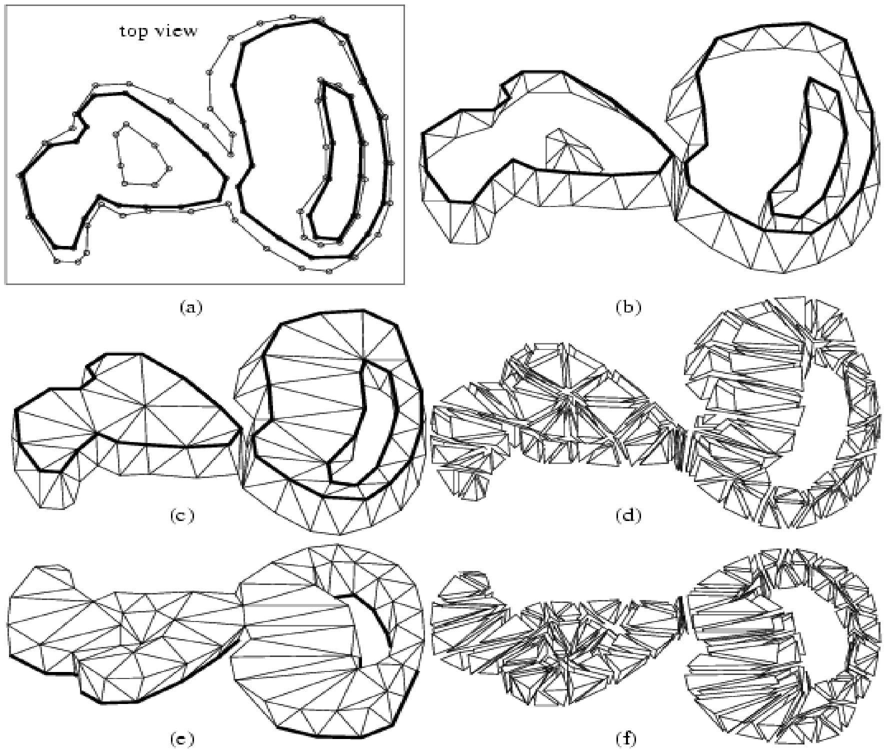



Nested prismatoids

# Multiple Tetrahedralizable Cases

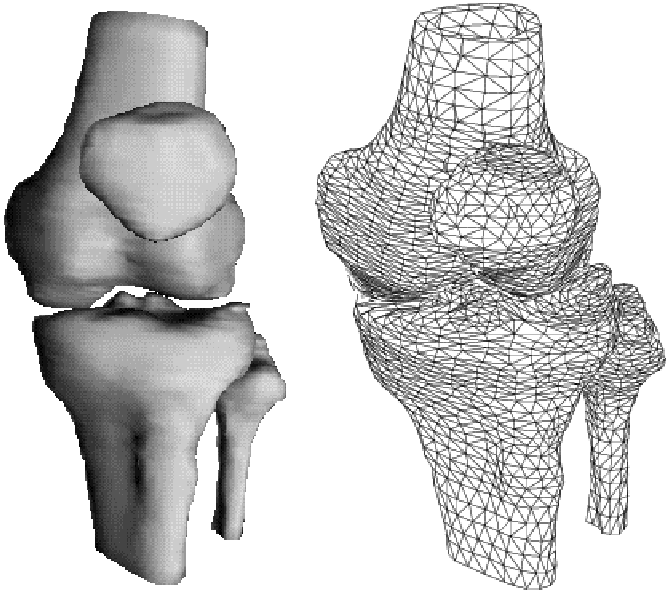



Multiply-nested prismatoid




Solid region between two slices of a human tibia

# Examples

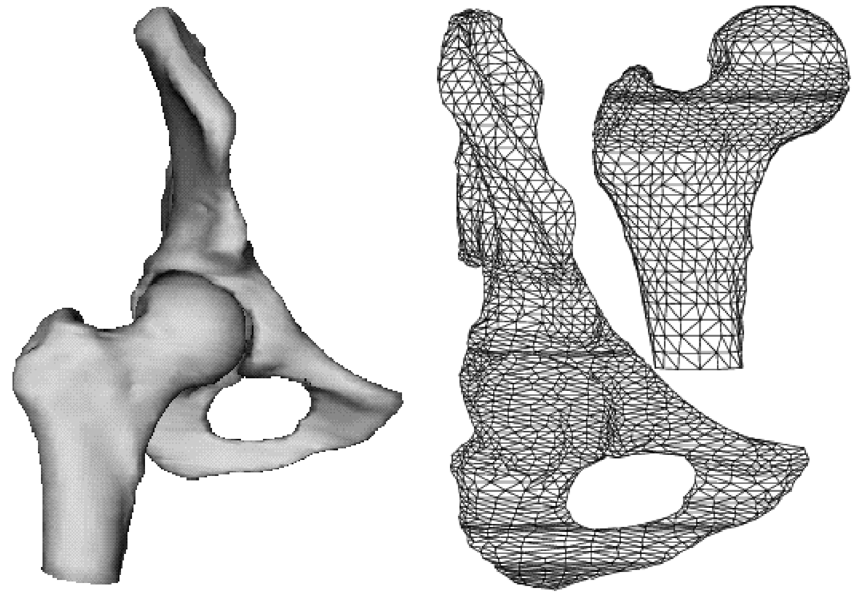


(a) (b)

Knee joint (the lower femur, the pper tibia and fibula and the patella)

(a) Gouraud shaded

(b) The tetrahedralization



(a) (b)

Hip joint (the upper femur and the pelvic joint)

(a) Gouraud shaded

(b) The tetrahedralization

# Mini-summary

- The characterization, avoidance of non-tetrahedralizable polyhedra is one of the main challenges
- The mix of numerical precision and topological decision making needs precise rules so errors don't propagate.

# Further reading


[1] C. Bajaj, E. Coyle, K. Lin. Arbitrary topology shape reconstruction from planar cross sections. *Graphical Models and Image Processing*, 58(6):524-543, Nov.1996.

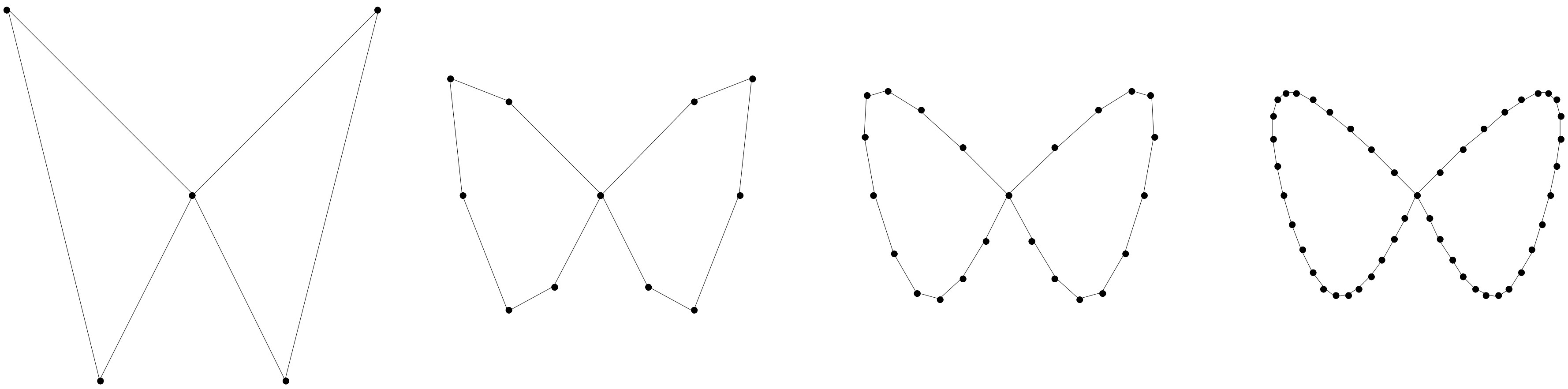
[2] C. Bajaj, T. Dey, Convex Decompositions of Polyhedra and Robustness. *Siam Journal on Computing*, 21, 2, (1992), 339-364.

[3] MEYERS, D., Multiresolution Tiling. *Computer Graphics Forum* 13, 5 (December 1994), 325--340.

[4] C. Bajaj, E. Coyle, K. Lin. Tetrahedral meshes from planar cross sections. *Computer Methods in Applied Mechanics and Engineering*, Vol. 179 (1999) 31-52

