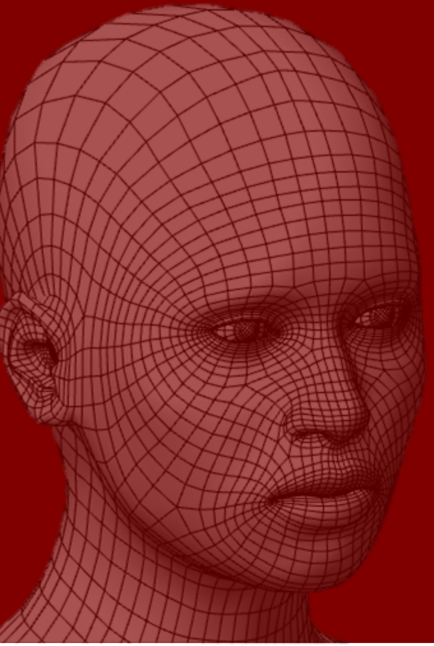


# Advanced Image Processing

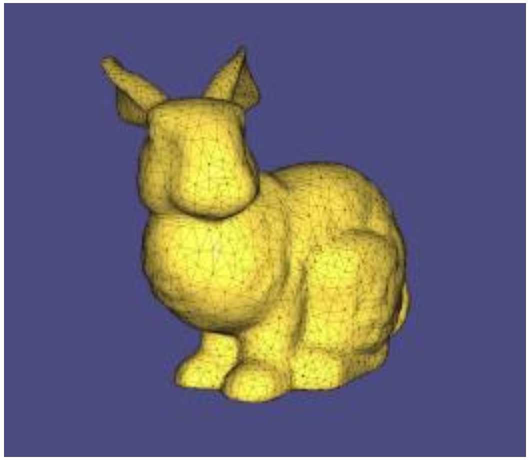
## Lesson 4: Mesh processing

Speaker: Alice OTHMANI, PhD
Associate professor at UPEC

Email: alice.othmani@u-pec.fr

UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE

- Geometry processing involves working with a shape.
- Shape is a basic property of most objects.
- The shape can live in a space of arbitrary dimensions.

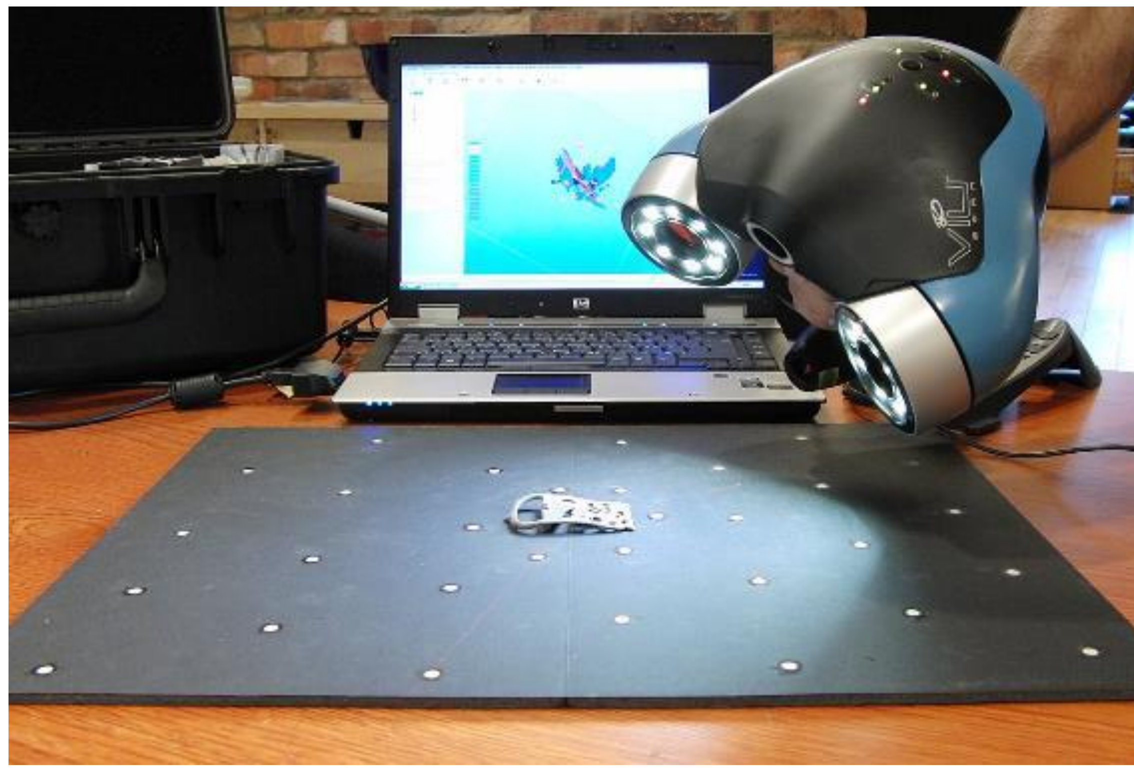


A mesh of the famous Stanford bunny. Shapes are usually represented as a mesh, a collection of polygons that delineate the contours of the shape.

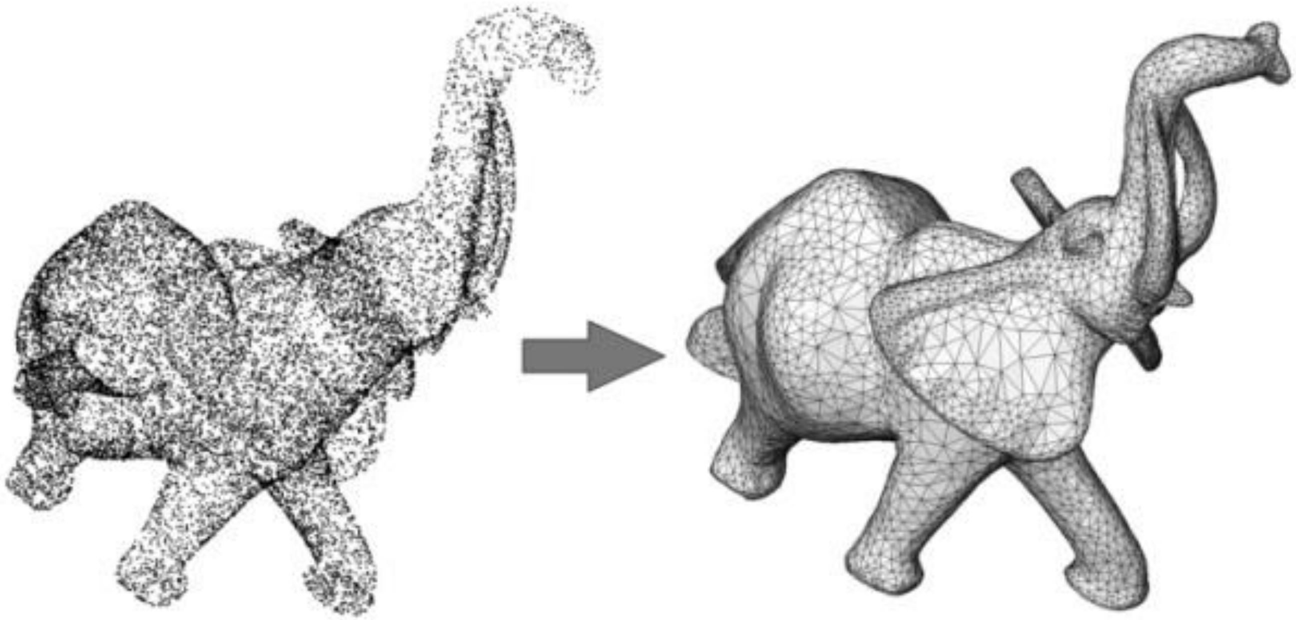
## Acquiring 3D geometry

- 3D scanning is the process of analyzing a real-world object or environment to collect data on its **shape** and possibly its **appearance** (e.g. **colour**).

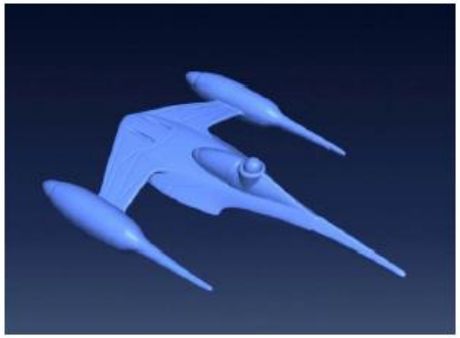
The collected data can then be used to construct digital **3D models**.



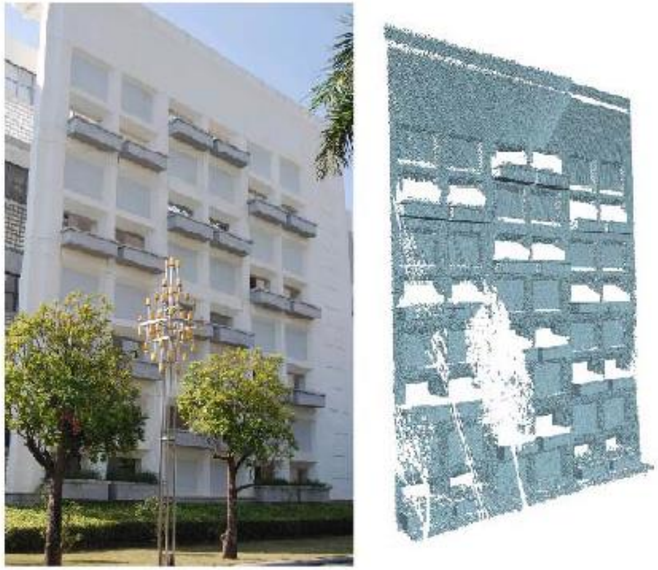
- The purpose of a 3D scanner is usually to create a **3D model**.
- This 3D model consists of a **point cloud** of geometric samples on the surface of the subject.
- These points can then be used to extrapolate the shape of the subject (a process called **reconstruction**).



