



3D Modeling Methods



000

Modeling methods: Objective

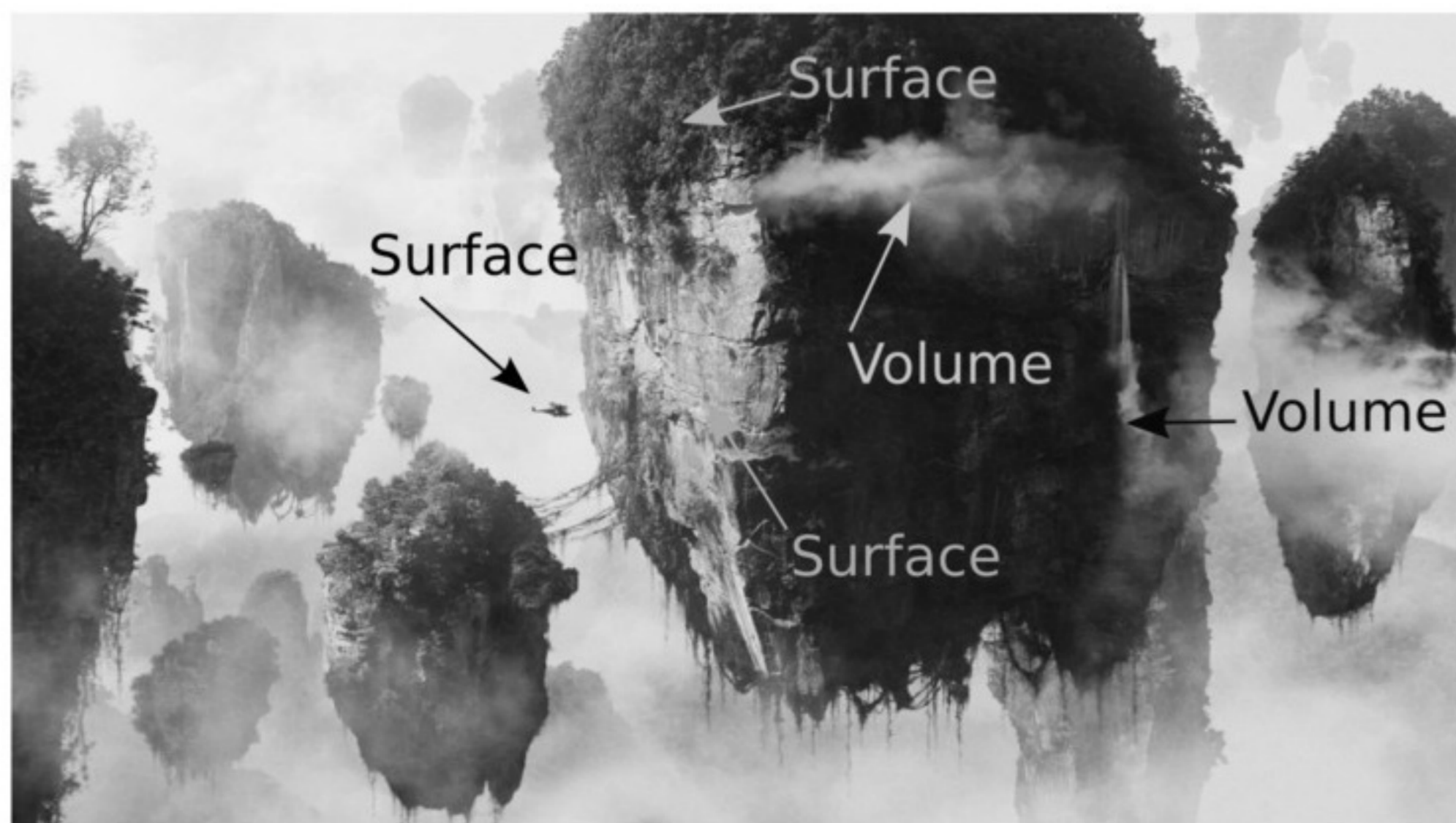
- Understand how to model a 3D object
- Which model is the most suitable for a given application



001

How do we model a 3D object

1. Surface modeling VS Volume modeling
2. How do we encode it on a computer



[Avatar]

002

3D model realism

Which are the virtual objects ?