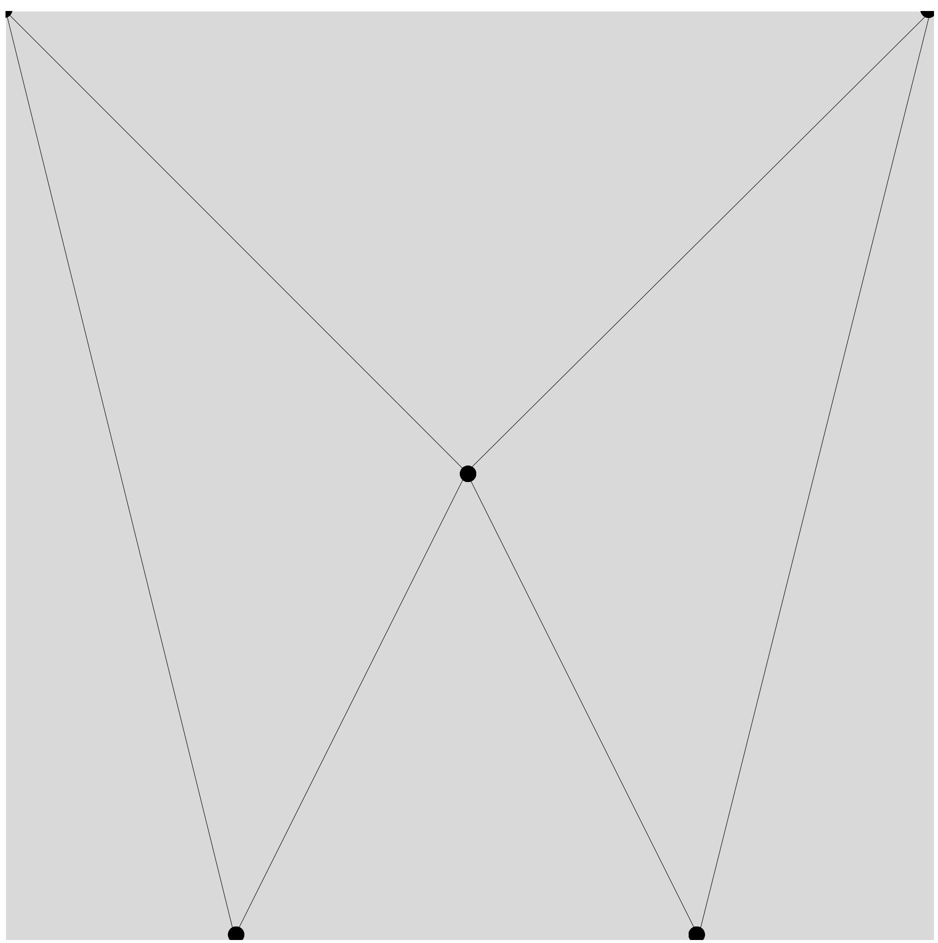
# Univariate Subdivision

## Subdivision Curves

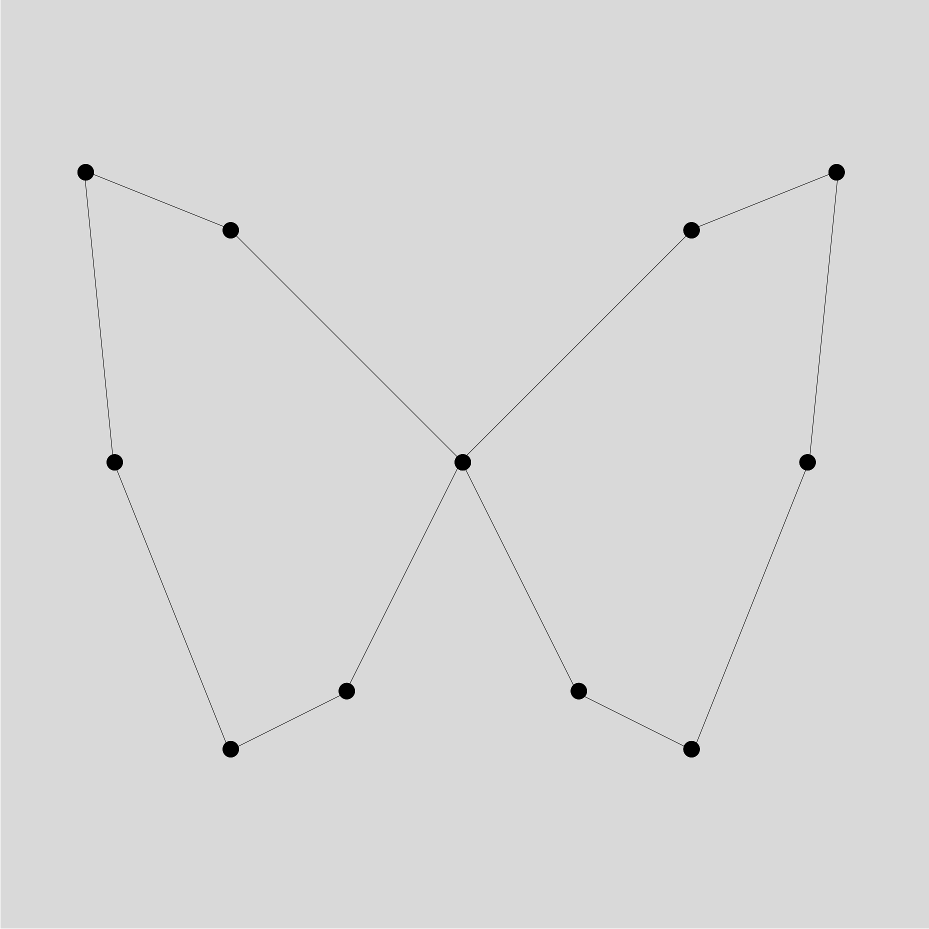


## Butterfly curve

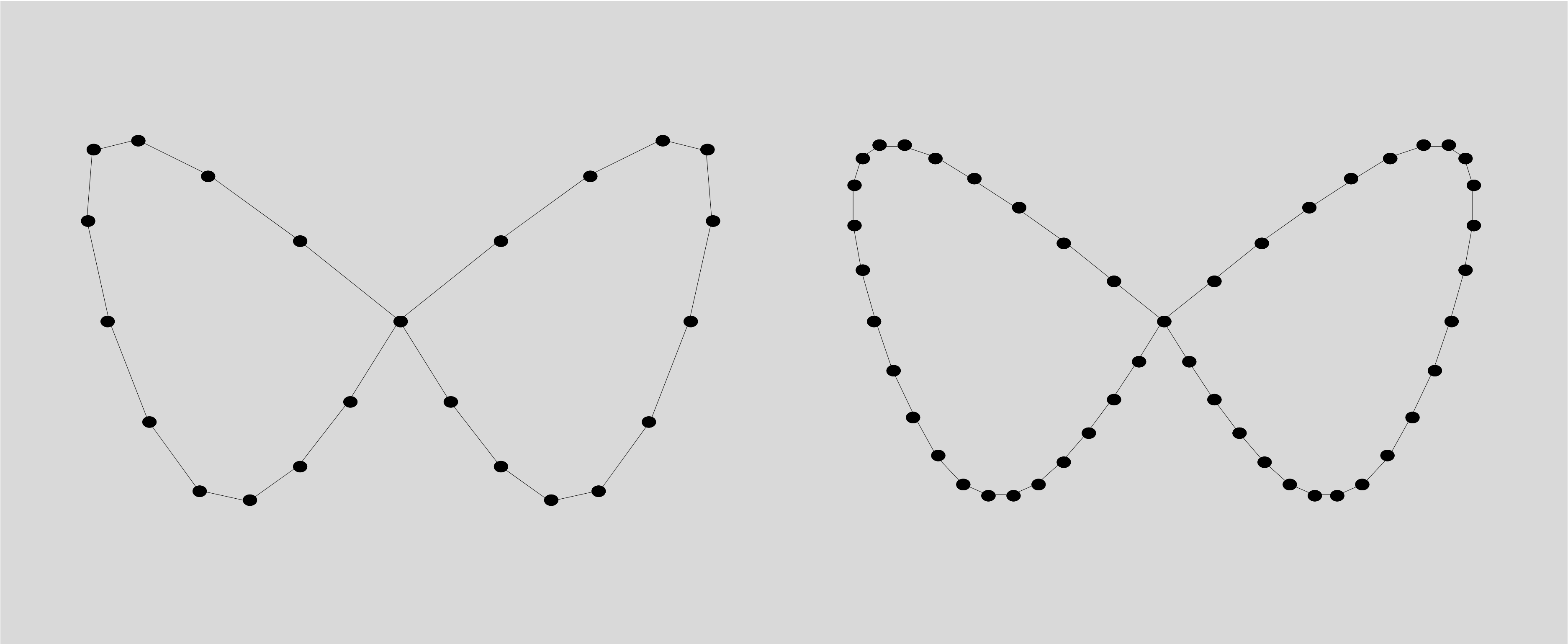
# Subdivision



# Subdivision

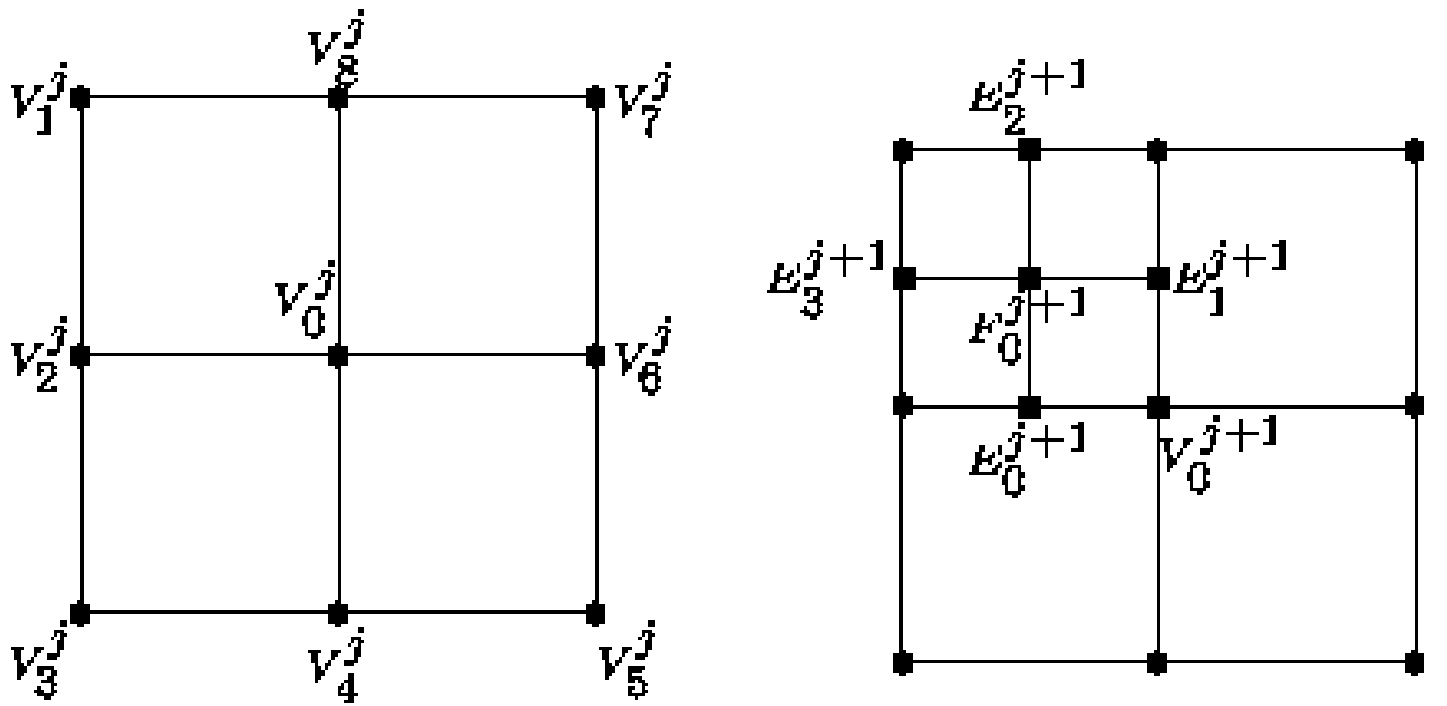


# Subdivision



# Bivariate Subdivision

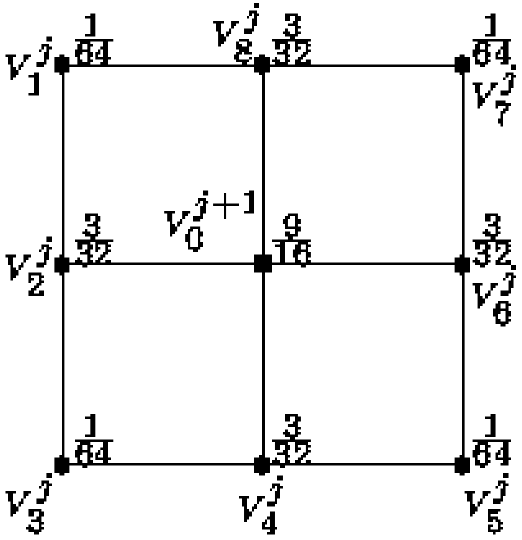
## Catmull Clark Subdivision[1]:



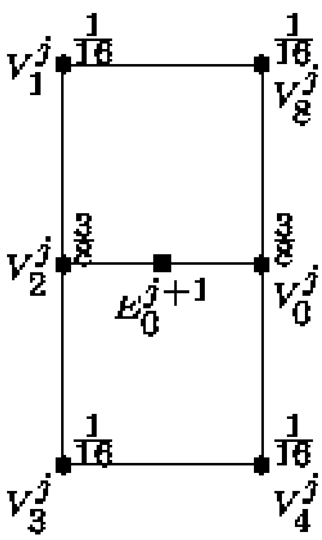

(contd.)

# Bivariariate Subdivision (contd)

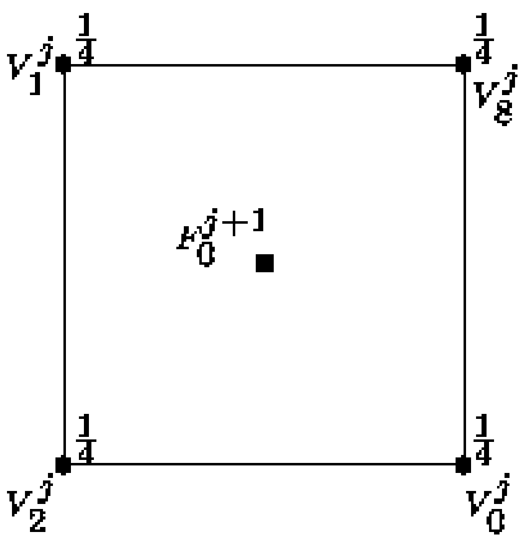
## Vertex Rule




## Edge Rule

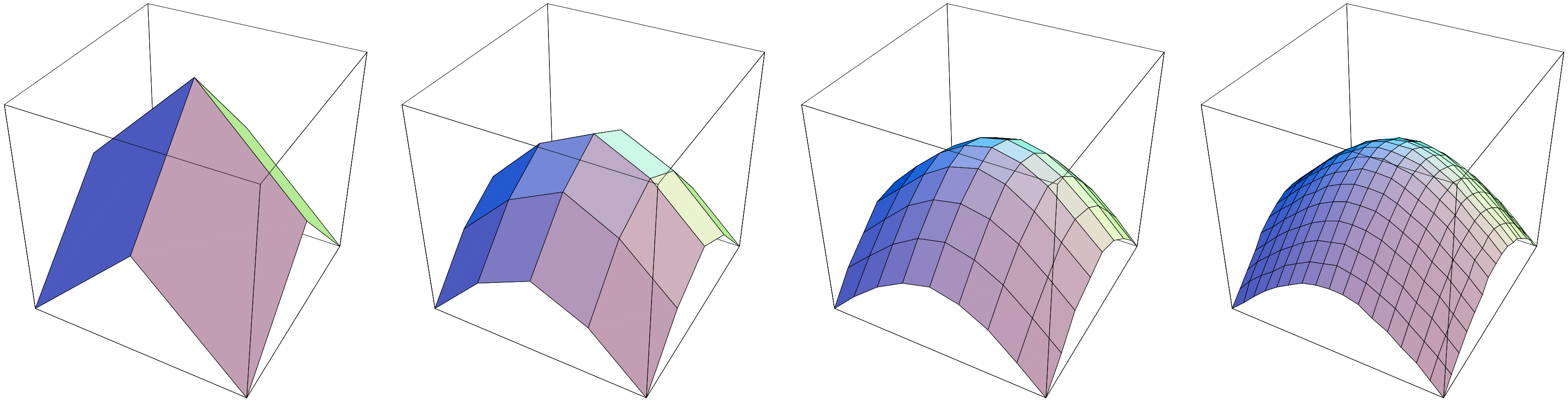



## Face Rule



# Surface Subdivision

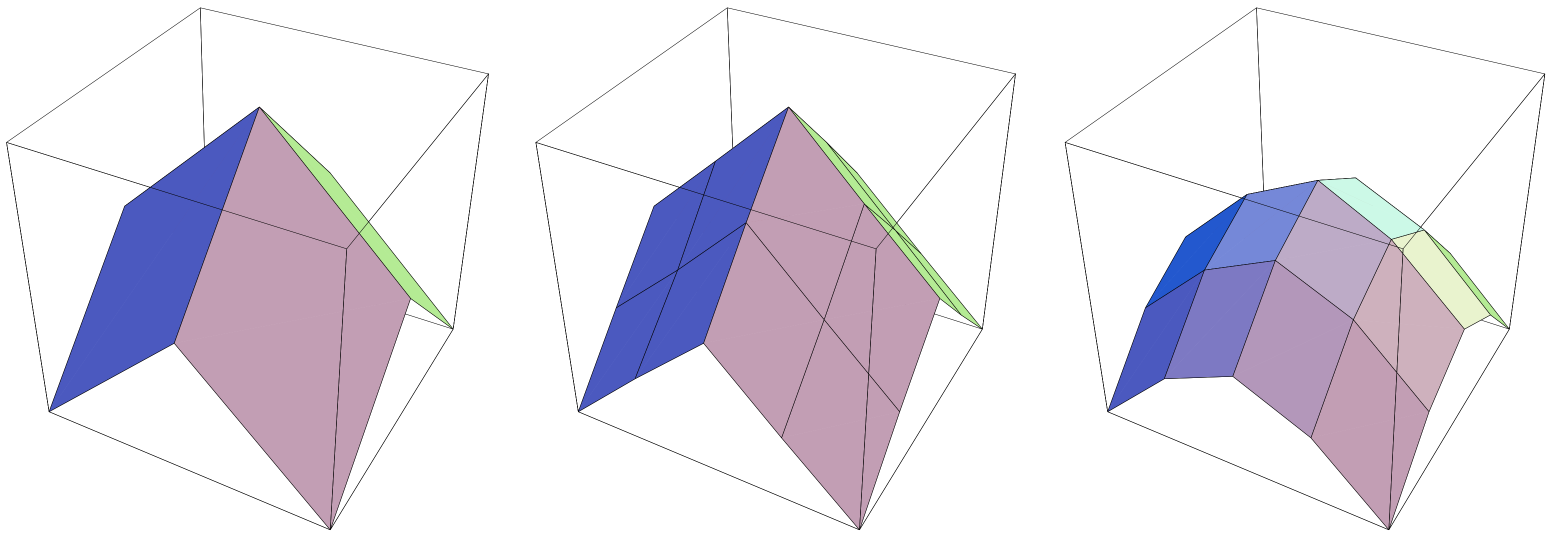
Limit surface is $C^2$



Tensor product cubic B-spline

# Bivariate Subdivision

- Alternate formulation of Catmull Clark:



- Bilinear subdivision plus smoothing

# Univariate Subdivision

The subdivision rule for cubic B-splines can be expressed as linear subdivision followed by smoothing with the mask $\left(\frac{1}{4} \quad \frac{1}{2} \quad \frac{1}{4}\right)$.