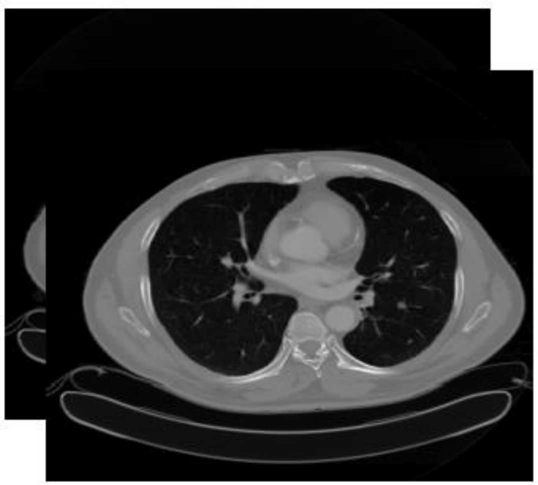
## Range Scanners

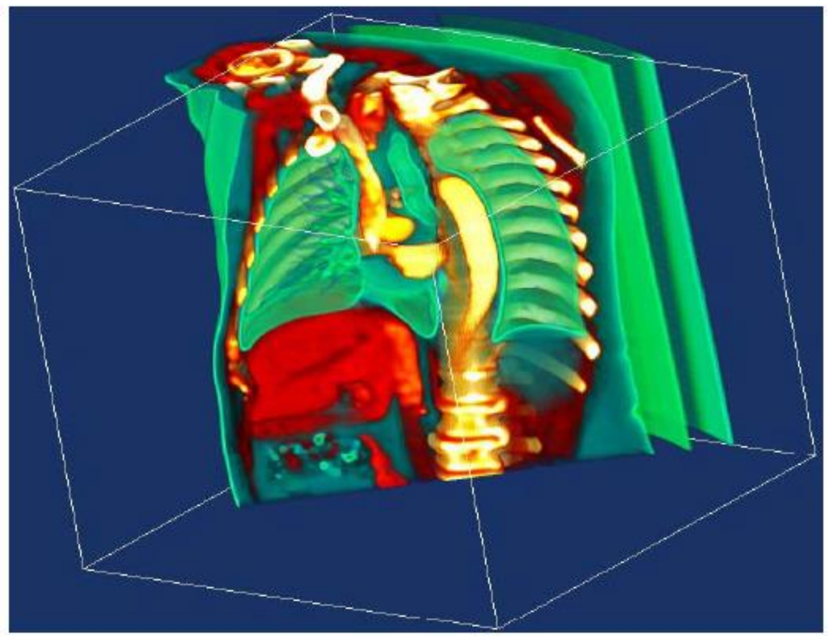



## Range Scanners

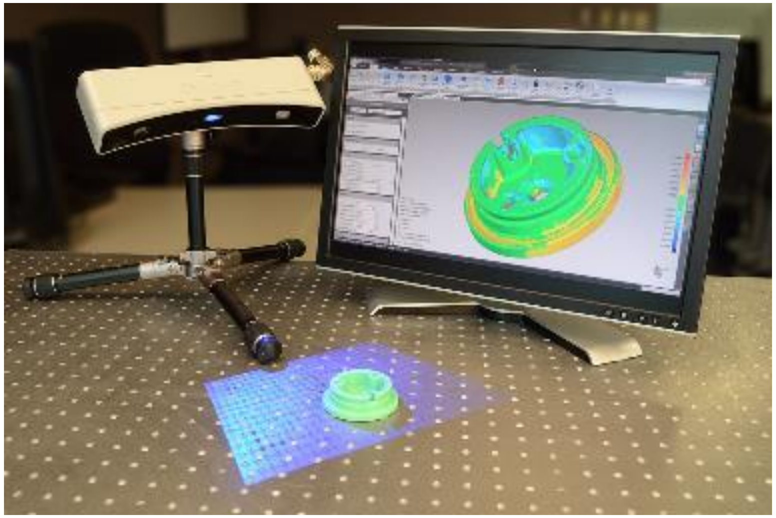


## Tomography





- A shape can be instantiated through one of three methods: a **model**, a **mathematical representation**, or a **scan**.
- After a shape is born, it can be analyzed and edited repeatedly in a cycle.
- Editing may involve **denoising**, **deforming**, or performing **rigid transformations**.



- Geometry processing is about the Creation & manipulation of 3D geometry

“SFMedu: A Structure from Motion System for Education”, Jianxiong Xiao
http://3dvision.princeton.edu/courses/SFMedu/

ICCV 2017

# BodyFusion

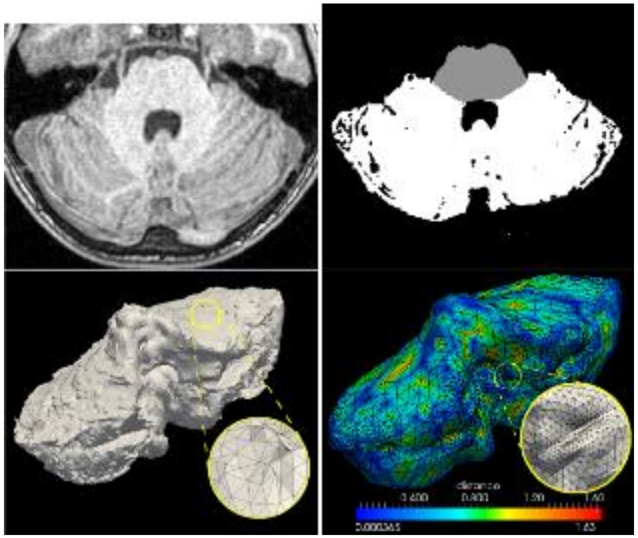
## Real-time Capture of Human Motion and Surface Geometry Using a Single Depth Camera

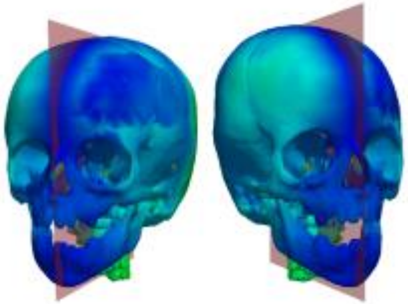
Tao Yu[1,2], Kaiwen Guo[2], Feng Xu[2], Yuan Dong[2], Zhaoqi Su[2],
Jianhui Zhao[1], Jianguo Li[3], Qionghai Dai[2], Yebin Liu[2]

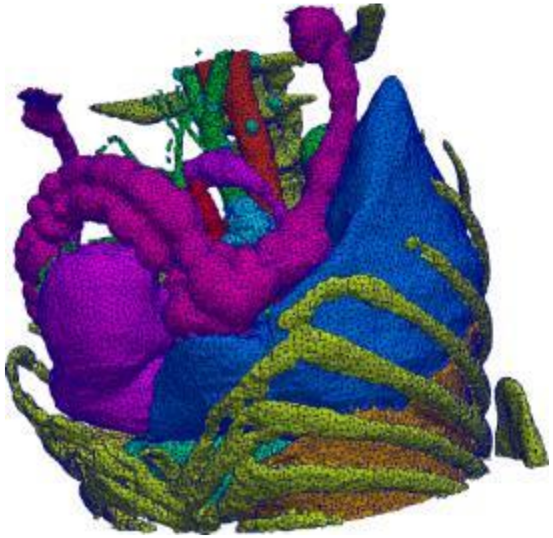
Beihang University, Beijing, China[1]
Tsinghua University, Beijing, China[2]
Intel Labs China, Beijing, China[3]

***A novel real-time motion tracking and fusion method called BodyFusion that reconstructs non-rigid surface motions of human performers using a single consumer-level depth camera.***

- **Focus on discrete (polygonal mesh) models**
  Typically triangular
- **Why?**
  - Simplicity – ease of description & transfer
  - Base data for rendering software/hardware
  - Input to most simulation/analysis tools
  - Output of most acquisition tools (CT, MRI, laser, etc..)



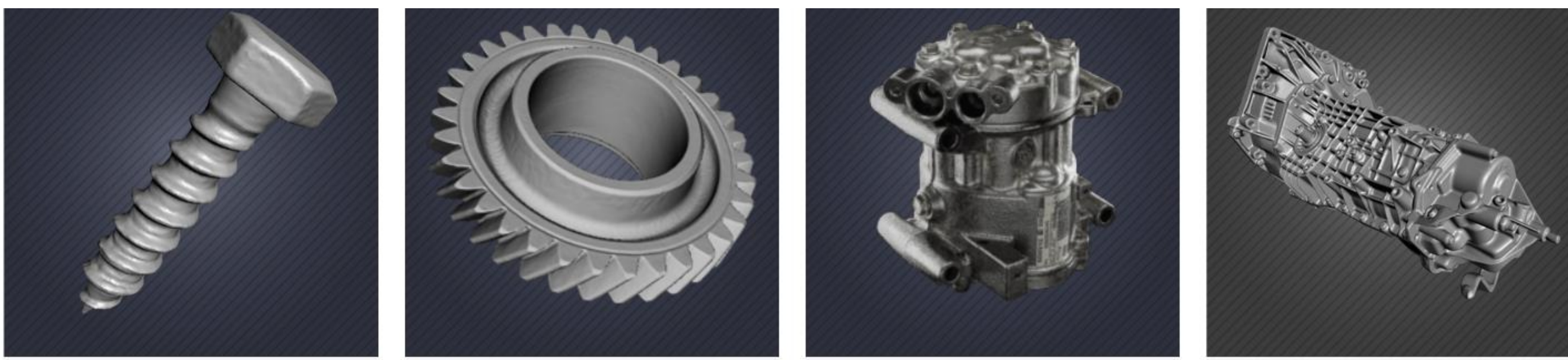




# Applications

## 3D Shape Capture for Heritage Preservation

## Industrial design and manufacturing

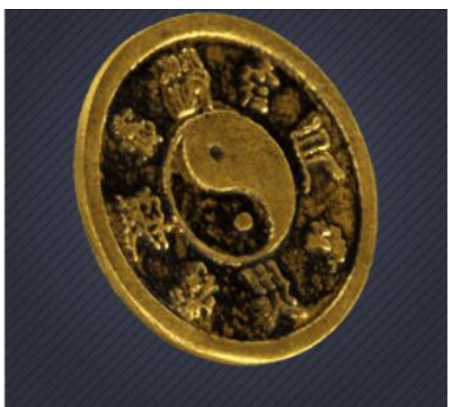
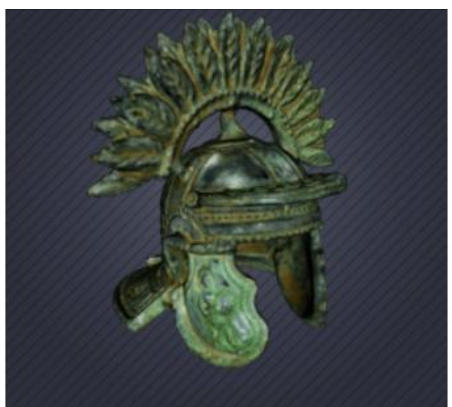


# Applications

## Science and education

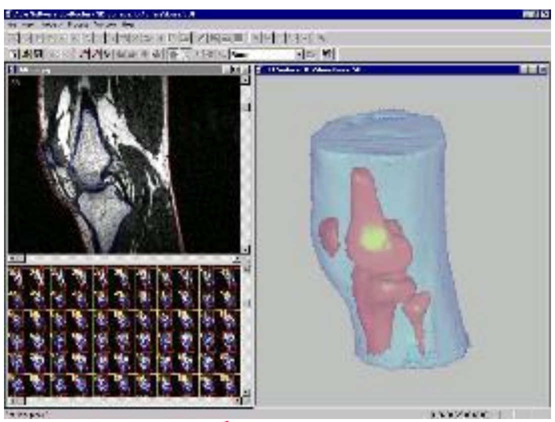


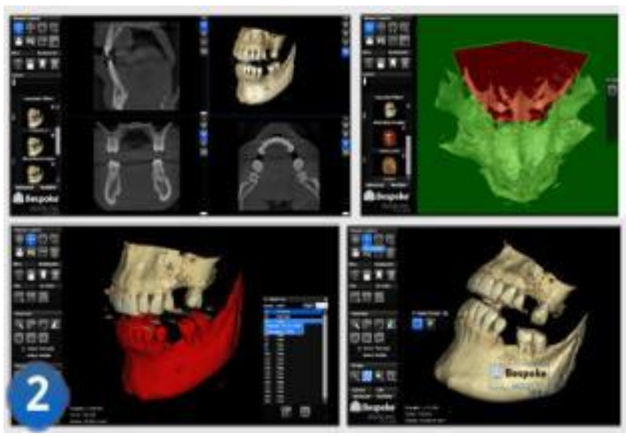

## Art and design

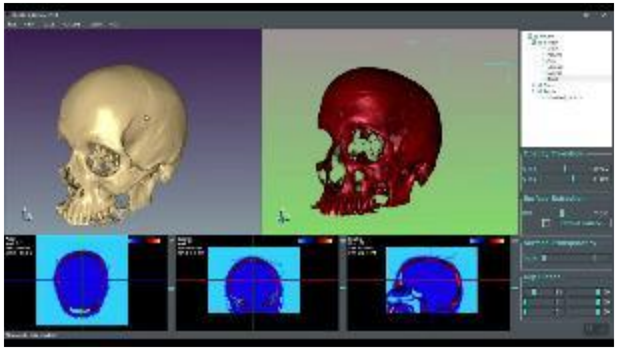



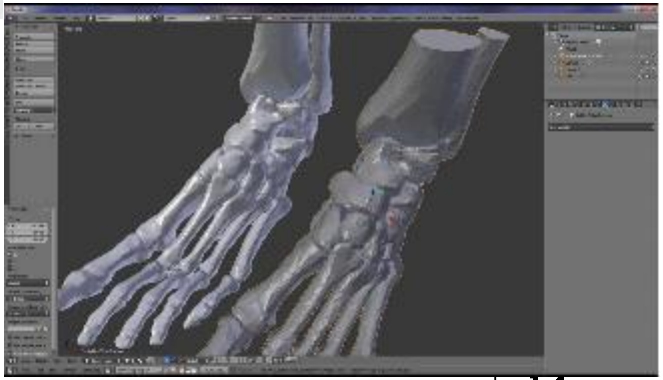
# Applications

## Medical Imaging

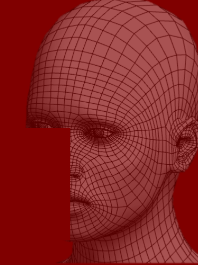








UPEC
UNIVERSITÉ
PARIS-EST CRÉTEIL
VAL DE MARNE
Connaissance - Action



## What is a Mesh?

- A Mesh is a pair *(P,K)*, where *P* is a set of point positions $P = \{ p_i \in R^3 \mid 1 \le i \le n \}$ and *K* is an abstract simplicial complex which contains all topological information.