[Day After Tomorrow]



[Lord of the Rings]

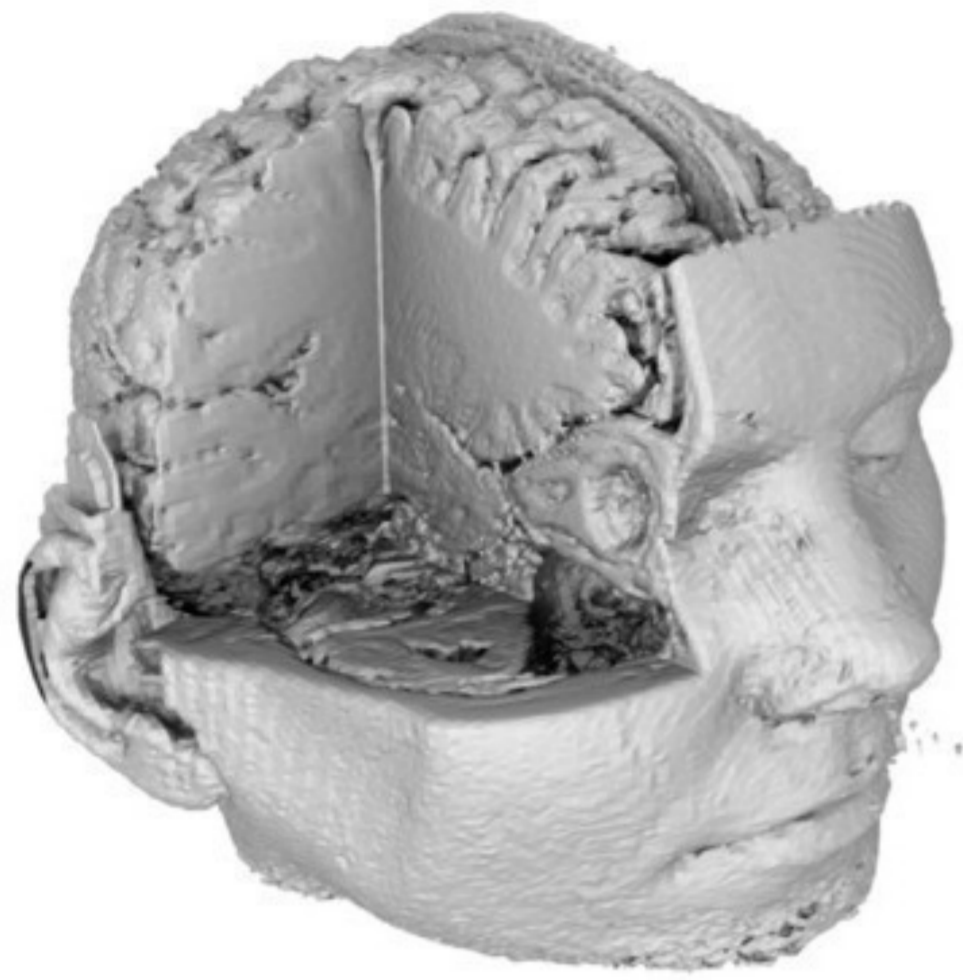


[Titanic]