Geometric interpretation of mask: reposition a vertex as the midpoint of the midpoints of the two segments that contain the vertex.

# Bivariate Subdivision

Bi-cubic subdivision is equivalent to Bi-linear subdivision followed by smoothing with the tensor product of the univariate mask with itself, i.e.

$$\begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{18} & \frac{1}{8} & \frac{1}{16} \end{pmatrix}.$$

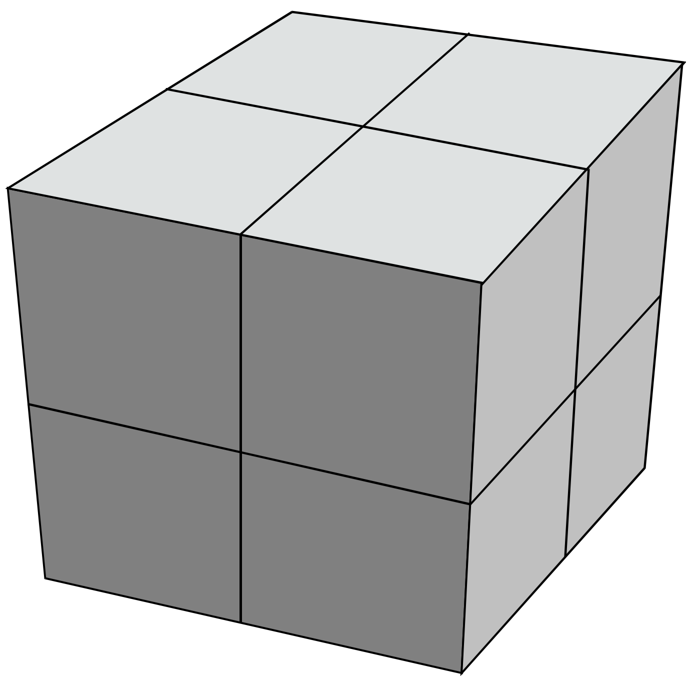
# Trivariate Subdivision

Centroid smoothing: Given a vertex v, compute the centroids of the topological d-cubes that contain v. Reposition v at the centroid of these centroids.

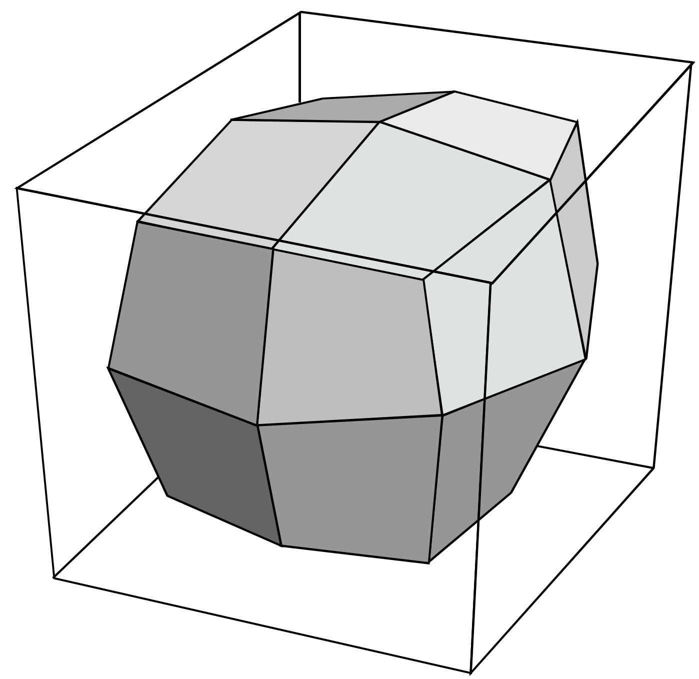
# Multivariate Subdivision

## Generalization to MLCS:

Multi linear Interpolation

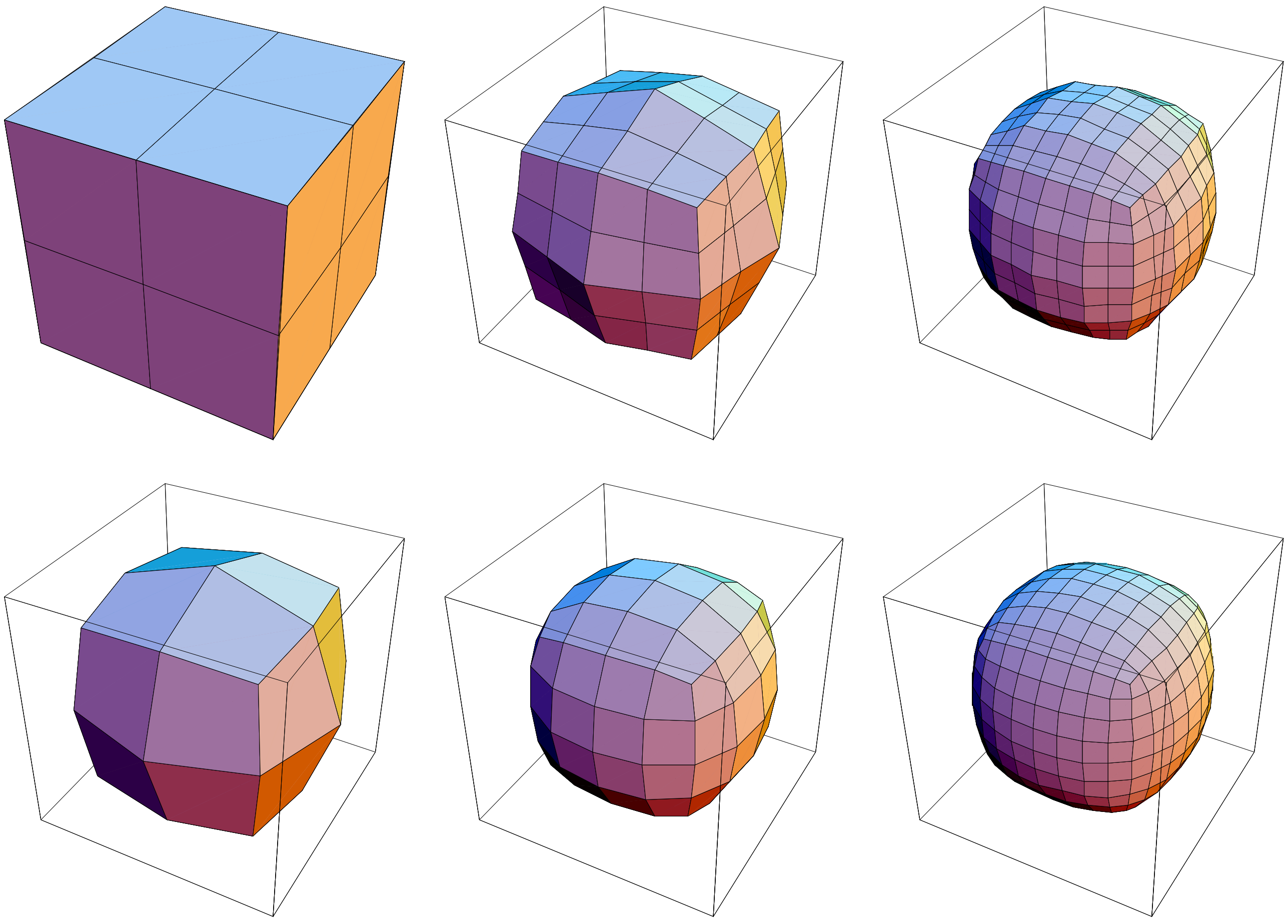


Centroid smoothing



# Subdivision

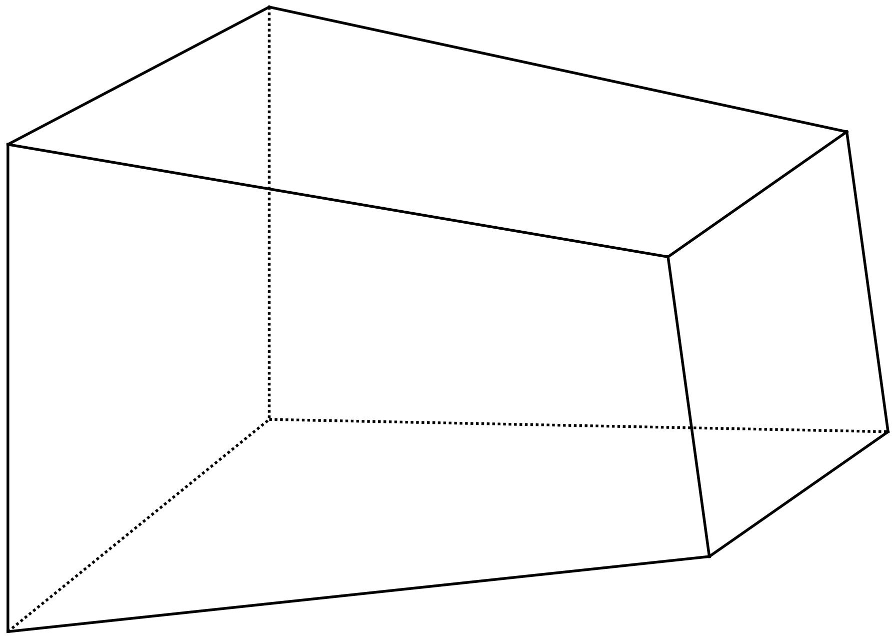
## MLCS Subdivision of a cube:



# Subdivision

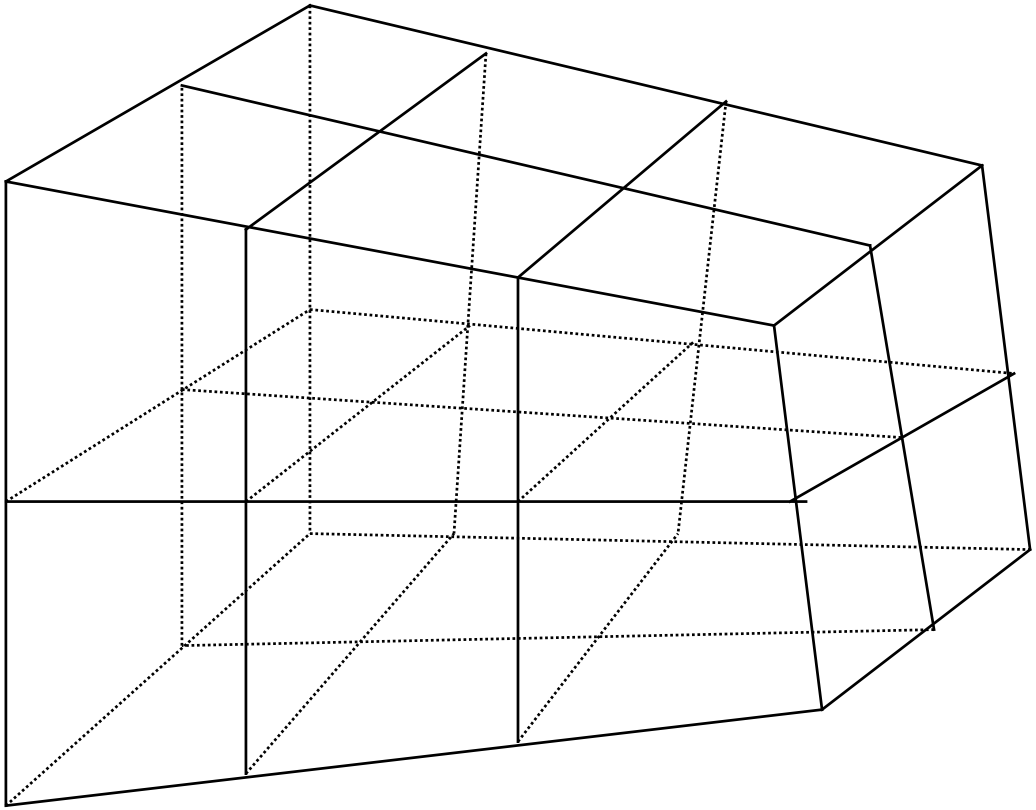
## Hexahedron:

- The hexahedron is a polyhedron with 6 planar faces.
- A hexahedral mesh consists of only the hexahedra