- *K* is a set of subsets of $\{1, \ldots, N\}$:
  - Vertices $v = \{i\} \in V$
  - Edges $e = \{i, j\} \in E$
  - Faces $f = \{i_1, i_2, \ldots, i_{n_f}\} \in F$

- $K = V \cup E \cup F$

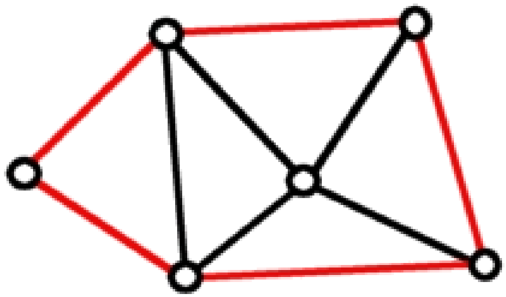
- Each edge must belong to at least one face, i.e.

  $$v = \{j\} \in V \text{ iff } \exists\, e = \{i, j\} \in E$$

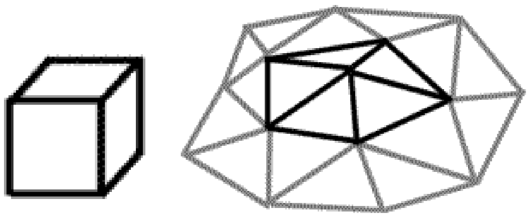
- Each vertex must belong to at least one edge, i.e.

  $$e = \{j, k\} \in E \text{ iff } \exists\, f = \{i_1, \cdots, j, k, \cdots, i_{n_f}\} \in F$$

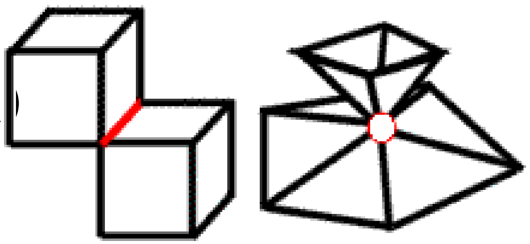
- An edge is a boundary edge if it only belongs to one face



- **A mesh is a manifold if**
  - Every edge is adjacent to one (boundary) or two faces
  - For every vertex, its adjacent polygons form a disk (internal vertex) or a half-disk (boundary vertex)



**Manifold**



**Non-manifold**

- A mesh is a **polyhedron** if
  - It is a manifold mesh and it is closed (no boundary)
  - Every vertex belongs to a cyclically ordered set of faces (local shape is a disk)

# Orientation of Faces

- Each face can be assigned an orientation by defining the ordering of its vertices
- Orientation can be clockwise or counter-clockwise.

- The orientation determines the normal direction of face. Usually counterclockwise order is the “front” side.

## Euler Formula

- The relation between the number of vertices, edges, and faces.

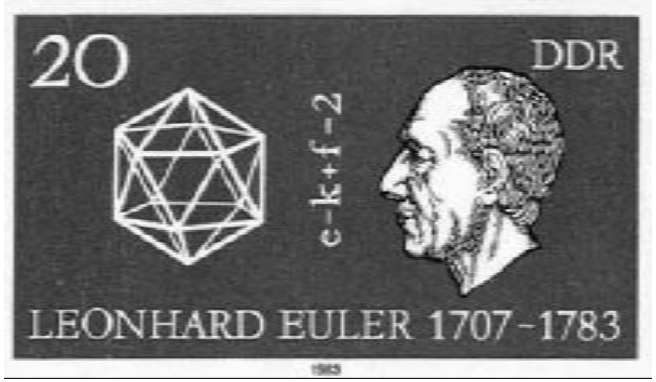
$$V - E + F = 2$$

- where

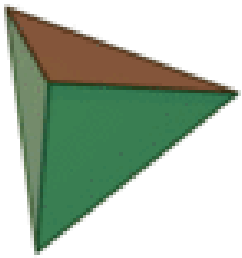
  V : number of vertices

  E : number of edges

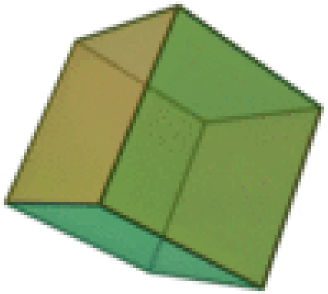
  F : number of faces



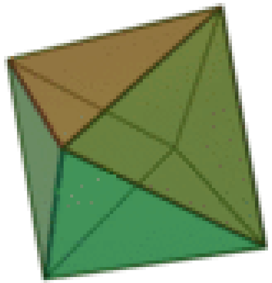
# Euler Formula



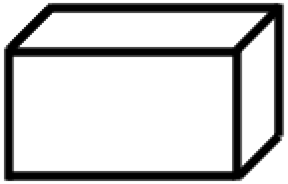
- **Tetrahedron**
  - V = 4
  - E = 6
  - F = 4
  - 4 - 6 + 4 = 2



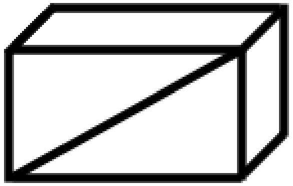
- Cube
  - V = 8
  - E = 12
  - F = 6
  - 8 -12 + 6 = 2



- Octahedron
  - V = 6
  - E = 12
  - F = 8
  - 6 -12 + 8 = 2



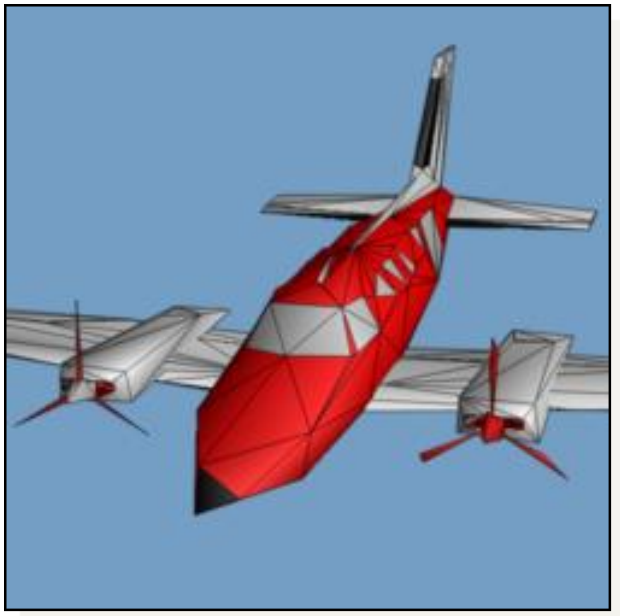
V = 8
E = 12
F = 6
8 - 12 + 6 = 2



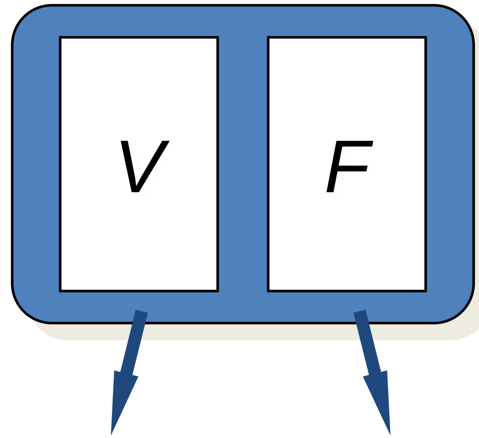
V = 8
E = 12 + 1 = 13
F = 6 + 1 = 7
8 - 13 + 7 = 2

# Mesh processing pipeline

Scan Reconstruct Clean Remesh ...



mesh $M$




Vertex 1 $x_1\ y_1\ z_1$
Vertex 2 $x_2\ y_2\ z_2$
…

Face 1 2 3
Face 3 2 4
Face 4 2 7
…

(appearance attributes:
normals, colors, textures, ...)

# Traditional Mesh Representation

```
v -6.4796930e-002 1.5210615e-001 -3.6185520e-002
v -6.4400320e-002 1.5834400e-001 -5.4256370e-002
v -6.6178120e-002 1.4218350e-001 -9.3766300e-003
v -6.7751430e-002 1.4605207e-001 -2.3333300e-002
v -6.4731580e-002 1.5410067e-001 -4.0464820e-002
v -2.4265590e-002 1.5687690e-001 -7.8509300e-003
v -1.5723180e-002 1.6312344e-001 -1.6396570e-002
v -7.0887660e-002 1.4404618e-001 -1.4908480e-002
v -4.4341830e-002 1.5113809e-001 -5.6859800e-003
v -6.2896810e-002 1.4694778e-001 -1.3098620e-002
v -6.3755400e-002 1.4428875e-001 -1.1395730e-002
v -6.8214560e-002 1.4390932e-001 -1.4984170e-002
v -5.0271440e-002 1.4336563e-001 1.5153000e-003
v -2.8535590e-002 1.6208479e-001 -1.4786030e-002
v -6.5810700e-002 1.4359119e-001 -1.2585380e-002
v -5.6179200e-002 1.3774406e-001 -4.0674300e-003
v -6.8866880e-002 1.4723338e-001 -2.8739870e-002
v -6.0965420e-002 1.7002113e-001 -6.0839390e-002
v -1.3895490e-002 1.6787168e-001 -2.1897230e-002
v -6.9413000e-002 1.5121847e-001 -4.4538540e-002
v -5.5039800e-002 5.7309700e-002 1.6990900e-002
f 1069 1647 1578
f 1058 909 939
f 421 1176 238
f 1055 1101 1042
f 238 1059 1126
f 1254 30 1261
f 1065 1071 1
f 1037 1130 1120
f 1570 2381 1585
f 2434 2502 2473
f 1632 1654 1646
f 1144 1166 669
f 1202 1440 305
```

**Exemple of mesh file with the format OBJ**

## Progressive Mesh

- New representation of triangular meshes.
- Simplify meshes through sequence of edge collapse transformations.
- Record the sequence of inverse transformations (vertex splits).
- It is necessary to undertake as many simplifications as needed to achieve the minimal model.
- hierarchical structure which helps to create a model in the chosen level of detail.