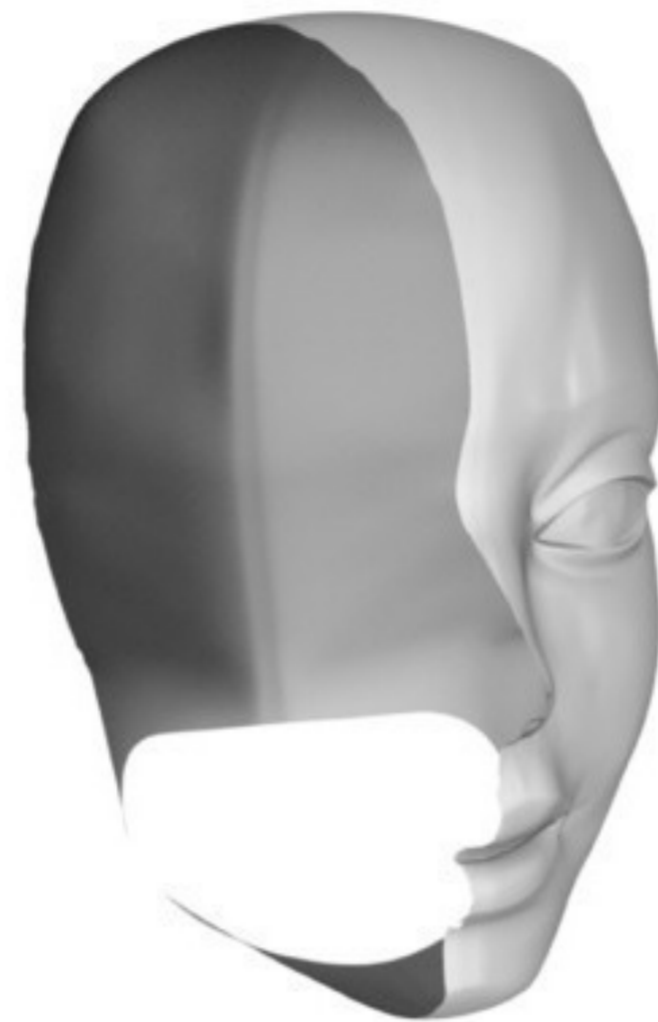
003

Modeling a 3D object

Volume modeling



Surface modeling



004

Representing a Surface

Explicit

BRep: *Boundary Representation*

Mesh

Parametric

Subdivision

CSG: *Constructive Solid Geometry*

Implicit

Voxels

Parametric

Skeleton

Analytic

Point-Sets

MLS

Surfels

Fractals

005

Explicit representation

Mesh

Parametric

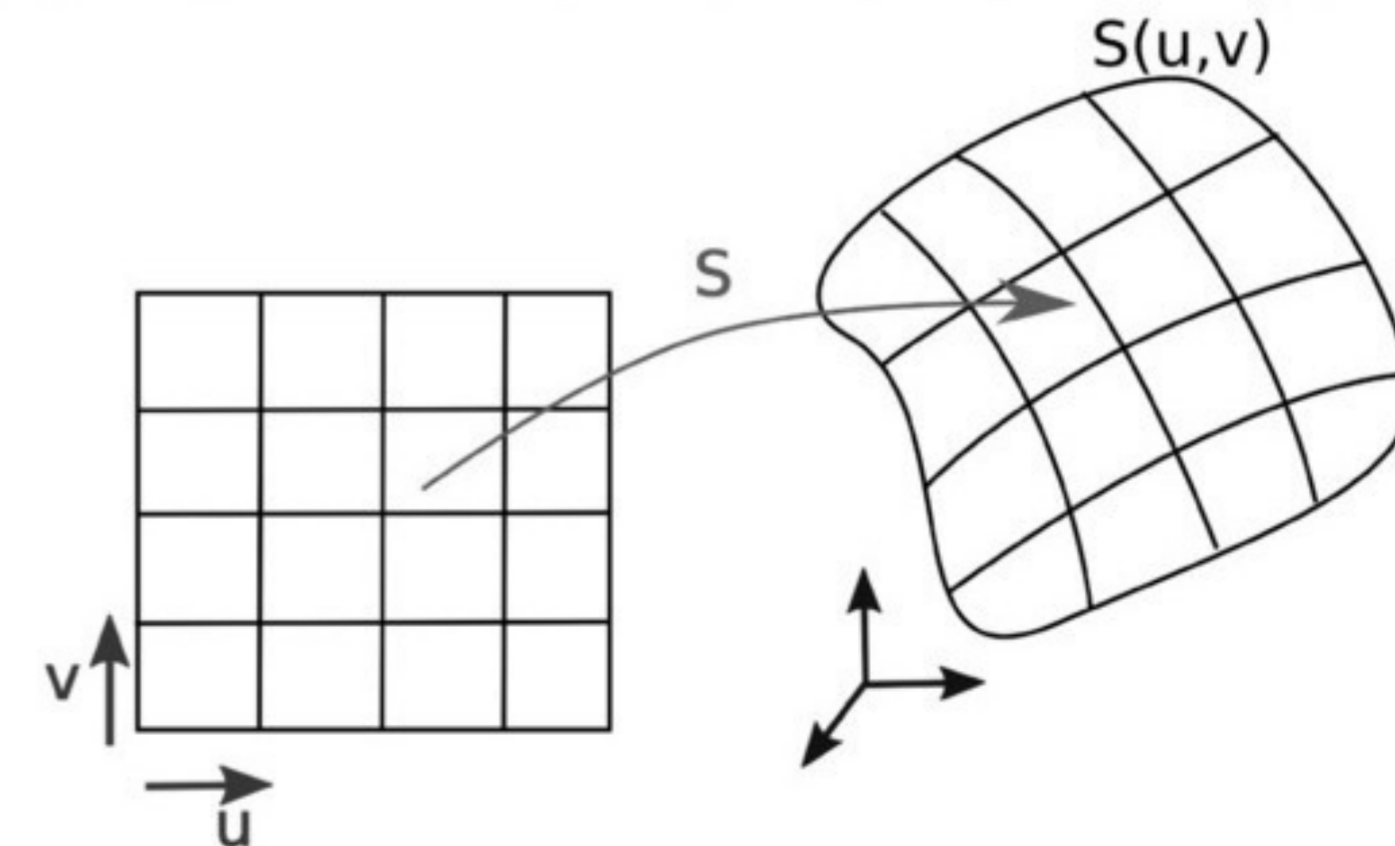
Subdivision

006

Explicit representation

Explicit representation ~ parametric

$$S: \begin{cases} \mathcal{D} \subset \mathbb{R}^2 & \rightarrow \mathbb{R}^3 \\ (u, v) & \mapsto S(u, v) = (S_x(u, v), S_y(u, v), S_z(u, v)) \end{cases}$$