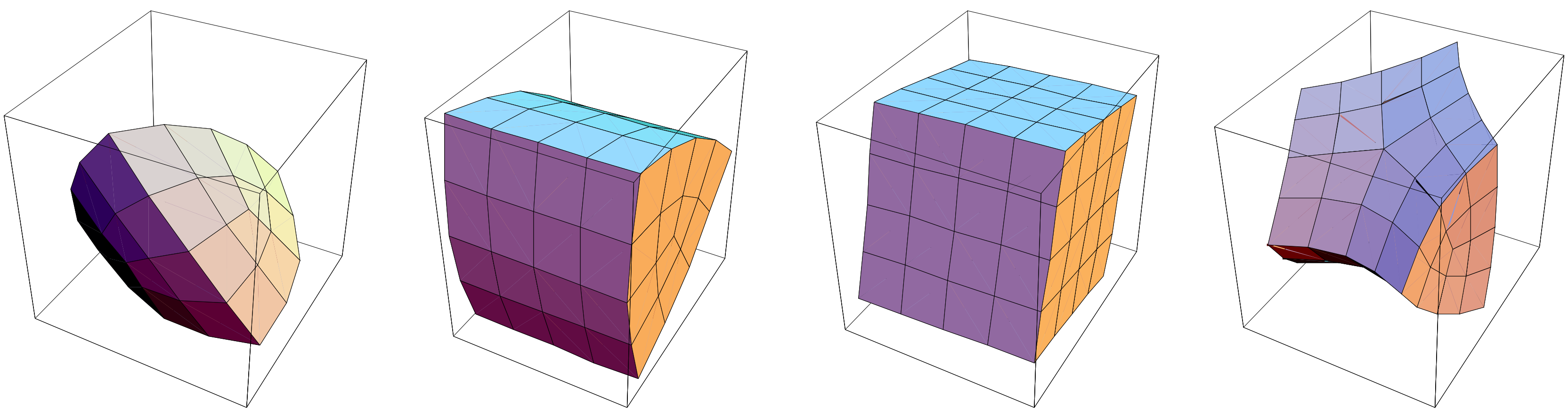
# Subdivision

## Hexahedral mesh with MLCS:



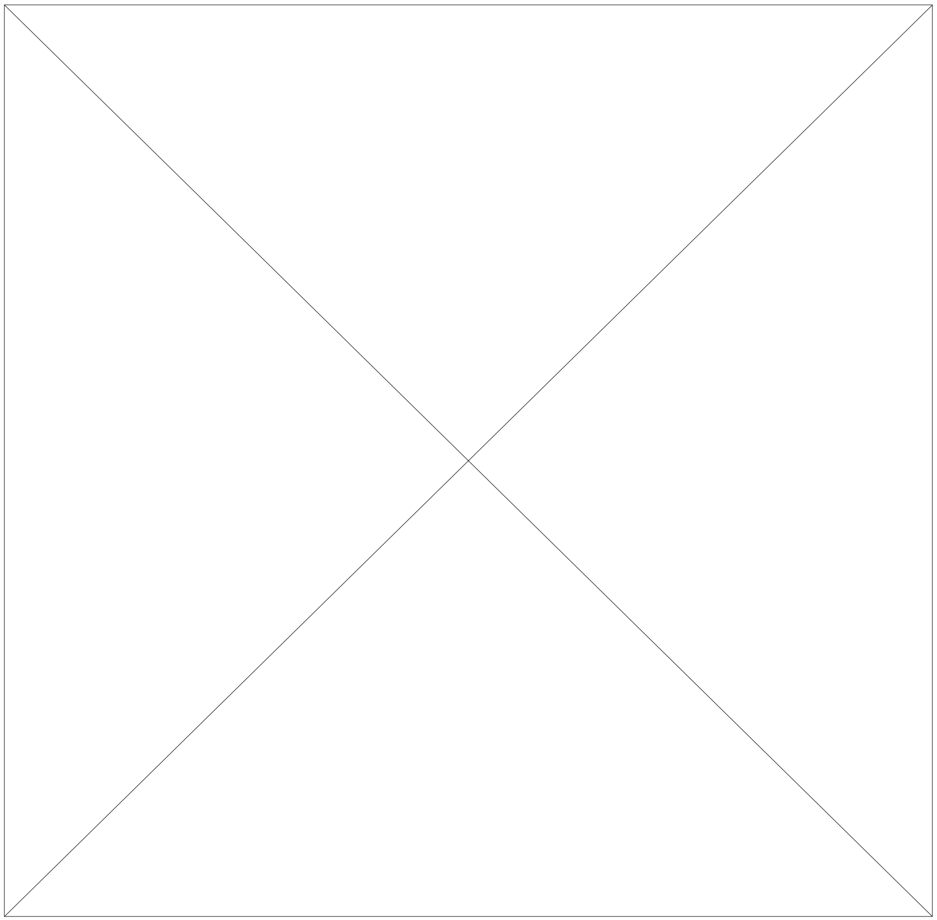
# Subdivision

## MLCS with Creases:

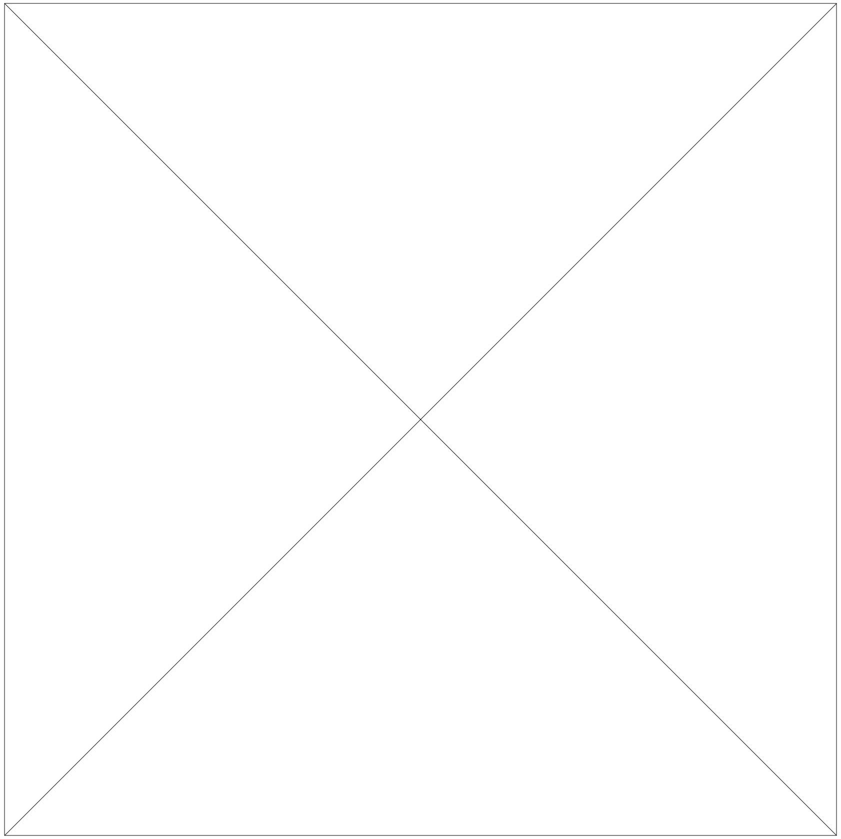


Bring on the coffee!  Or
Mineral Wasser !!

# Filtering Meshes and Function Fields on Meshes



Surface Mesh



Function on surface mesh

# Filtering Problem

Given a discretized noisy triangular surface mesh $G_d \subset \mathbb{R}^3$ (geometric information) and a discretized noisy function-vector $F_d \subset \mathbb{R}^{k-3}$ .

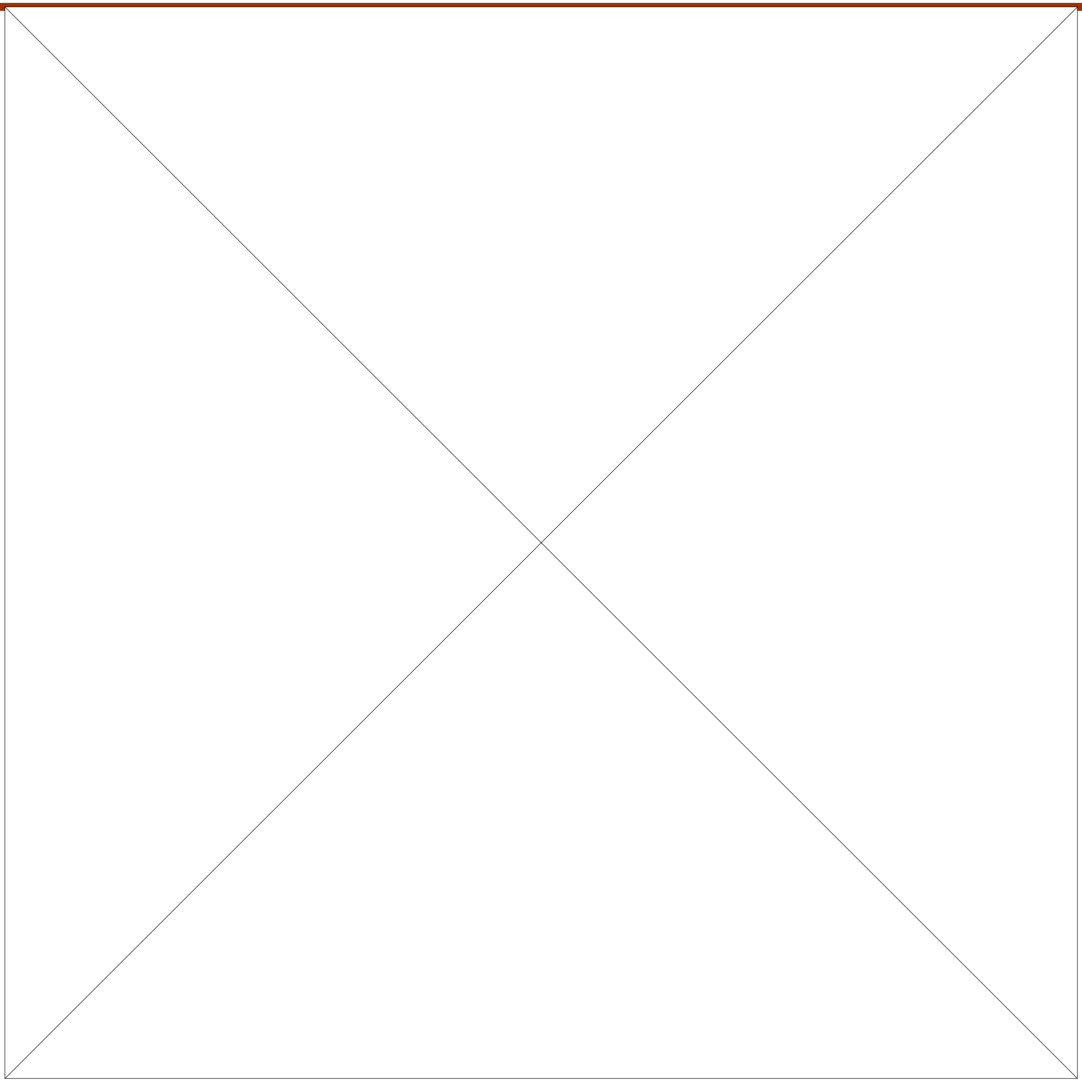
Our goals are :

- Smooth out the noise and to obtain smooth geometry as well as surface function data at different scales.
- Construct continuous (non-discretized) representations for the smoothed geometry and surface function data.
- Provide approaches for visualizing the smoothness of both the geometric and physical information during the smoothing process.

# De-Noising/Fairing Surfaces

Triangle
Surface
Mesh

# Fairing Functions on Surface: Texture Map



# Related work in Image Processing

- Gabor ,1965, PDE based image processing, Jian, 1977, Took off thanks to Koenderink, 1984 Witkin 1983.
- Perona and Malik, 1990, anisotropic diffusion, smoothing and enhancing sharp features.
- Osher and Sethian, 1988, curvature based velocities.
- Mumford and Shah, 1989, PDE based segmentation.
- Terzopoulos et al, 1988, PDE based on active contours for image segmentation.

# Previous Work for Mesh Fairing

1. Optimization

   a. Minimize thin plate energy (Kobbelt 1996, Desbrun, Meyer, Schroder, 1999).

   $$E_p(f) = \int f_{uu}^2 + 2f_{uv}^2 + f_{vv}^2$$

   b. Minimize membrane energy(Kobbelt, 1998 , Desbrun, Meyer, Schroder, 1999).

   $$E_m(f) = \int f_u^2 + f_v^2$$

   c. Minimize curvature (Welch, Witkin, 1992).

   $$E_c(S) = \int \kappa_1^2 + \kappa_2^2$$

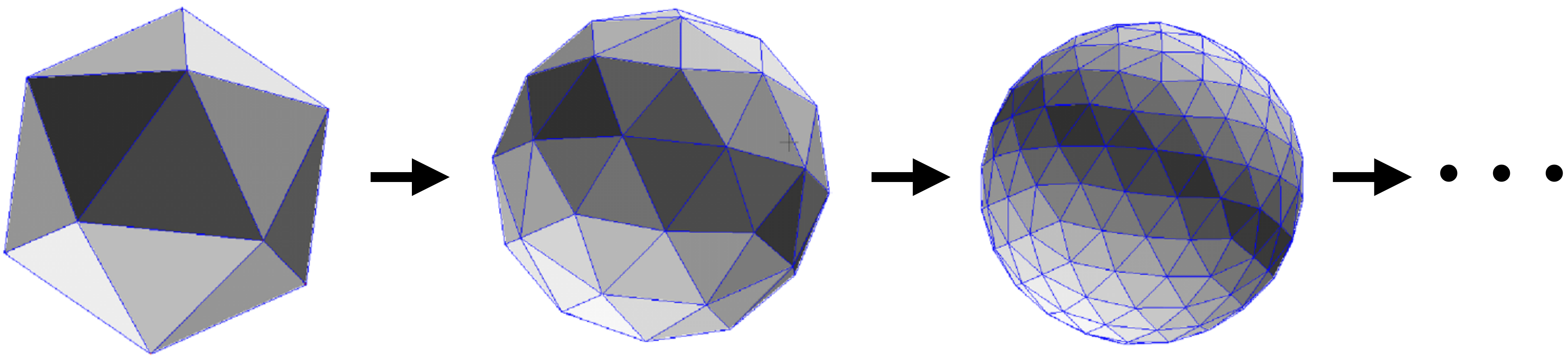
   d. Spring energy( 2000).

2. Signal Processing(Guskov, Sweldens, Schroder,1999; Taubin, 1995) using surface relaxation as low pass filter

$$Rp_i = \sum_{j \in V_2(i)} w_{i,j} p_j$$

where $w_{i,j}$ are chosen to minimize something, e.g. the dihedral angles.