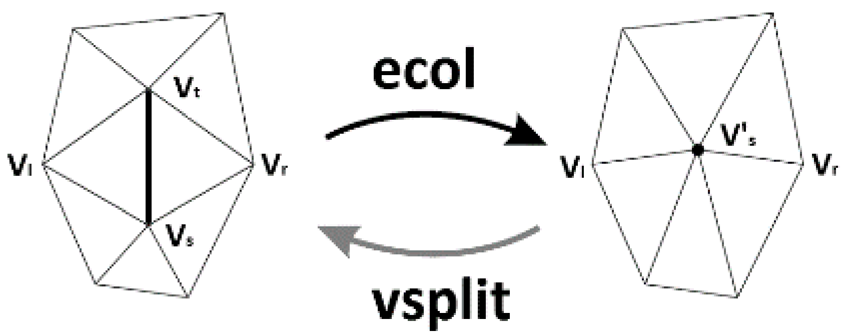
Hoppe, Progressive mesh, Siggraph 96
Hoppe, View-dependent Refinement of Progressive Meshes, Siggraph 97

- **Edge collapse**

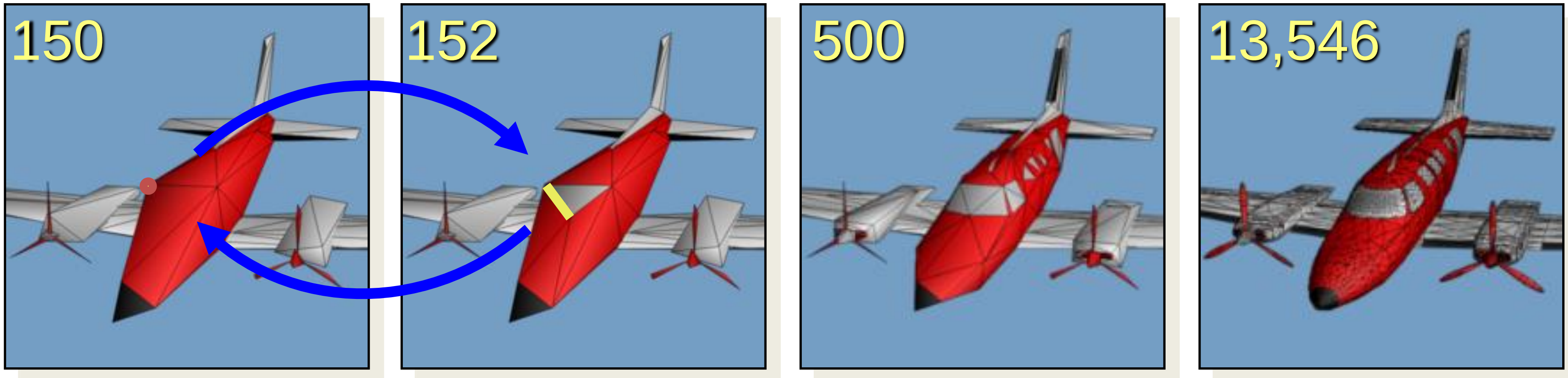
This simplistic operation - ecol takes two connected vertices and replaces them with a single vertex. Two triangles {vs, vt, vl} and {vt, vs, vr} which were connected by the edge are also removed during this operation.

- **Vertex split**

Vertex split (vsplit) is the inverse operation to the edge collapse that divides the vertex into two new vertexes. Therefore, a new edge {vt, vs} and two new triangles {vs, vt, vl} and {vt, vs, vr} arise.

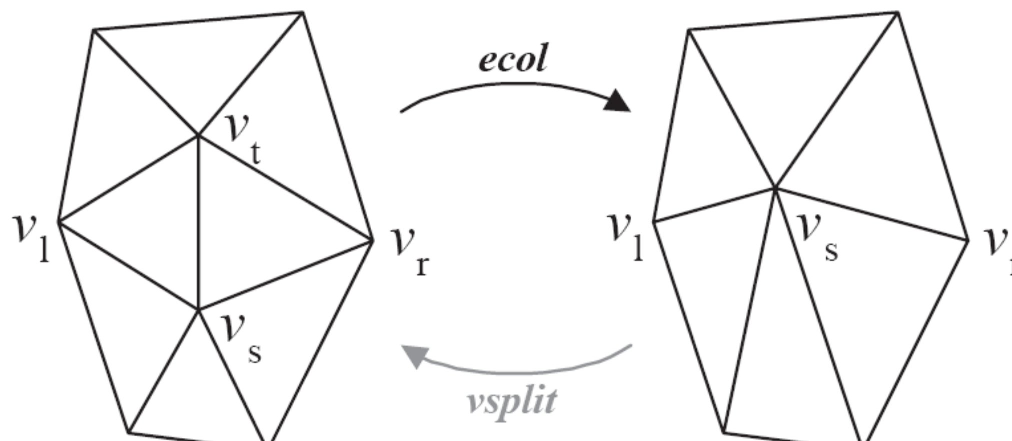


# Progressive Mesh Representation

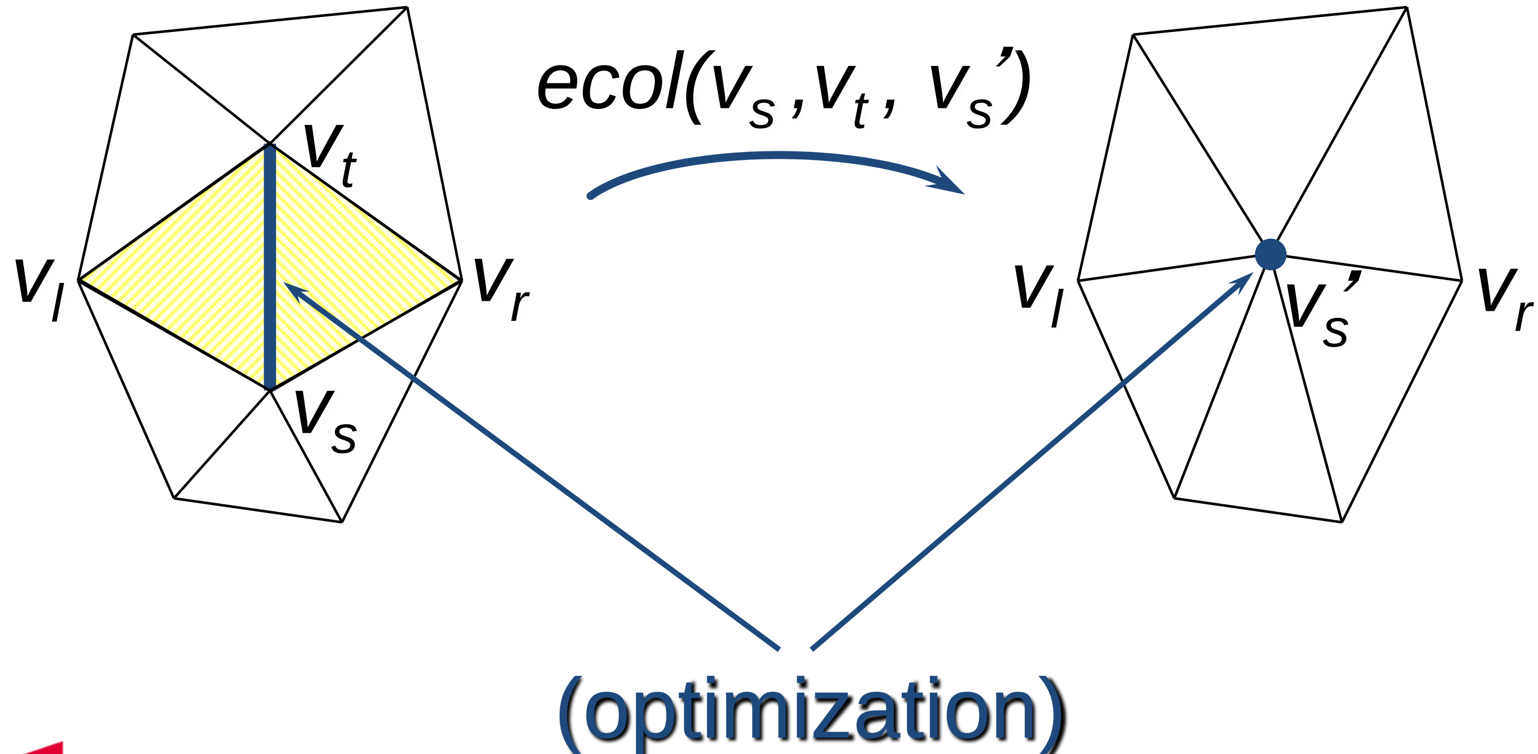



$M^0$ ⟷ $M^1$ ⟷ $M^{175}$ ⟷ $M^n$

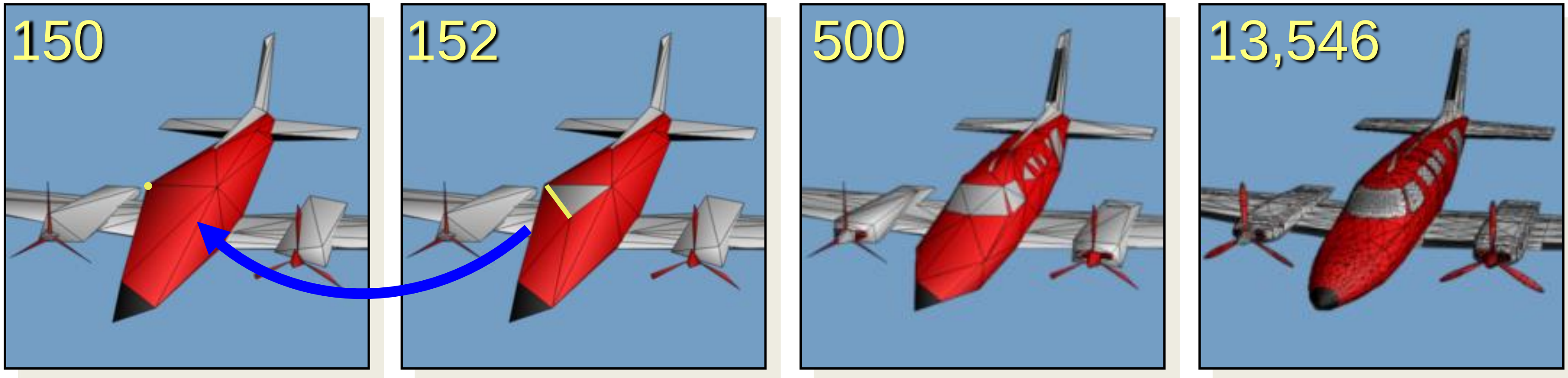
base mesh — Original mesh



- Idea: apply a sequence of edge collapses:



## Simplification: Edge Collapse

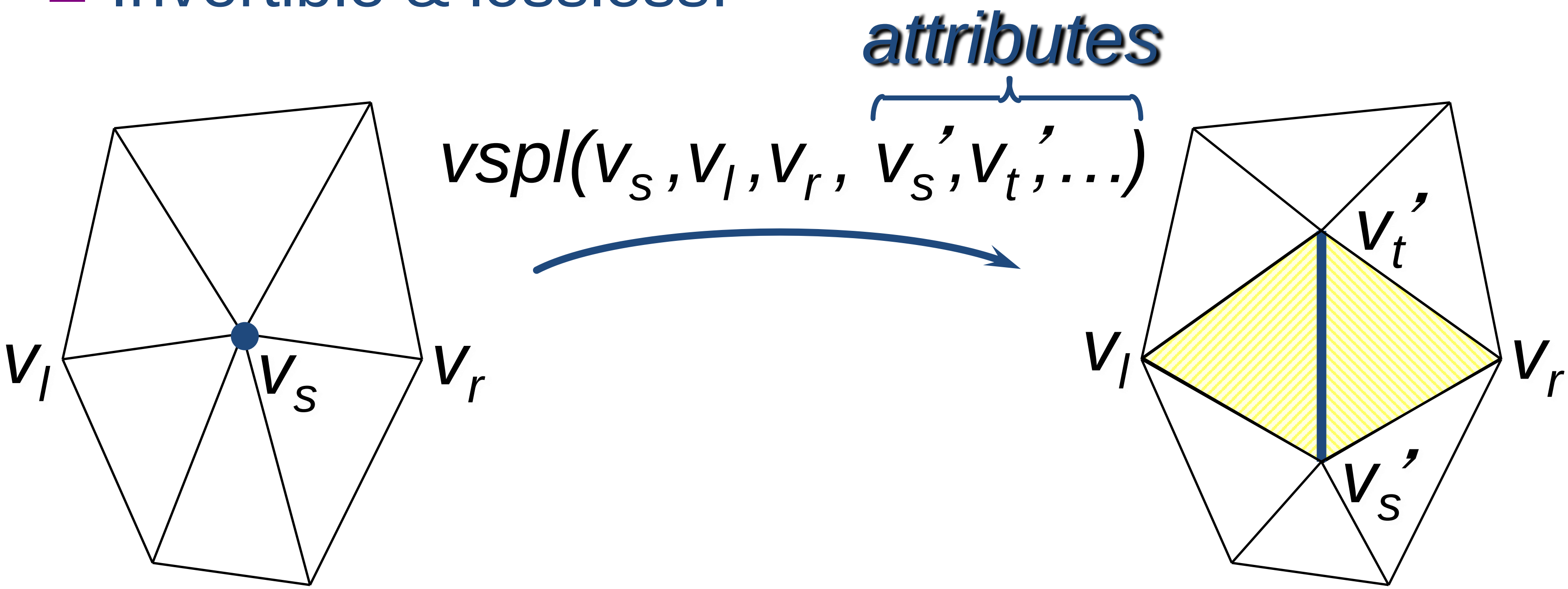



$$M^0 \xleftarrow{ecol_0} M^1 \quad \ldots \xleftarrow{ecol_i} \ldots \quad M^{175} \xleftarrow{ecol_{n-1}} M^n$$