$$S = \text{mapping} \neq \text{Surface } \Gamma$$

trace of S in \mathbb{R}^3

007

Triangular Mesh

Triangular mesh = Simplest BRep

When S is not known analytically

We compute a discrete local approximation

$$S = \bigcup_i S_i$$

Simplest mapping : Linear mapping

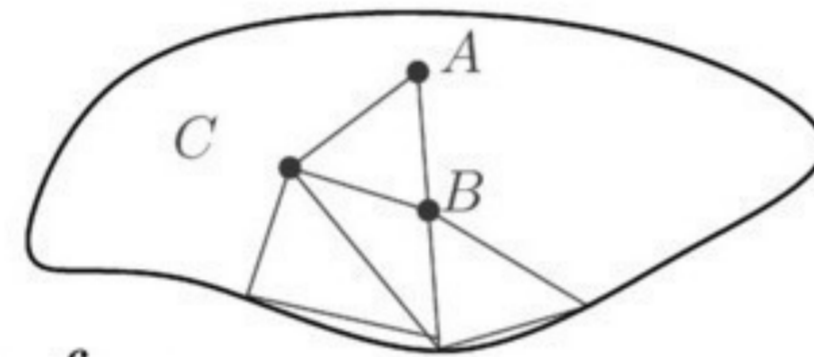
$$S_i : \begin{cases} \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3 \\ (u, v) \mapsto S_i(u, v) = u\vec{AB} + v\vec{AC} + \vec{OA} \end{cases}$$

$$\mathcal{D} : (u, v) \in [0, 1]^2, 0 \leq u + v \leq 1$$

S is globally \mathcal{C}^0

S is never \mathcal{C}^1 (exc. plane)