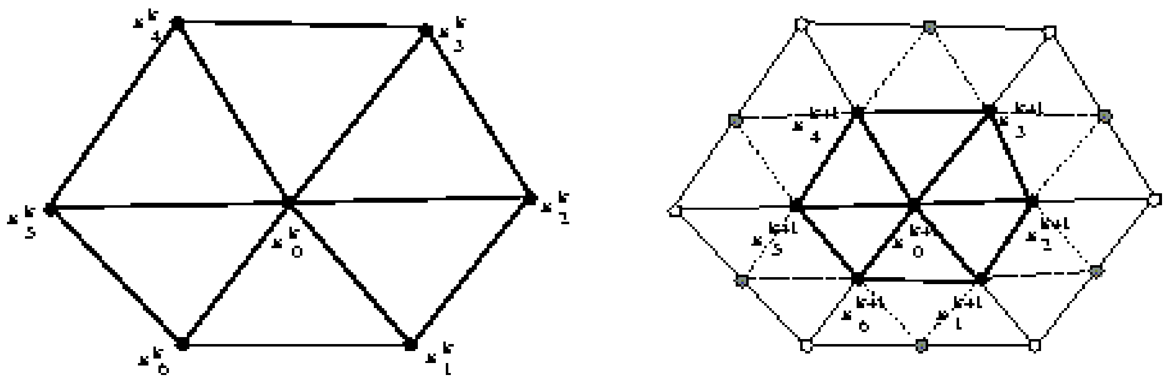
# Loop Subdivision Surfaces



# Loop's Subdivision Surface Scheme

## Edge rule:

$$x_i^{k+1} = \frac{3x_0^k + 3x_i^k + x_{i-1}^k + x_{i+1}^k}{8}, i = 1, \ldots, n,$$



Refinement of a triangular mesh around a vertex

## Vertex rule:

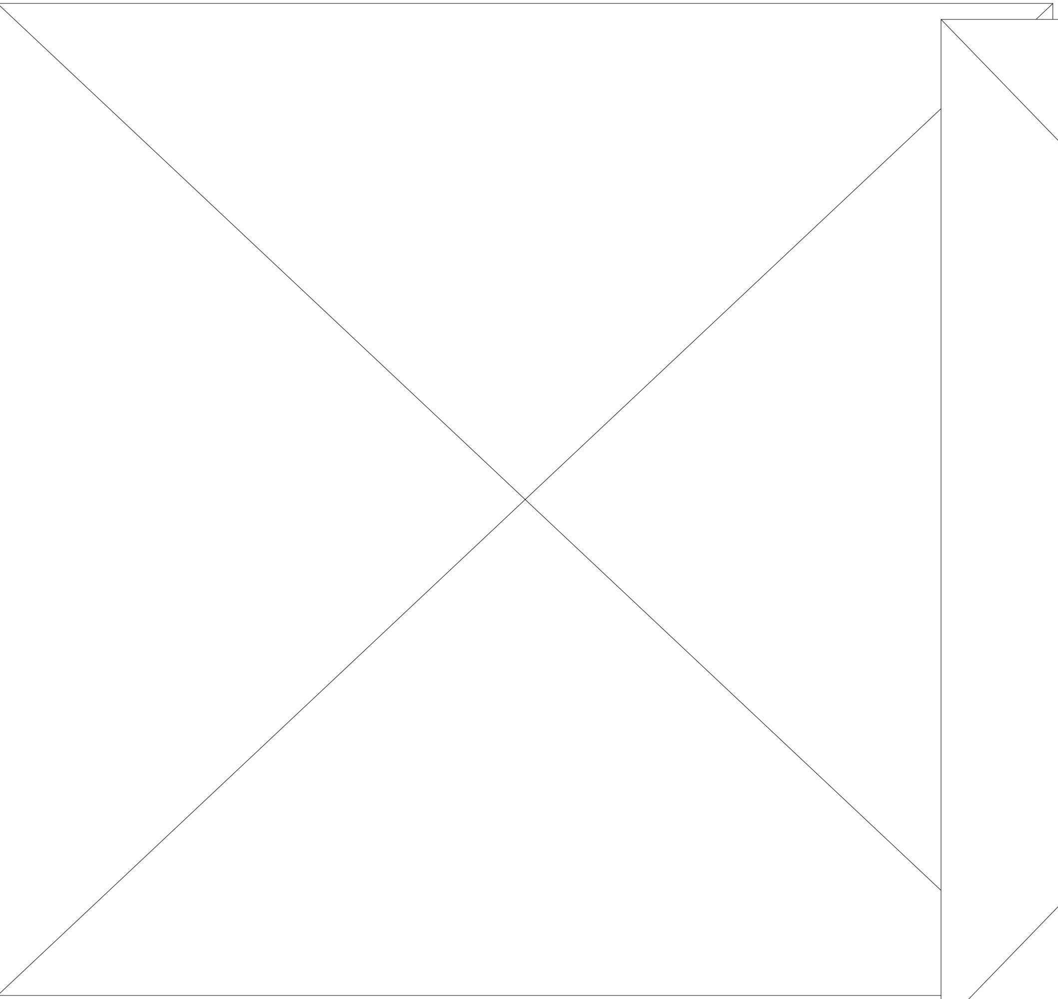
$$x_0^{k+1} = (1 - na)x_0^k + a(x_1^k + x_2^k + \ldots + x_n^k).$$

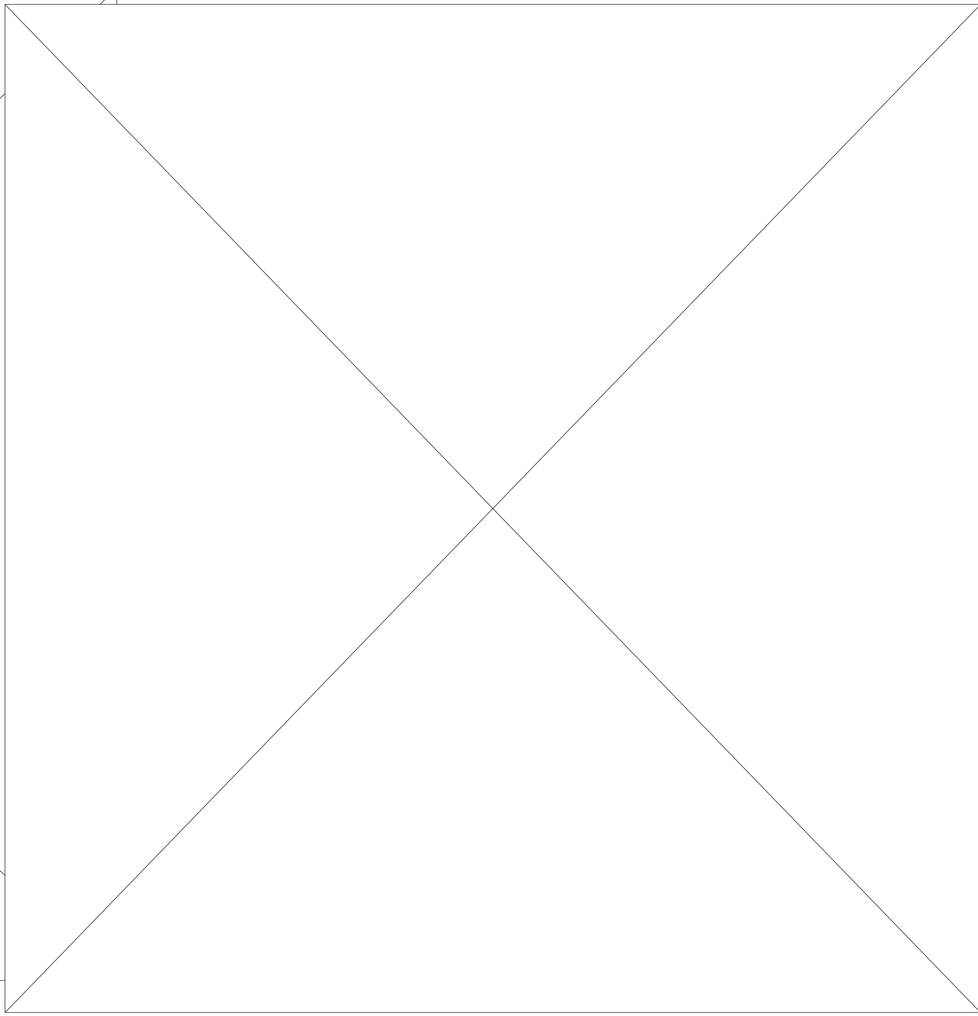
# Filtering by Loop Subdivision Limit Surfaces

Triangulation Meshes

Limit Surfaces

# Isocontours from volumetric data





# Geometry Driven Diffusion

## Evolution (time dependent)

Linear heat conduction equation.

$$\partial_t \rho - \Delta\rho = 0\,, \qquad \Delta = \mathrm{div} \cdot \nabla$$

For equalizing spatial variation in concentration

For the surface *M*, the counterpart of the Laplacian $\Delta$ is the Laplace Beltrami operator $\Delta_M$. Hence, one obtains the geometric diffusion equation

$$\partial_t x - \Delta_M x = 0, \qquad \Delta_M = \mathrm{div}_M \cdot \nabla$$

for surface point $x(t)$ on the surface $M(t)$

# Model of Geometric Diffusion

Partial Differential Equation

$$\partial_t x(t) - \mathrm{div}_{M(t)}(\nabla_{M(t)} x(t)) = 0$$

$$M(0) = M$$

where M(t) is the solution surface at time t, x(t) is surface point.

Divergence $\mathrm{div}_{M(t)}\ \boldsymbol{v}$ for a vector field $v \in V$ is defined as the dual operator of the gradient:

$$\int_M \mathrm{div}_M v \phi dx := -\int_M v^T \nabla \phi dx, \quad \forall \phi \in C^\infty(M)$$

# More Related Work

- Desbrun et al (1999), use an implicit discretization of geometric diffusion to obtain strongly stable numerical smoothing scheme.

- Clarenz et. al (2000) introduce the anisotropic geometric diffusion to enhance features while smoothing

Above based on a discretized surface model.
Hence, the first and the second order derivative information, such as normals, tangents and curvatures are estimated using some local averaging or fitting scheme.

# Variational form

$$\left(\partial_t x(t), \theta\right)_{M(t)} + \left(\nabla_{M(t)} x(t), \nabla_{M(t)} \theta\right)_{TM(t)} = 0,$$

$$\forall \theta \in C^{\infty}(M(t))$$

where

$$(f, g)_M = \int_M f g dx, \qquad (\phi, \psi)_{TM} = \int_M \phi^T \psi dx$$

- How to represent $M(t)$ ?
- How to choose $\theta$ ?

# Spatial Discretization

$$(\partial_t x(t), \theta)_{M(t)} + (\nabla_{M(t)} x(t), \nabla_{M(t)} \theta)_{TM(t)} = 0, \qquad \forall \theta \in V_{M(t)}$$

Let

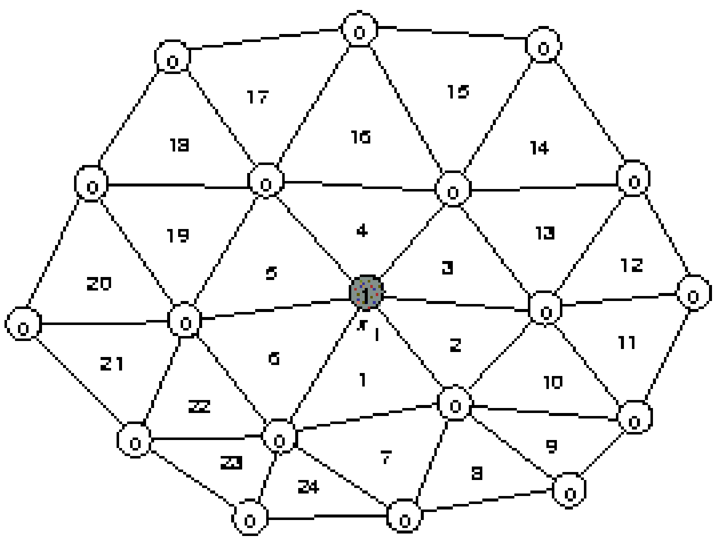
$$x(t) = \sum_{i=1}^{m} c_i(t)\phi_i(x), \qquad \theta = \phi_j(x)$$

Then we have a set of ordinary differential equations

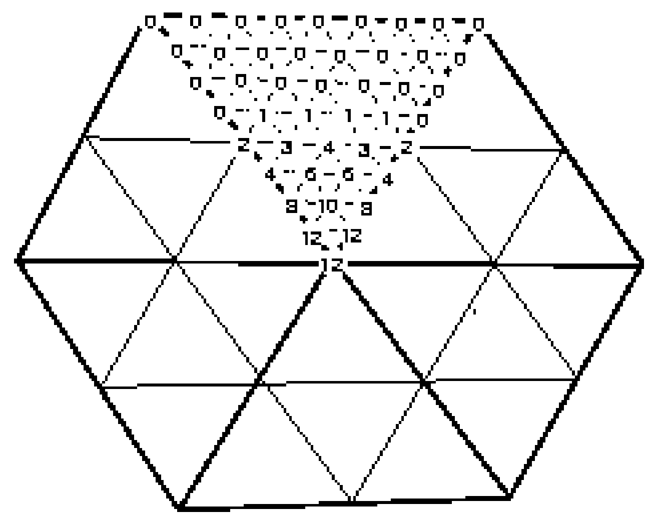
$$\sum_{i=1}^{m} c_i'(t)(\phi_i(x), \phi_j(x))_{M(t)} + \sum_{i=1}^{m} c_i(t)(\nabla_{M(t)}\phi_i(x), \nabla_{M(t)}\phi_j(x))_{TM(t)} = 0$$

$$j = 1, \ldots, m$$

## Where $\phi_i$ are the Loop limit surface basis functions (box splines)



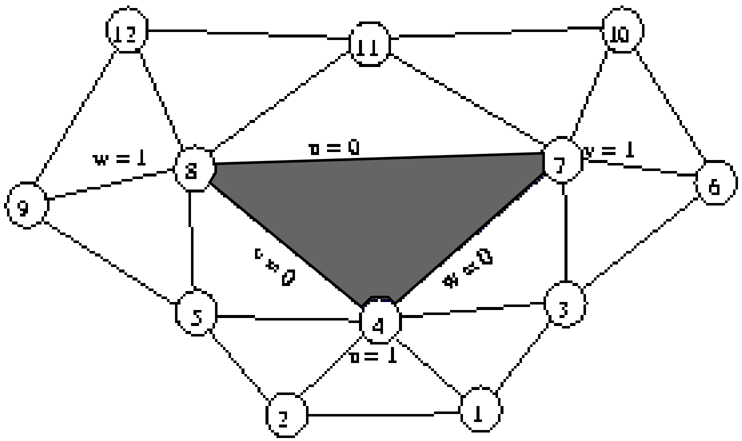

(a)