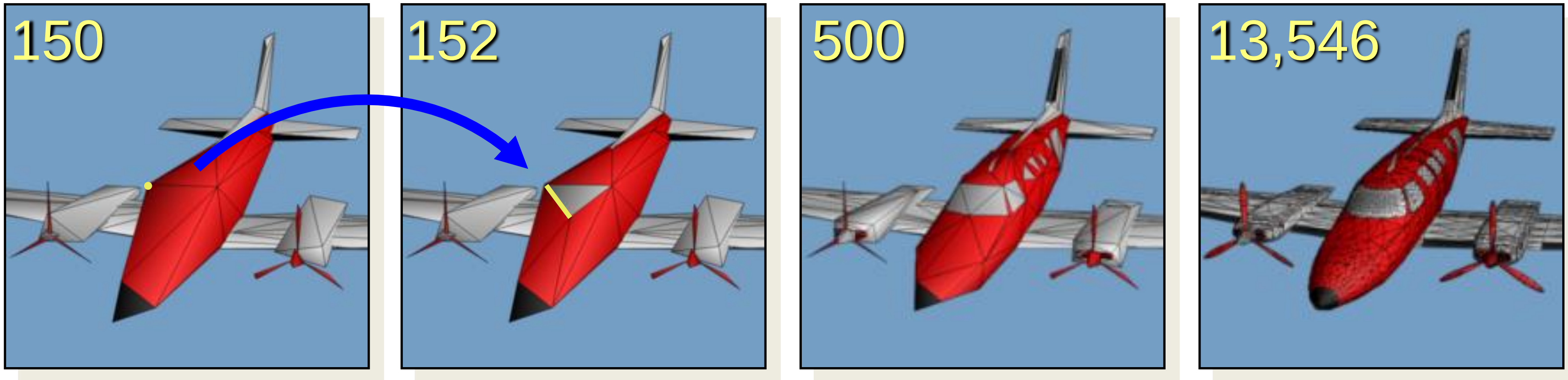
$M^0$ $\quad$ $M^i$ $\quad$ $M^n = \hat{M}$

## Reconstruction: Vertex Split

- Invertible & lossless!



# Reconstruction: Vertex Split

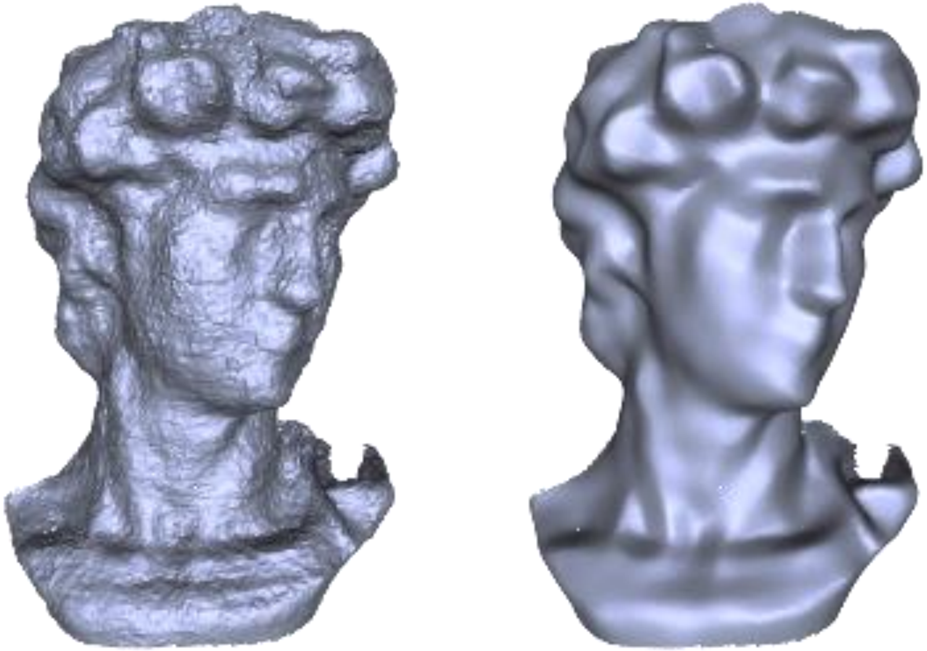



$$M^0 \xrightarrow[vspl_0]{} M^1 \;\ldots\; \xrightarrow[vspl_i]{} \;\ldots\; M^{175} \xrightarrow[vspl_{n-1}]{} M^n$$

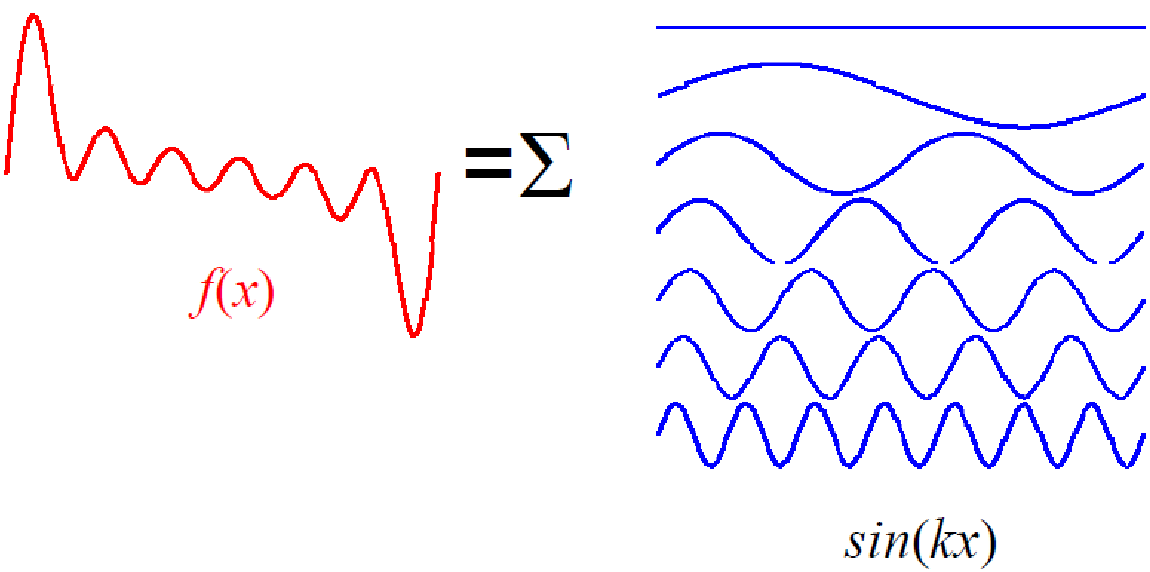
$M^0$ $\longleftrightarrow$ $M^n = \hat{M}$