S can interpolate any discrete set of points on a surface



008

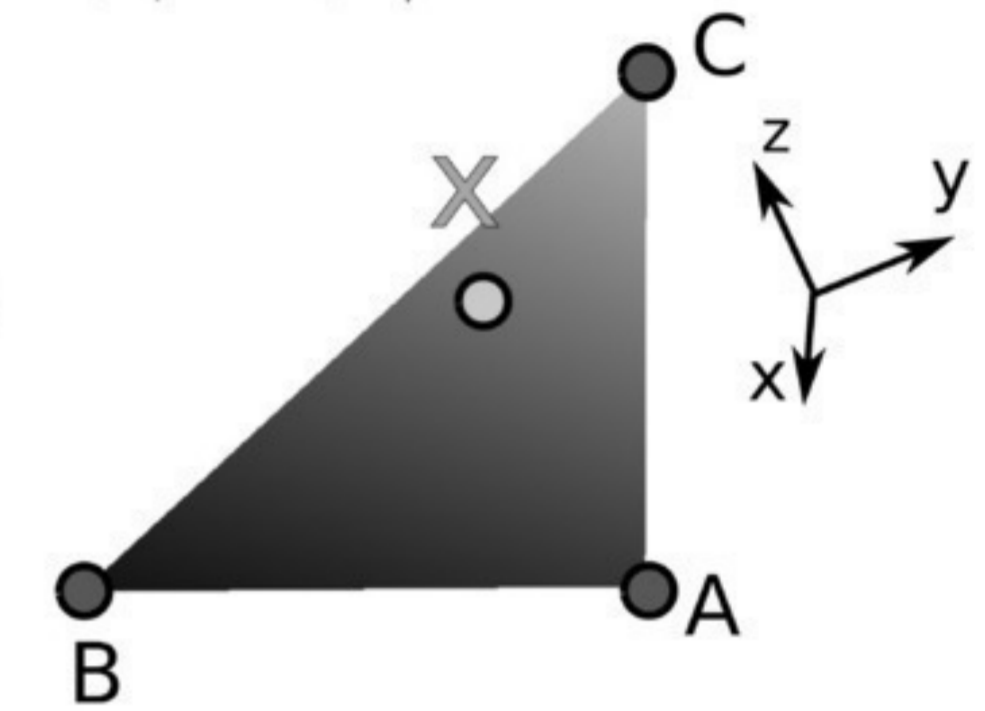
Barycentric Coordinates

$$(u, v) \mapsto S_i(u, v) = u\vec{AB} + v\vec{AC} + \vec{OA}$$

$$(\alpha, \beta, \gamma) \mapsto S'_i(\alpha, \beta, \gamma) = \alpha A + \beta B + \gamma C$$