(b)

# Loop Subdivision Limit Surfaces (Box splines)

$$x(t) = \sum_{i=1}^{m} c_i(t)\phi_i(x), \quad \theta = \phi_j(x)$$




$$N_1 = \tfrac{1}{12}(u^4 + 2u^3 v),$$

$$N_2 = \tfrac{1}{12}(u^4 + 2u^3 w),$$

$$N_3 = \tfrac{1}{12}[u^4 + v^4 + 6u^3 v + 6uv^3 + 12u^2 v^2 + (2u^3 + 2v^3 + 6u^2 v + 6uv^2)w],$$

$$N_4 = \tfrac{1}{12}[6u^4 + 24u^3(v+w) + u^2(24v^2 + 60vw + 24w^2) + u(8v^3 + 36v^2 w + 36vw^2 + 8w^3) + (v^4 + 6v^3 w + 12v^2 w^2 + 6vw^3 + w^4)]$$

$$(u, v, w) \rightarrow (v, w, u): \quad N_1, N_2, N_3, N_4 \rightarrow N_{10}, N_6, N_{11}, N_7$$

$$(u, v, w) \rightarrow (w, u, v): \quad N_1, N_2, N_3, N_4 \rightarrow N_9, N_{12}, N_5, N_8$$

# Time Discretization

Let $X^n$ be approximation of $x(n\tau)$ , where $\tau$ is the timestep. Then
The semi-implicit discretization is

$$\left(\frac{X^{n+1}-X^n}{\tau}, \phi_i\right)_{M(n\tau)} +$$

$$(\nabla_{M(n\tau)} X^{n+1}, \nabla_{M(n\tau)} \phi i)_{TM(n\tau)} = 0, i = 1, \cdots, m$$

Since

$$x(t) = \sum_{i=1}^{m} c_i(t)\phi_i(x)$$

Then we have a linear system.

$$(M^n + \tau L^n)C((n+1)\tau) = M^n C(n\tau)$$

where $C(t) = [c_1(t), \cdots, c_m(t)]$

$$M^n = \left((\phi_i, \phi_j)_{M(n\tau)}\right)_{i,j=1}^m$$

and

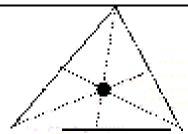
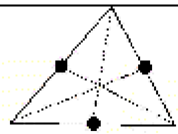
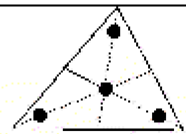
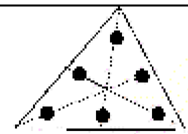
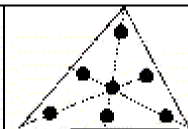
$$L^n = \left((\nabla_{M(n\tau)}\phi_i, \nabla_{M(n\tau)}\phi_j)_{TM(n\tau)}\right)_{i,j=1}^m$$

# Solving the linear system

- $M^n$ and $L^n$ are sparse.
- $M^n$ is symmetric and positive definite.
- $L^n$ is symmetric and nonnegative definite.
- $M^n + \tau L^n$ is symmetric and positive definite.

The system is solved by the conjugate gradient method.

# Numerical Integration

| |  |  |  |  |  |
|---|---|---|---|---|---|
| $v_1$ | 0.3333333333 | 0.0 | 0.1333333333 | 0.8168475729 | 0.05961587 |
| $v_2$ | | 0.5 | 0.1333333333 | 0.0915762135 | 0.47014206 |
| $v_3$ | | 0.5 | 0.7333333333 | 0.0915762135 | 0.47014206 |
| $v_4$ | | | 0.3333333333 | 0.1081030181 | 0.79742699 |
| $v_5$ | | | | 0.4459484909 | 0.10128651 |
| $v_6$ | | | | 0.4459484909 | 0.10128651 |
| $v_7$ | | | | | 0.33333333 |
| $w_1$ | 0.3333333333 | 0.5 | 0.7333333333 | 0.0915762135 | 0.47014206 |
| $w_2$ | | 0.0 | 0.1333333333 | 0.8168475729 | 0.05961587 |
| $w_3$ | | 0.5 | 0.1333333333 | 0.0915762135 | 0.47014206 |
| $w_4$ | | | 0.3333333333 | 0.4459484909 | 0.10128651 |
| $w_5$ | | | | 0.1081030181 | 0.79742699 |
| $w_6$ | | | | 0.4459484909 | 0.10128651 |
| $w_7$ | | | | | 0.33333333 |
| $W_1$ | 1.0 | 0.3333333333 | 0.5208333333 | 0.1099517436 | 0.13239415 |
| $W_2$ | | 0.3333333333 | 0.5208333333 | 0.1099517436 | 0.13239415 |
| $W_3$ | | 0.3333333333 | 0.5208333333 | 0.1099517436 | 0.13239415 |
| $W_4$ | | | 0.5625 | 0.2233815896 | 0.12593918 |
| $W_5$ | | | | 0.2233815896 | 0.12593918 |
| $W_6$ | | | | 0.2233815896 | 0.12593918 |
| $W_7$ | | | | | 0.225 |
| $p$ | 1 | 2 | 3 | 4 | 5 |

Integration rules over triangle. $(1-v_i-w_i, v_i, w_i)$ are barycentric coordinates of the nodes, $W_i$ are the weights. The last row represents the algebraic precision.

# Anti-Shrinking

Denote the $x$, $y$ and $z$ components of the surface point $x(t)$ as $x_1(t)$, $x_2(t)$ and $x_3(t)$, respectively. Then, we have

$$(\partial_t x_i(t), x_i(t))_{M(t)} = -(\nabla_{M(t)} x_i(t), \nabla_{M(t)} x_i(t))_{TM(t)}$$

and

$$\frac{\partial (x(t), x(t))_{M(t)}}{\partial t} = 2(\partial_t x(t), x(t))_{M(t)} = -4Area(M(t))$$

$$\frac{\partial (Area(M(t)))}{\partial t} = -\int_{M(t)} H^2 dx$$

Since $Area(M(t)) > 0$, the surface point $x(t)$ shrinks towards the origin at the average speed of $4Area(M(t))$.

To avoid such a shrinking, the surface point is magnified by a factor

$$\alpha = \sqrt[4]{\frac{(x(0), x(0))_{M(0)}}{(x(t), x(t))_{M(t)}}} \geq 1$$

# Shrink Bunny

# Diffusion Tensor

$$\partial_t x(t) - div(a(x)\nabla_{M(t)} x(t)) = 0$$

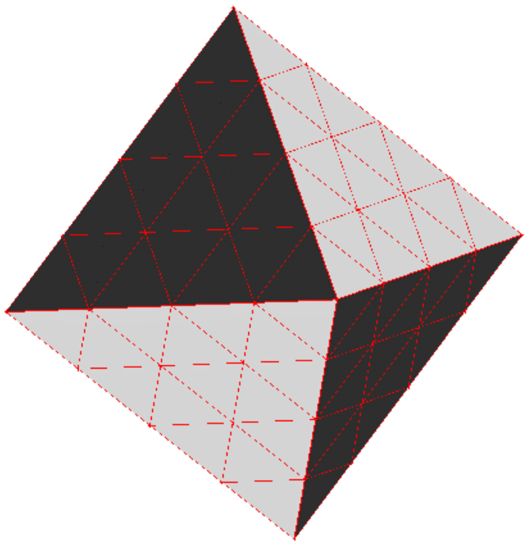
*a(x)* is a symmetric, positive define linear mapping on the Tangent space

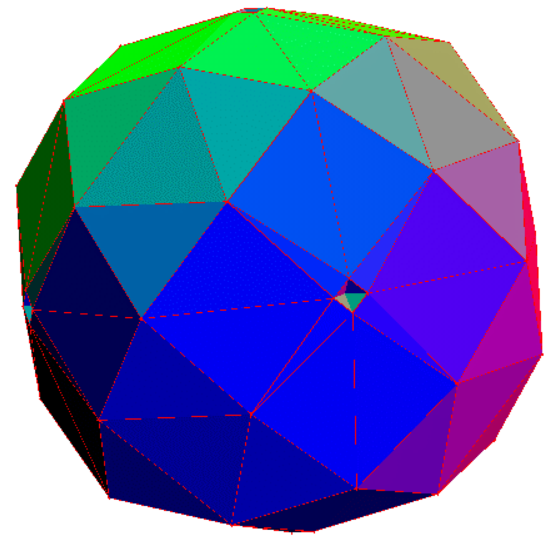
$$a(x) : TM \to TM$$

The problem is how to choose the diffusion tensor?

# Anti – Crease by Diffusion Tensor

Initial Mesh

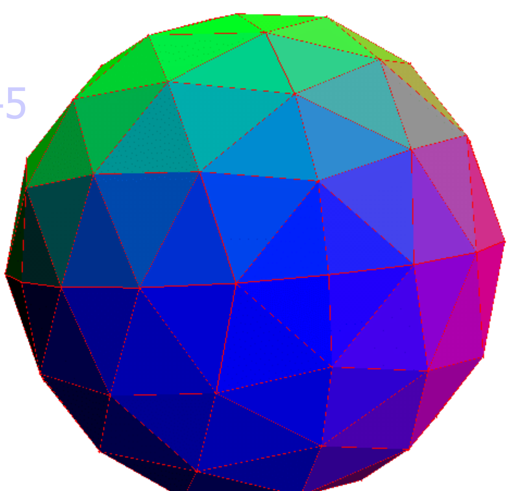


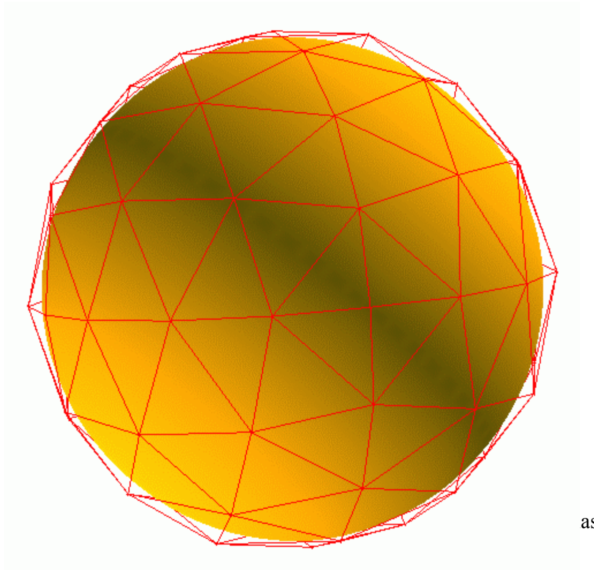


After 11,114 iterations

0.45

a=(area) of triangle. After 2228 iterations.

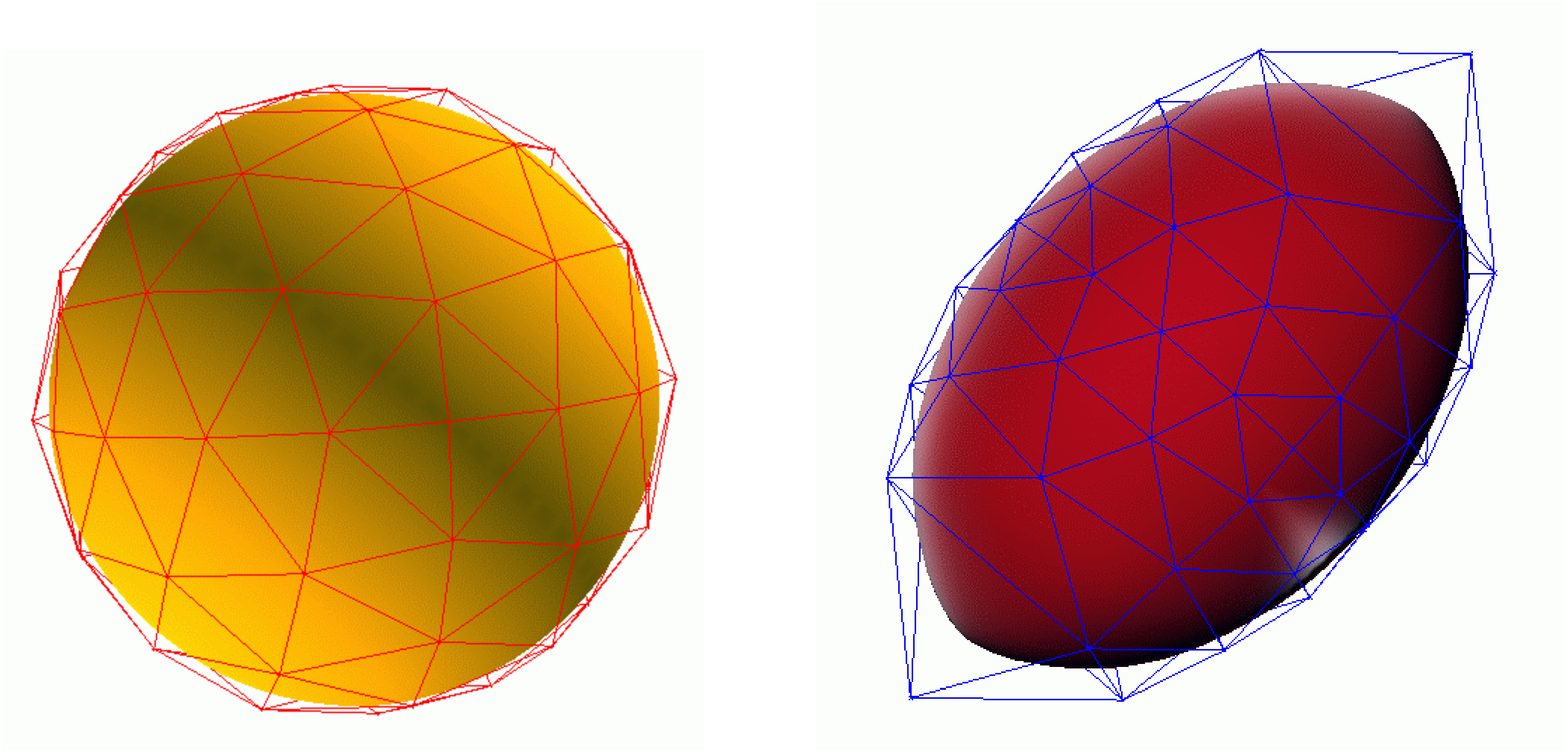




Limit surface

as at Austin

# Change Shape by Diffusion Tensor



$$a(x) = x_1^2 + x_2^2, \quad where \quad x = (x_1, x_2, x_3)$$

# Enhance Sharp Features

Let $v^{(1)}(x), v^{(2)}(x),$ be the principle directions of $M(t)$ at print $x(t)$.