$M^i$ $\longrightarrow$

**Input**: Noisy mesh (scanned or other)
**Output**: Smooth mesh
**How**: Filter out high frequency noise

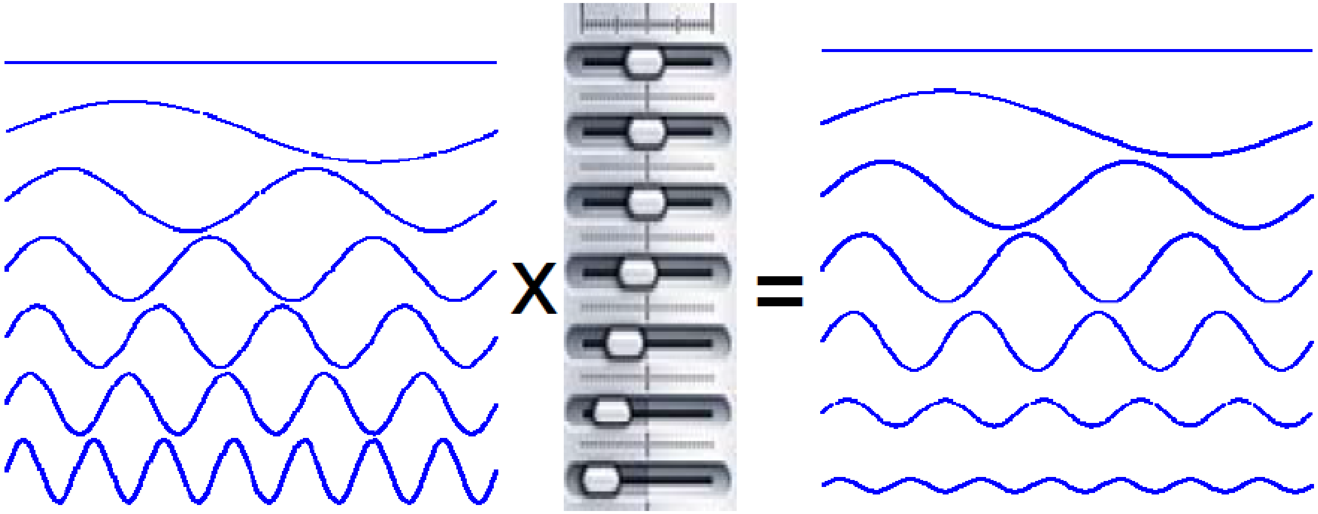


# Fourier Transform

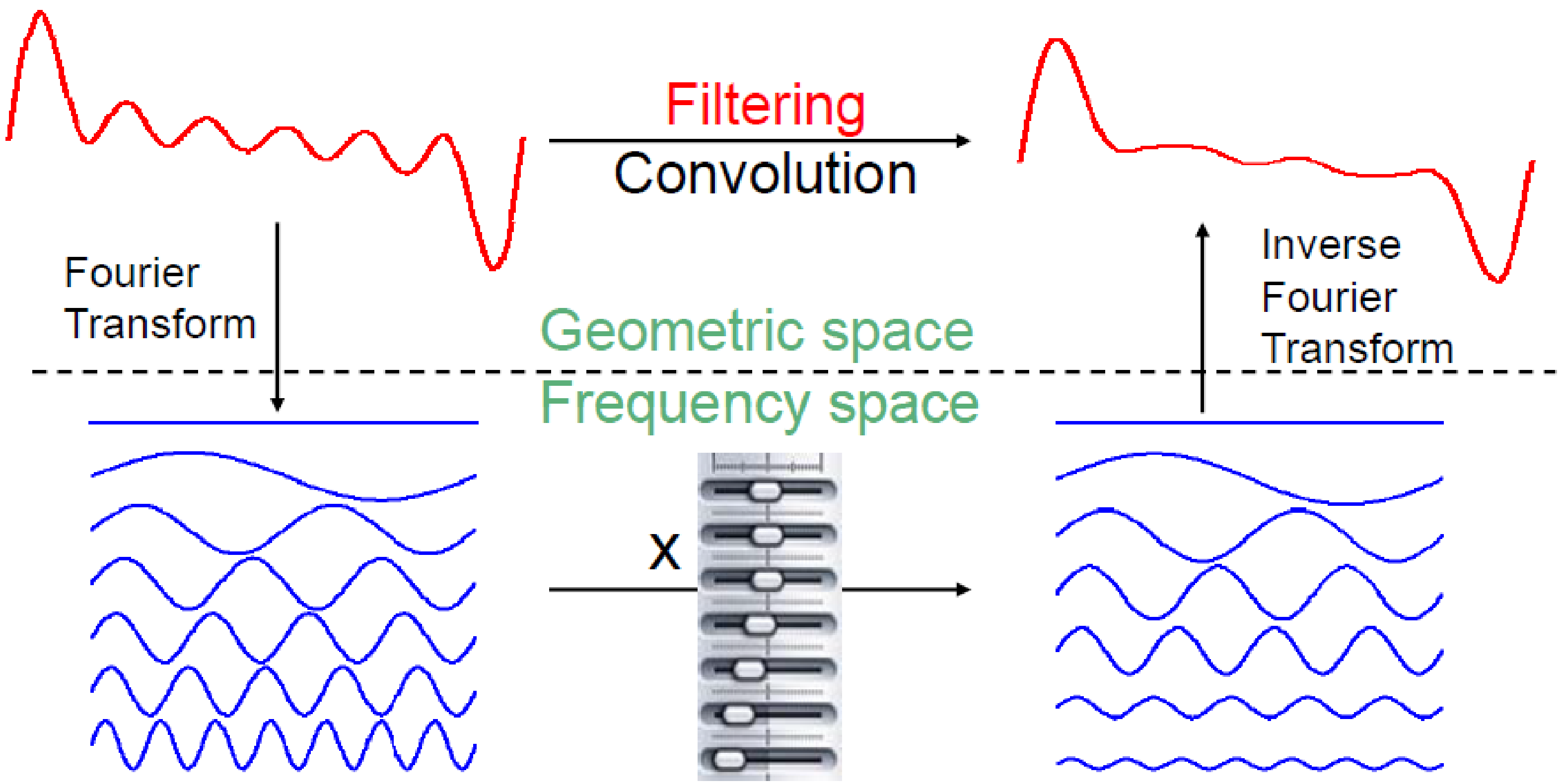



Slides by Levy et al., SigAsia Course 2009

# Fourier Transform

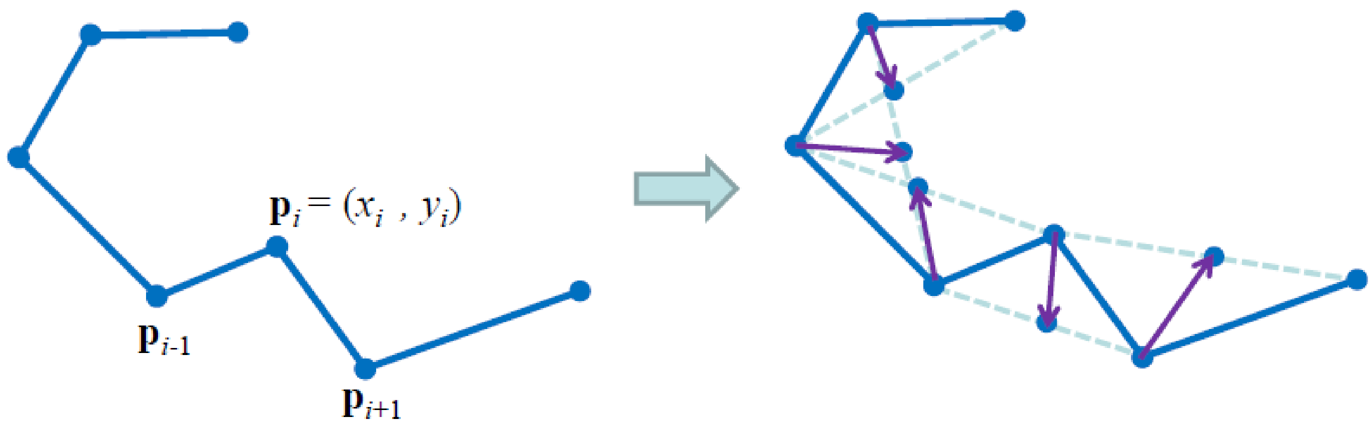



Slides by Levy et al., SigAsia Course 2009




Slides by Levy et al., SigAsia Course 2009

# An easier problem: How to smooth a curve?

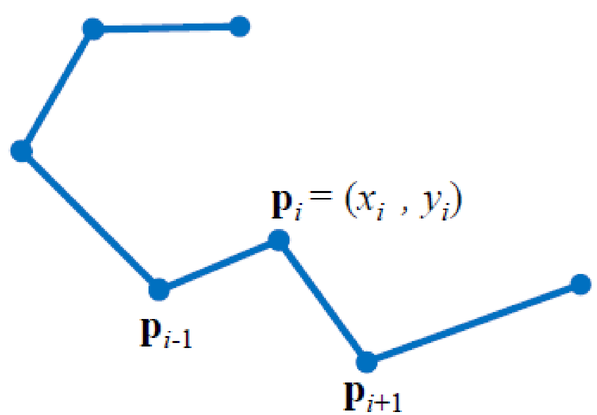


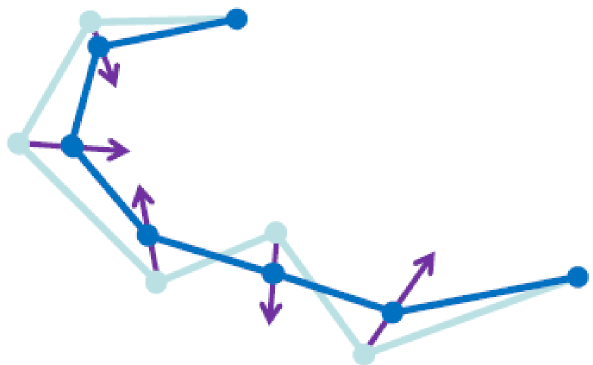

$$(\mathbf{p}_{i\text{-}1} + \mathbf{p}_{i+1})/2 - \mathbf{p}_i$$

$$L(\mathbf{p}_i) = \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_i) + \frac{1}{2}(\mathbf{p}_{i-1} - \mathbf{p}_i)$$

UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE

# An easier problem: How to smooth a curve?



$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \frac{1}{2} L(\mathbf{p}_i)$$

Finite difference discretization of second derivative
= Laplace operator in one dimension

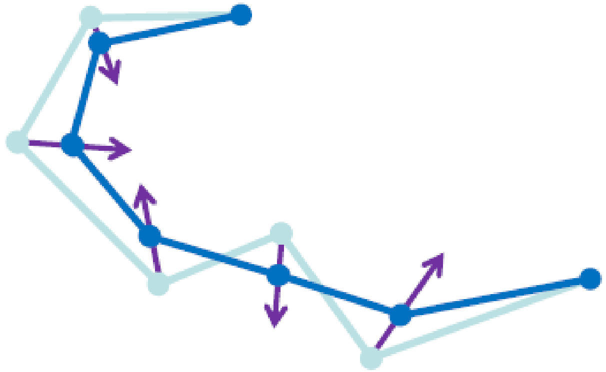
$$L(\mathbf{p}_i) = \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_i) + \frac{1}{2}(\mathbf{p}_{i-1} - \mathbf{p}_i)$$

UNIVERSITÉ PARIS-EST CRÉTEIL VAL DE MARNE

Algorithm:

Repeat for *m* iterations (for non boundary points):

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \lambda L(\mathbf{p}_i)$$



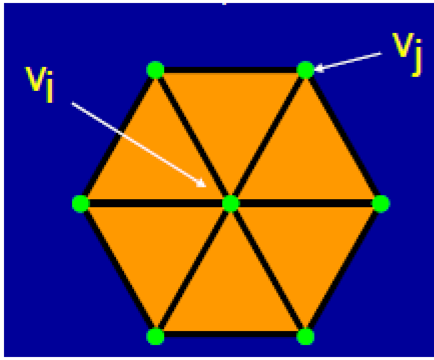
For which λ?

$0 < \lambda < 1$

Closed curve converges to?

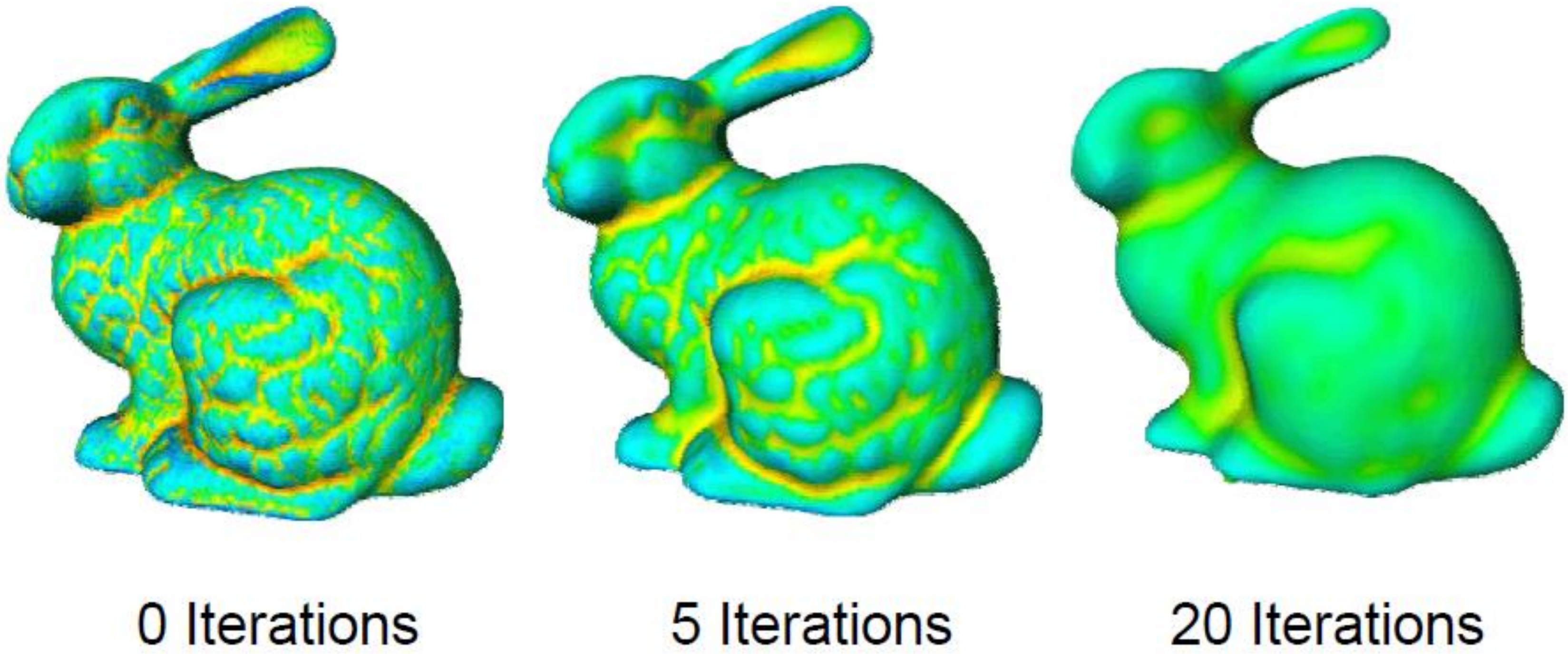
Single point

- Keep boundary vertices fixed
- Move each internal vertex to the barycenter of its neighbors

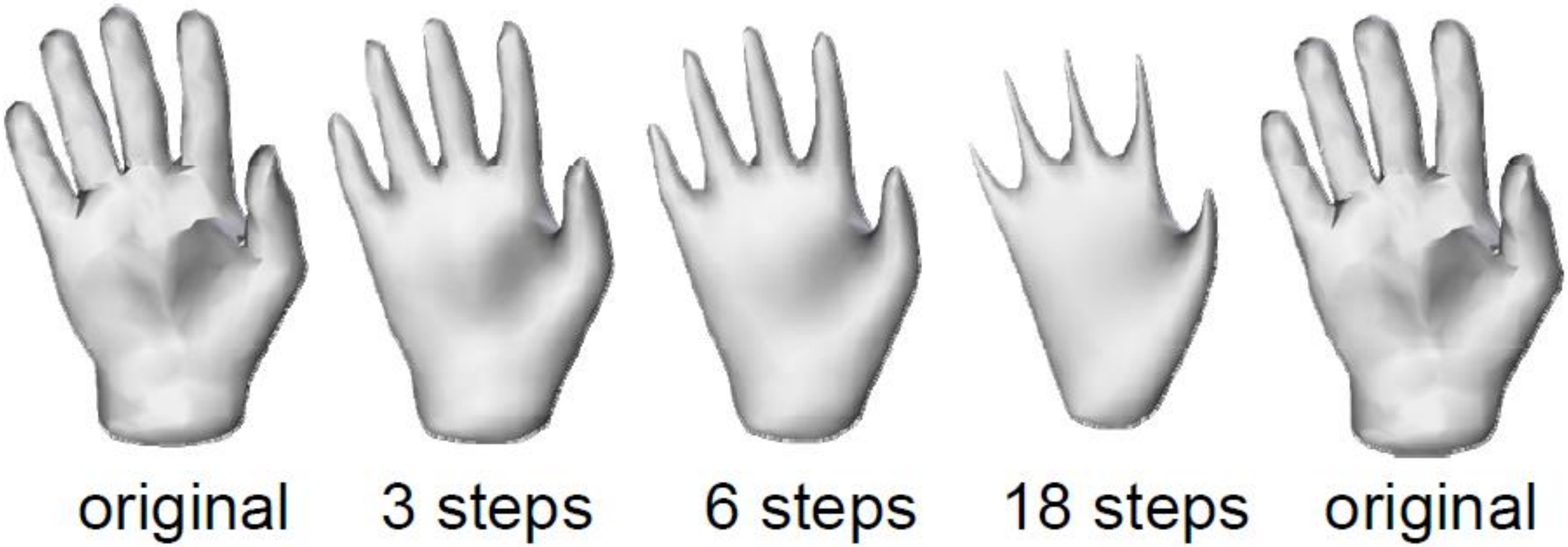



$$v_i' = \frac{1}{n_i} \sum_{j \in i^*} v_j$$

$$v_i' = v_i + \lambda \Delta v_i$$


0 Iterations
5 Iterations
20 Iterations

Repeated iterations of Laplacian smoothing shrinks the mesh



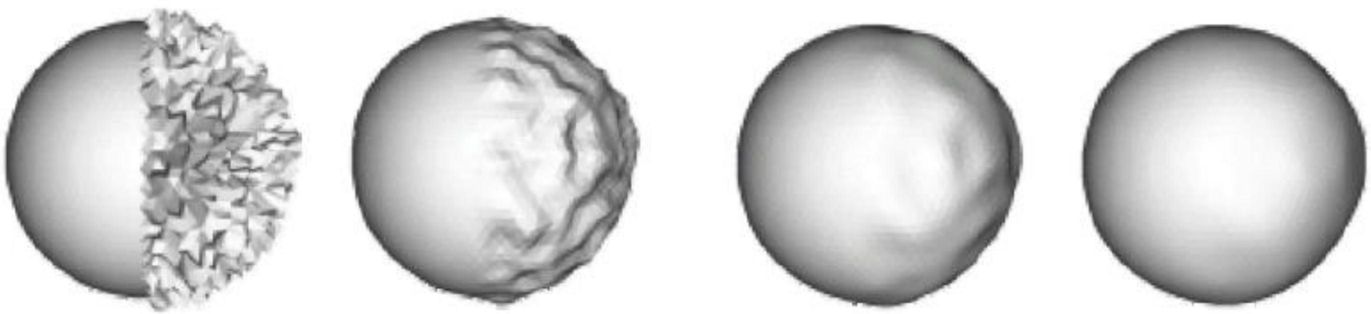
# Taubin Smoothing

Iterate:

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \lambda \Delta \mathbf{p}_i \qquad \text{Shrink}$$

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \mu \Delta \mathbf{p}_i \qquad \text{Inflate}$$