$N(x)$ Be the normal at that point.

Then any vector z in the tangent plane could be expressed as

$$z = \alpha v^{(1)}(x) + \beta v^{(2)}(x) + \delta N(x)$$

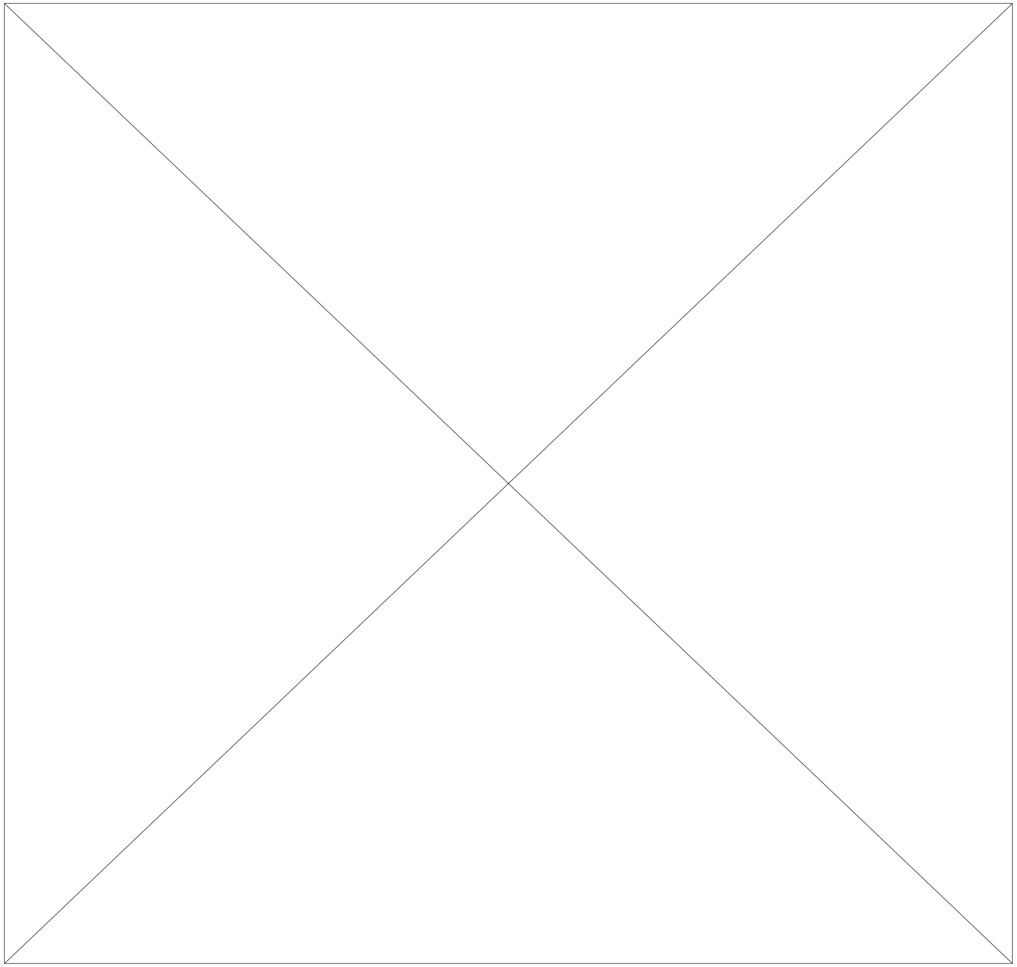
Then define $a$, such that

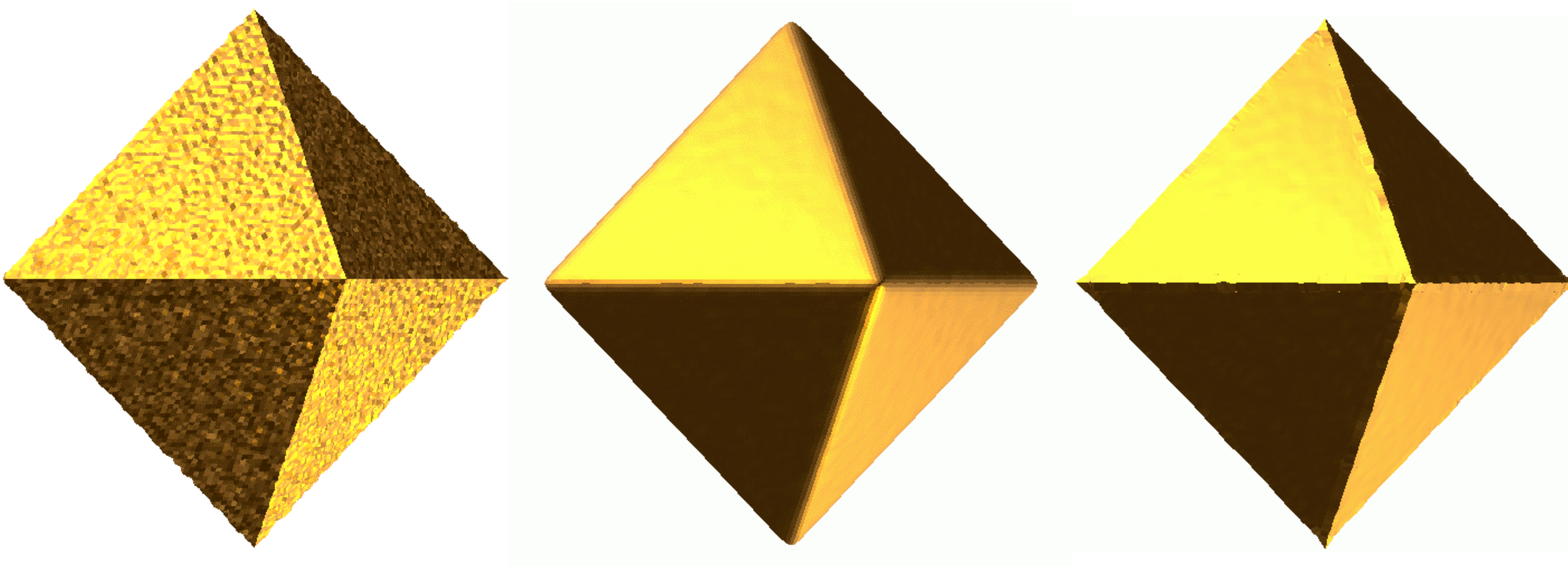
$$az = g(k_1)\alpha v^{(1)}(x) + g(k_2)\beta v^{(2)}(x) + \delta N(x)$$

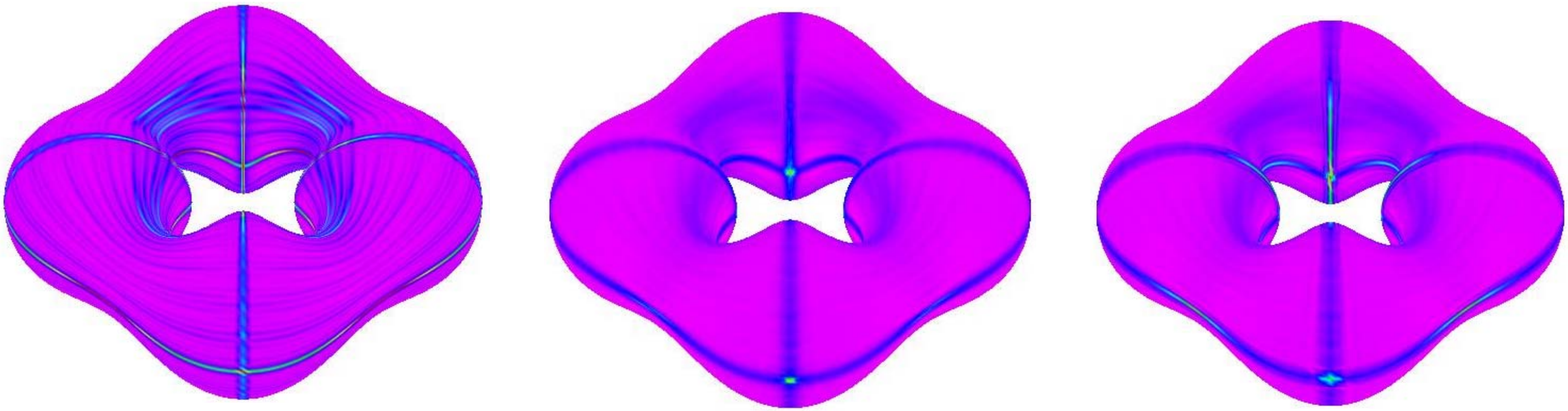
where

$$g(s) = \begin{cases} 1, & s \le \lambda \\ 2(1+\frac{s^2}{\lambda^2})^{-1}, & s > \lambda \end{cases}$$

$\lambda > 0$ is given constant.





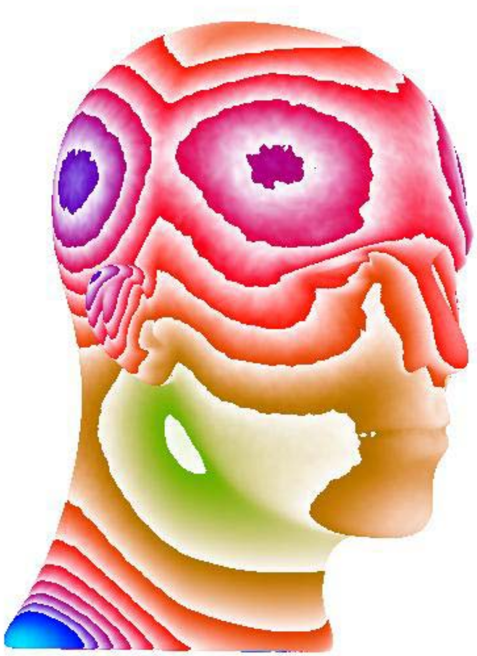


Initial functions | After three iterations | After five iterations

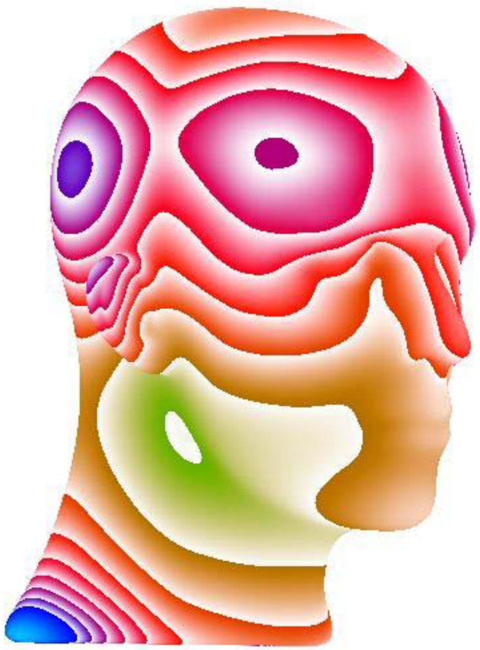
Mean curvature plot: non-smooth functions at x=0, y=0, z=0