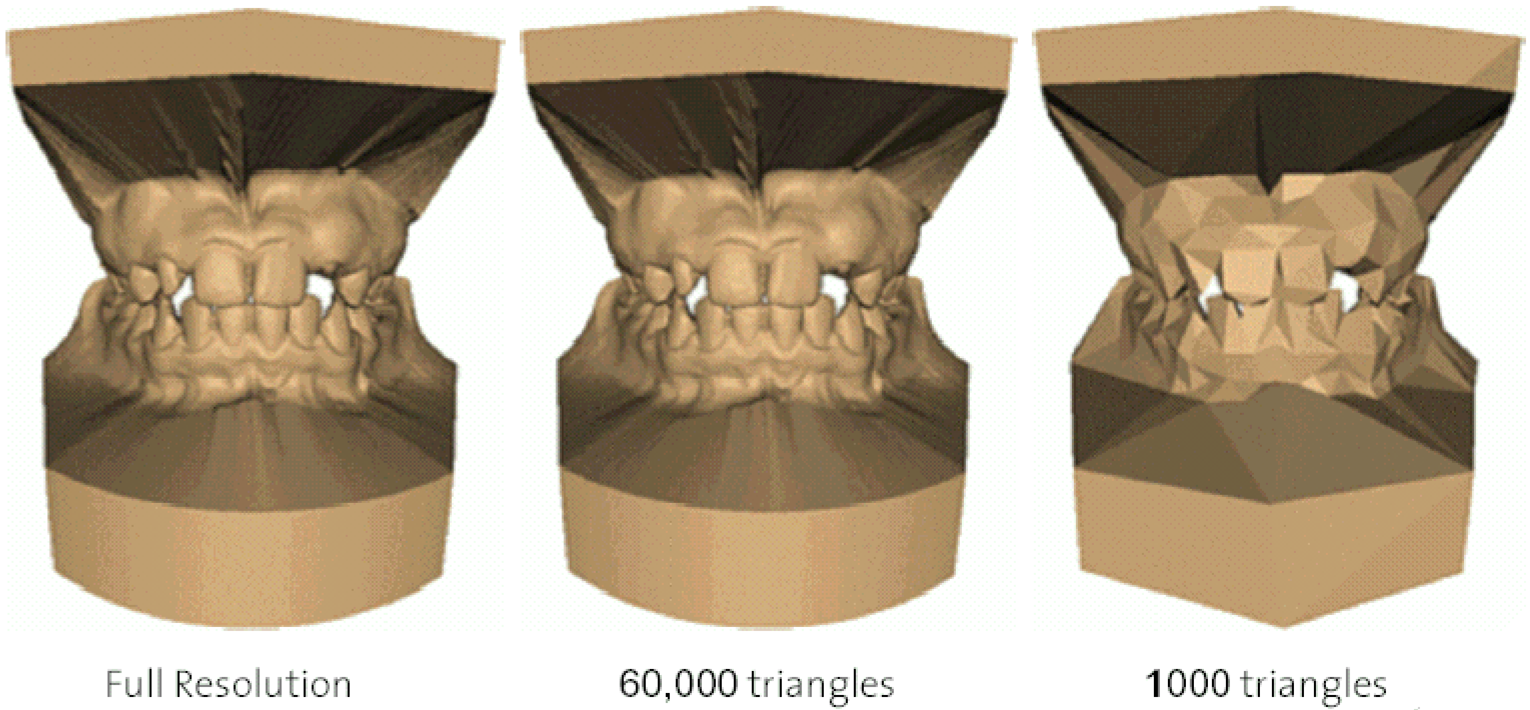
with $\lambda > 0$ and $\mu < 0$



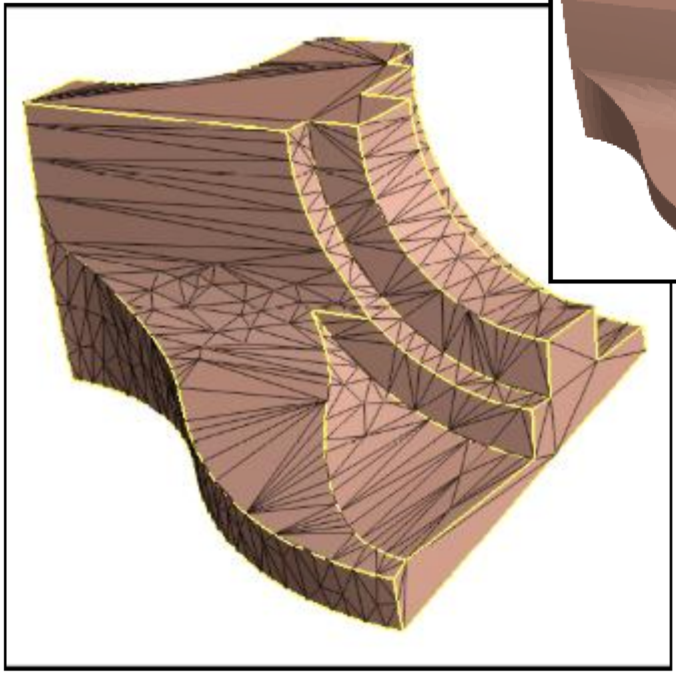
original 10 steps 50 steps 200 steps

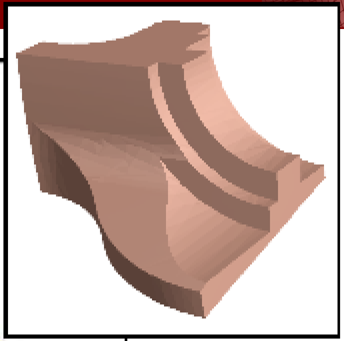
From Taubin, Siggraph 1995

# Mesh Simplification



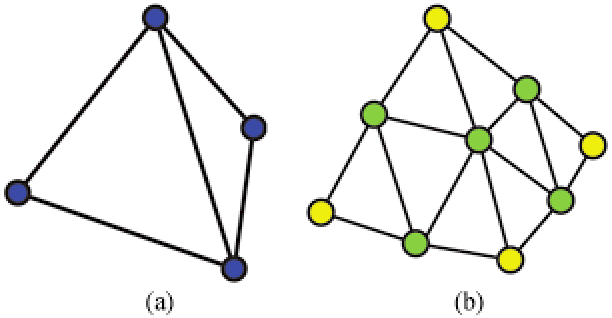
## Application – Mesh compression

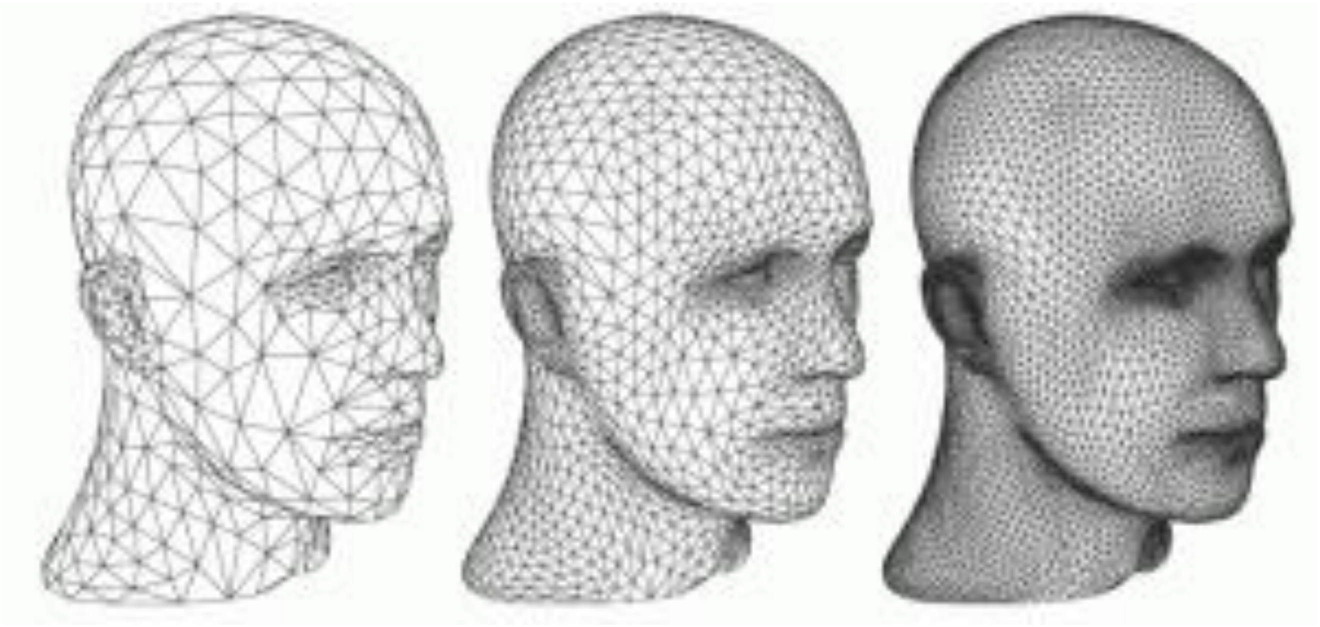




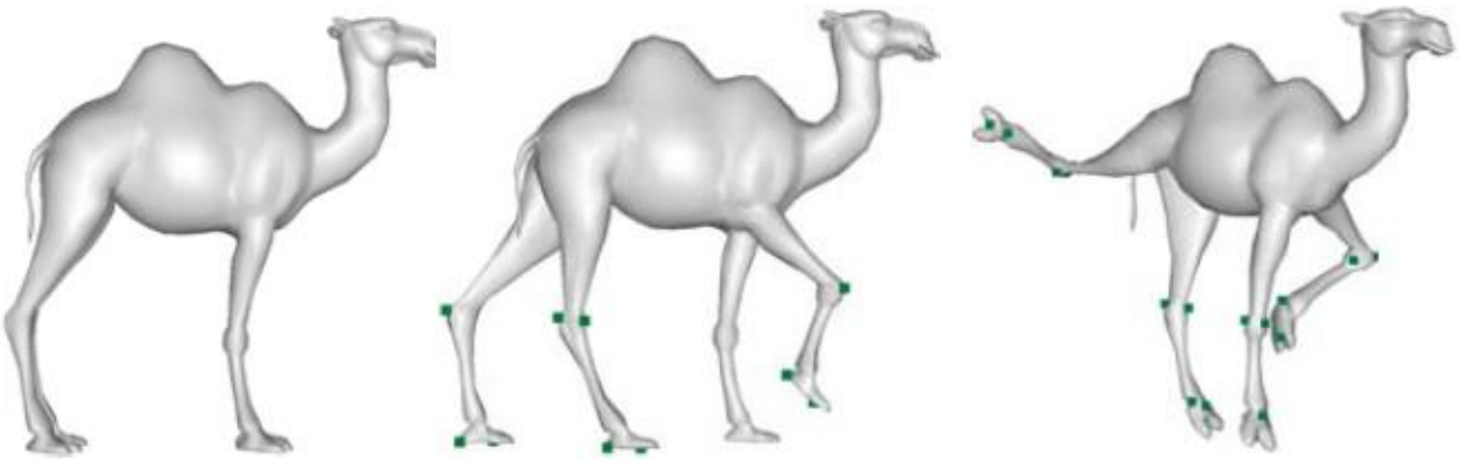
12964 faces → 1000 faces

# Mesh Subdivision

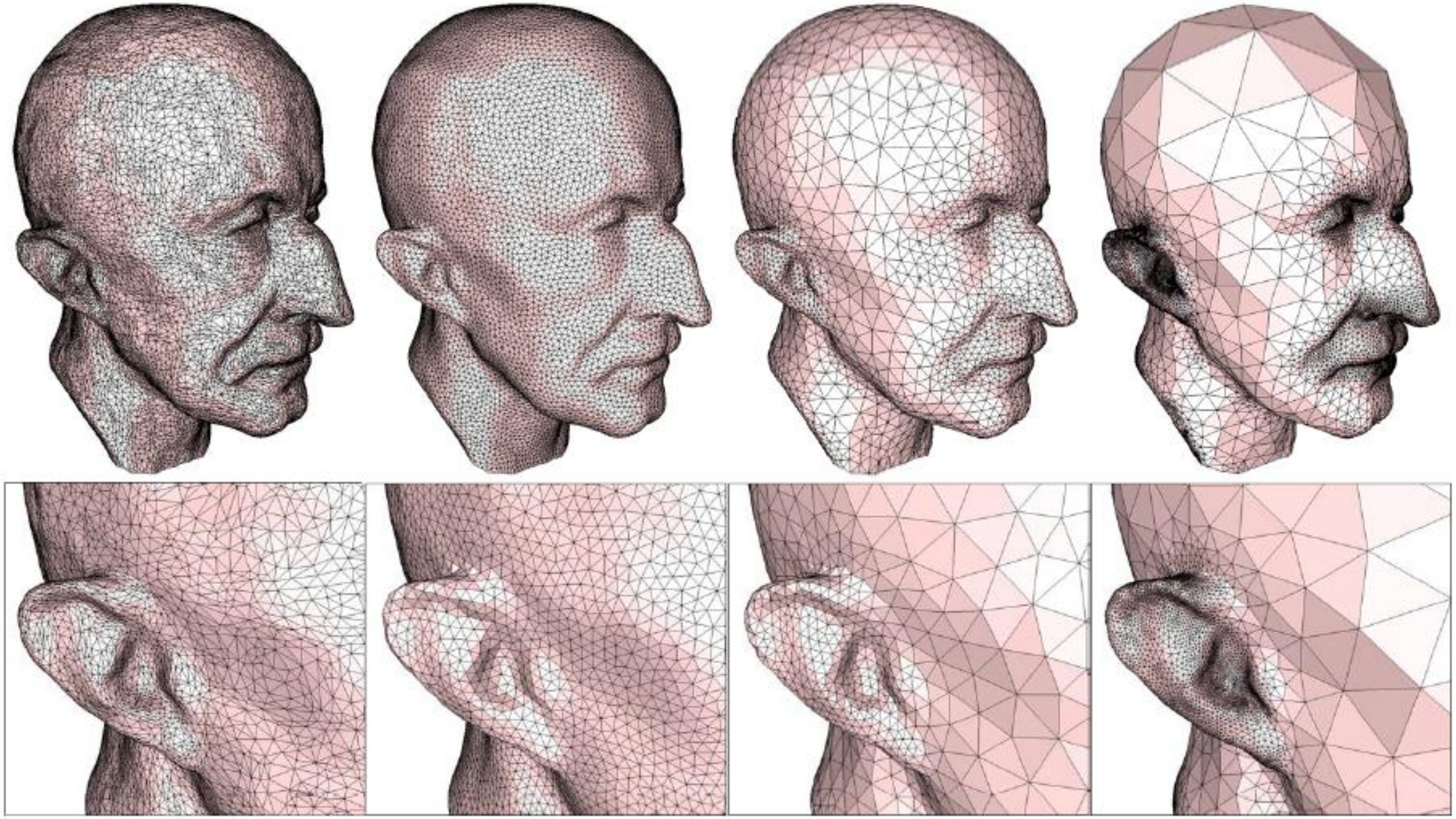


# Mesh Deformation

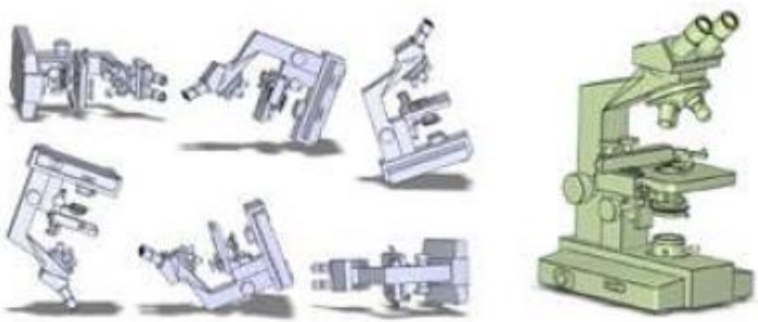


# Remeshing

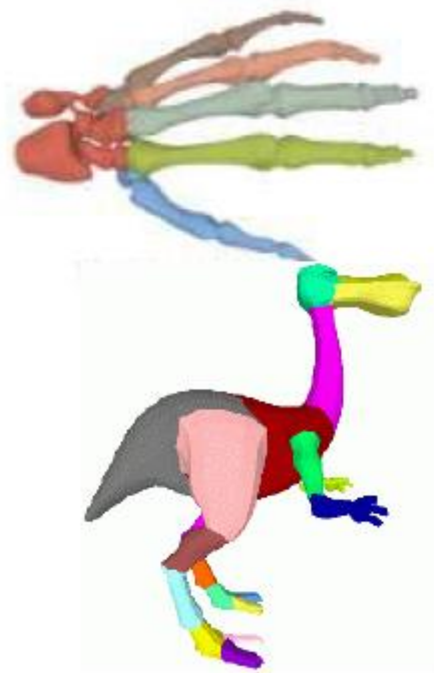


# Mesh Analysis

Matching



Orientation/View Selection



Segmentation

OpenMesh:

- OpenMesh web page
- OpenMesh documentation

Mesh manipulation:

- MeshLab
- Graphite

Models:

- AIM@SHAPE Repository
- Princeton Segmentation Database (82M)

Library

- CGAL
- PCL

# References

- **Book**

  "Polygon Mesh Processing"

  *by Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, Bruno Levy*

- **Eurographics 2008 course notes**

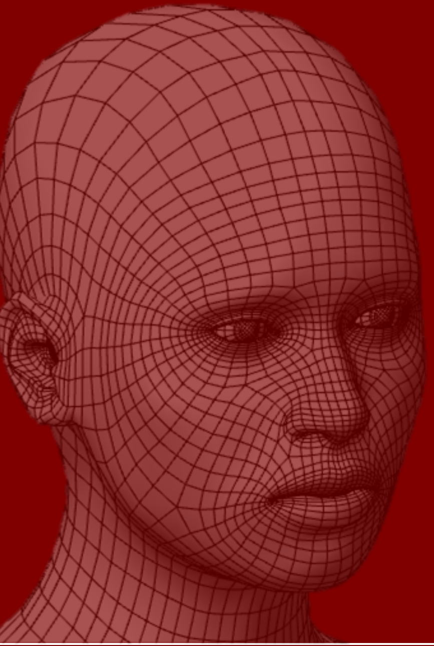
  **"**Geometric Modeling Based on Polygonal Meshes"

  *by Mario Botsch, Mark Pauly, Leif Kobbelt, Pierre Alliez, Bruno Levy, Stephan Bischoff, Christian Rössl*

- **Tutorials and papers**

  "Polygonal Mesh –Data Structure and Processing" *by Chiew-Lan Tai*

- **Download Meshlab** http://www.meshlab.net/#download
- Download Meshes from learning platform
- Fun begin, play with meshes !!



# Thank you for your attention

UPEC
Connaissance - Action
UNIVERSITÉ
PARIS-EST CRÉTEIL
VAL DE MARNE