# Smoothness visualization

Iso – Contour plot

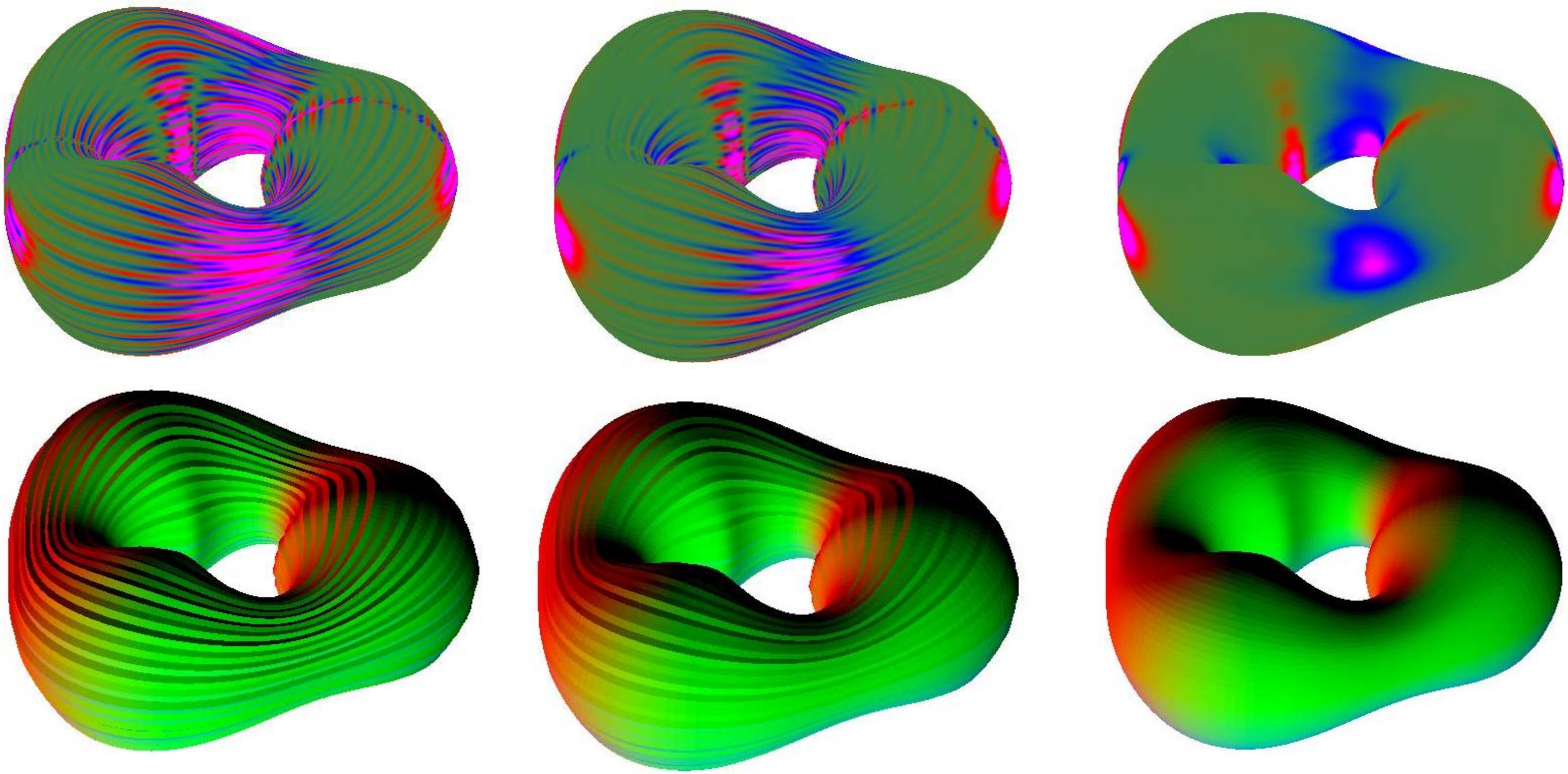


Initial data



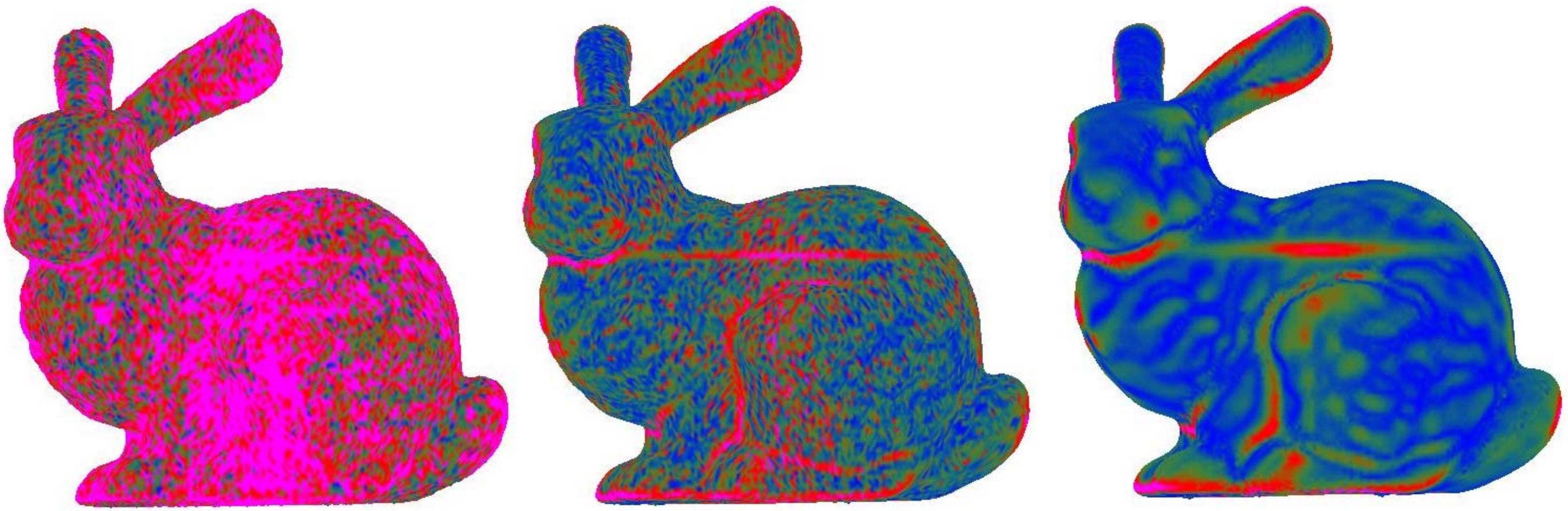
After 4 fairing iterations

# Riemannian Curvature Plot



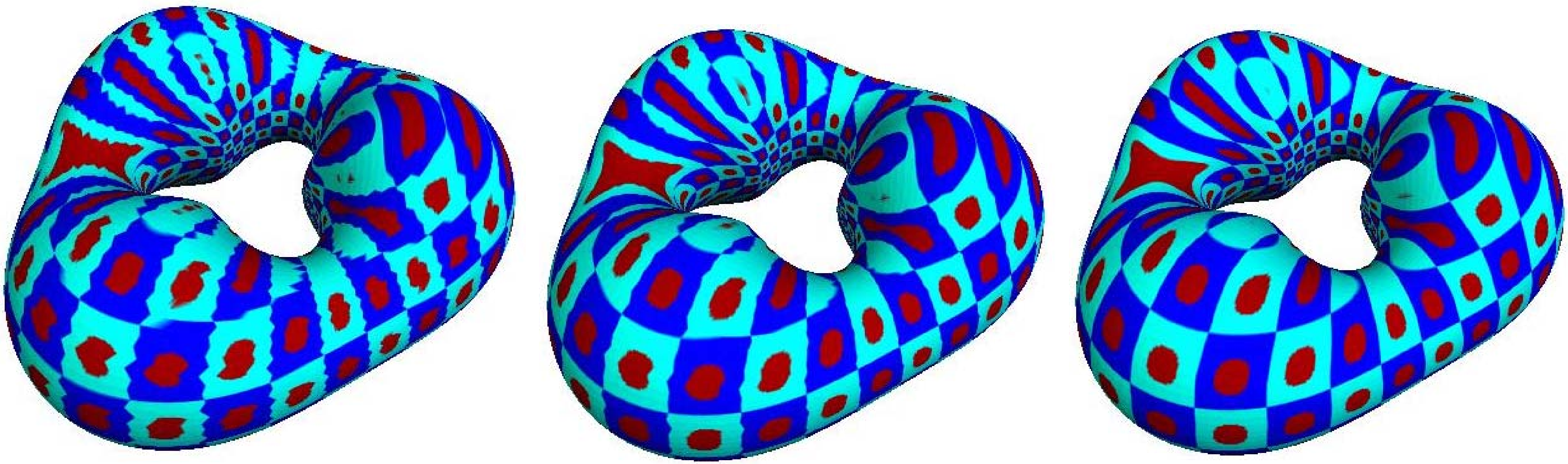
Initial data | After 1 iteration | After 4 iterations

# Mean Curvature Plot



Initial data    After 1 iteration    After 4 iterations

# Functions on Surface: Texture Map



Initial dada After 1 iteration After 4 iterations

# Remaining Problems

1. Error analysis
2. Play with the tensor.
3. Bad triangulations.

# Further reading

[1] Bajaj, C. and Pascucci, V. 2000. Time critical adaptive refinement and smoothing. In *proceeding of the ACM/IEEE Volume Visualization and Graphics Symposium 2000*, Salt Lake City, Utah, 33-42.

[2] C. Bajaj, T. Dey, Convex Decompositions of Polyhedra and Robustness. *Siam Journal on Computing*, 21, 2, (1992), 339-364.

[3] C. Bajaj, E. Coyle, K-N. Lin, *Tetrahedral Meshes from Planar Cross Sections*, Computer Methods in Applied Mechanics and Engineering, Vol. 179, 1999, 31-52.

[4] C. Bajaj, E. Coyle, K-N. Lin, *Arbitrary Topology Shape Reconstruction from Planar Cross Sections*, Graphical Modeling and Image Processing, Vol. 58, No. 6, 1996, 524-543.

[5] C. Bajaj, G. Xu, *Smooth Shell Construction with Mixed Prism Fat Surfaces*, Geometric Modeling, Springer Verlag, Computing Supplementum 14, 2001, 19-36.

[6] C. Bajaj, G. Xu, R. Holt, A. Netravali, *Hierarchical Multiresolution Reconstruction of Shell Surfaces*, Computer Geometric Aided Design, 2002.

[7] Bern M. W., and Dey T. K., 2002. Quality meshing with weighted delaunay refinement. *In Proc. 13$^{th}$ ACM-SIAM Sympos. Discrete Algorithms (SODA 2002)*, 137-146.

[8] S.-W. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello, S. Teng, *Sliver Exudation*, Proc. Journal of ACM, Vol. 47, 2000, 883-904.

# Further Reading


[9] S.-W, Cheng, T. K. Dey, *Quality Meshing with Weighted Delaunay Refinement*, Proc. 13th ACM-SIAM Sympos. Discrete Algorithms (SODA 2002), 2002, 137-146.

[10] I. Fujishiro, Y. Maeda, H. Sato, Y. Takeshima, *Volumetric Data Exploration Using Interval Volume*, IEEE Transactions on Visualization and Computer Graphics, Vol. 2, No. 2, June 1996, 144-155.

[11] T. Ju, F. Losasso, S. Schaefer, J. Warren, *Dual Contouring of Hermite Data*, Computer Graphics (SIGGRAPH 02 Proceedings). To Appear. ACM Press/ACM SIGGRAPH, 2002.

[12] L. Kobbelt, M. Botsch, U. Schwanecke, H., Seidel, *Feature Sensitive Surface Extraction from Volume Data*, Computer Graphics (SIGGRAPH 01 Proceedings), 2001, 57-66.

[13] A. Lopes, K. Brodlie, *Improving the Robustness and Accuracy of the Marching Cubes Algorithm for Isosurfacing*, IEEE Transactions on Visualization and Computer Graphics.

[14] W. Lorensen, H. Cline, *Marching Cubes: a high resolution 3D surface construction algorithm*, Computer Graphics (SIGGRAPH 87 Proceedings), 1987, 163-169.

[15] MEYERS, D., Multiresolution Tiling. *Computer Graphics Forum* 13, 5 (December 1994), 325--340.

[16] Mitchell S. A., and Vavasis S. A. 2000. Quality mesh generation in higher dimensions. *SIAM J. Comput 29*, 1334-1370.

[17] D. Moore, *Subdividing Simplices*, Graphics Gems III, 1992, 244-249.

[18] D. Moore, *Understanding Simploids*, Graphics Gems III, 1992, 250-255.

[19] D. Moore, J. Warren, *Compact Isocontours from Sampled Data*, Graphics Gems III, 1992, 23-28.

[20] V. Natarajan, H. Edelsbrunner, *Simplification of 3-Dimensional Density Maps*, Second Year Project Report, Dept. of Computer Science, Duke Univ., 2001.

[21] G. M. Nielson, J. Sung, *Interval Volume Tetrahedrization*, IEEE Visualization `97, 1997, 221-228.

[22] G. M. Nielson, B. Hamann, *The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes*, in Proceedings of Visualization `91, 1991, 83-90.

[23] J. R. Shewchuk, *Constrained Delaunay Tetrahedrizations and Provably Good Boundary Recovery*, Submitted to 11th International.

[24] J. R. Shewchuk, *Tetrahedral Mesh Generation by Delaunay Refinement*, Proceedings of the 4th Annual Symposium on Computational Geometry (Minneapolis, Minnesota), Association for Computational Machinery, June 1998, 86-95.

# Further reading


[25] J. R. Shewchuk, *Two Discrete Optimization Algorithms for the Topological Improvement of Tetrahedral Meshes*, Submitted to 11th International Meshing Roundtable, 2002.

[26] J. R. Shewchuk, *What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures*, Submitted to 11th International Meshing Roundtable, 2002.

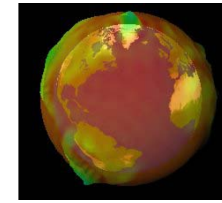
[27] S. A. Mitchell, S. A. Vavasis, *Quality Mesh Generation in Higher Dimensions*, SIAM J. Comput., Vol. 29, 2000, 1334-1370.

[28] R. Westermann et al., Real-time Exploration of Regular Volume Data by Adaptive Reconstruction of Isosurfaces

[29] Z. Wood, M. Desbrun, P. , D. Breen, *Semi-Regular Mesh Extraction from Volumes*, Proceedings of Visualization 2000.

[30] C. Bajaj, G. Xu **Anisotropic Diffusion of Subdivision Surfaces and Functions on Surfaces**, *To appear in ACM Transactions on Graphics, 2002*

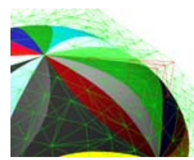
[31] C. Bajaj, G. Xu **Anisotropic Diffusion of Noisy Surfaces and Noisy Functions on Surfaces**, *Proc. of Information Visualization 2001—Computer Aided Geometric Design, London, UK, 2001*



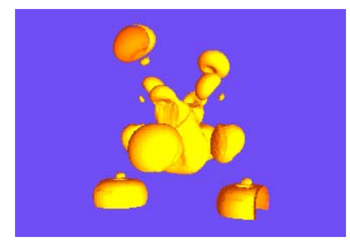
# Computational Visualization

1. Sources, characteristics, representation

2. Mesh Processing

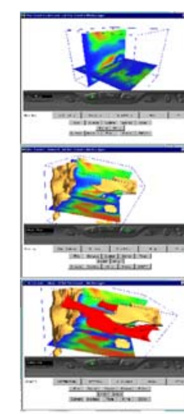


3. Contouring



4. Volume Rendering

5. Flow, Vector, Tensor Field Visualization



6. Application Case Studies