$$\alpha + \beta + \gamma = 1$$

$$(\alpha, \beta, \gamma) \in [0, 1]^3$$



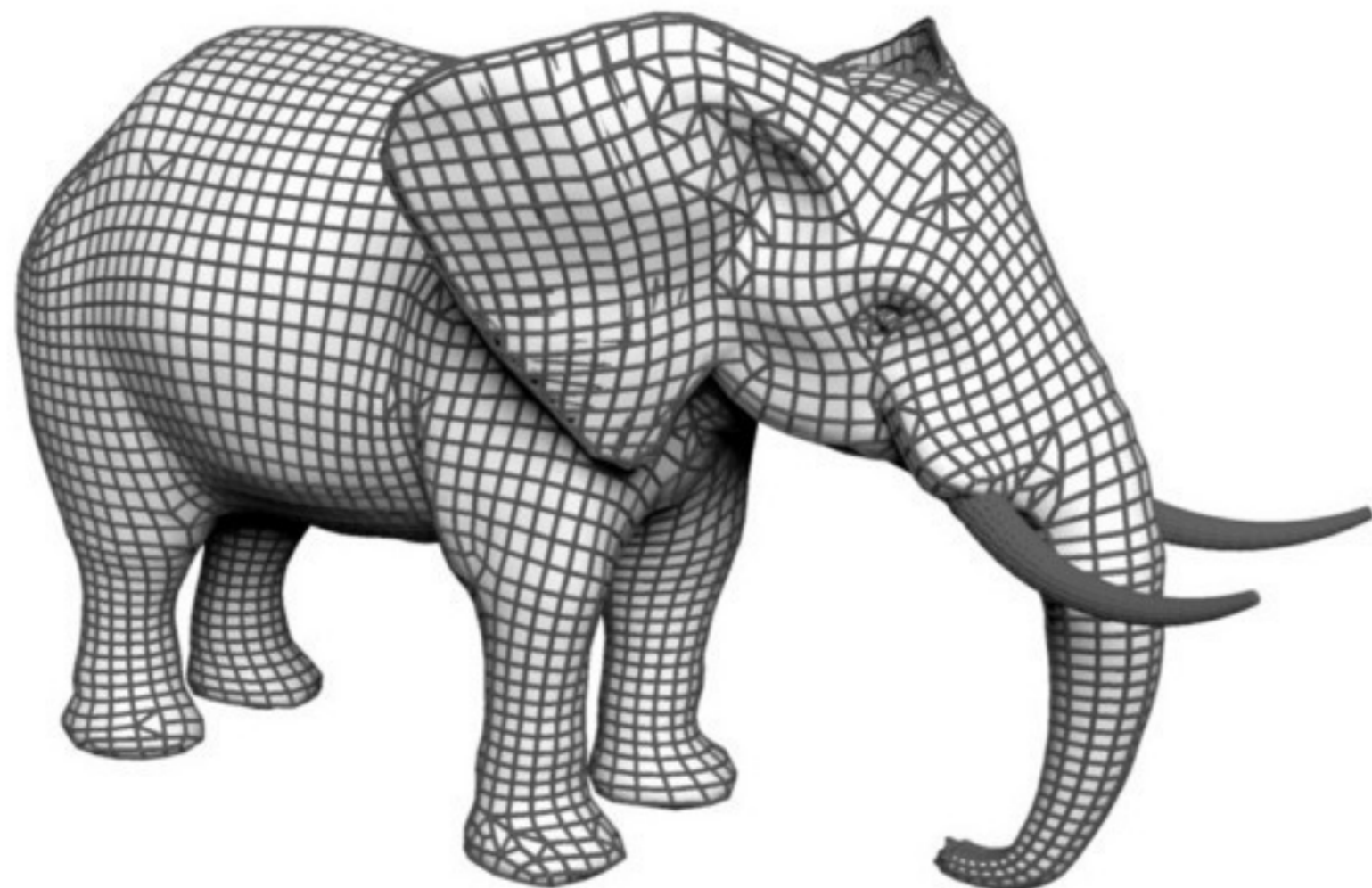
009

Mesh

Special case:

A mesh can have polygons made of N vertices ($N > 3$)

A good polygon should have N coplanar vertices (otherwise, need triangulation)



010

Mesh : Approximation

S : true smooth surface

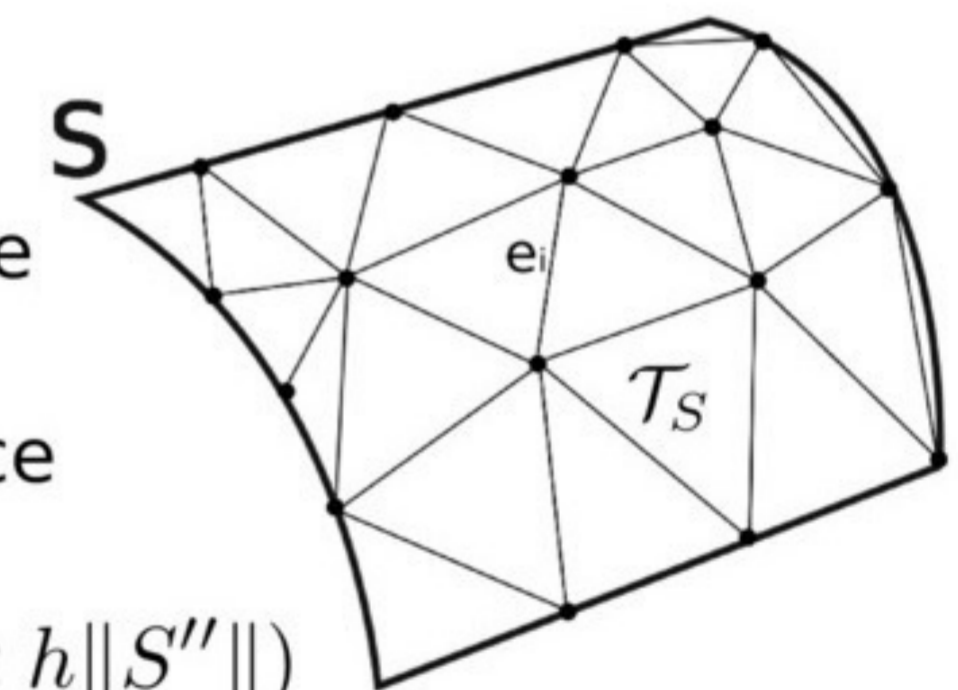
\mathcal{T}_S : triangulated surface

$$\|S - \mathcal{T}_S\| = h |\kappa_{\max}| (\simeq h \|S''\|)$$

Linear approximation of a surface S (first order)

$$h = K \max_i e_i$$

$$\Rightarrow \|S - \mathcal{T}_S\| = \mathcal{O} \left(\max_i e_i |\kappa_{\max}| \right)$$



011

Mesh : Volume

Linear element in volume = **Tetrahedron**

$$\mathbf{x} = u(\mathbf{x}_1 - \mathbf{x}_0) + v(\mathbf{x}_2 - \mathbf{x}_0) + w(\mathbf{x}_3 - \mathbf{x}_0) + \mathbf{x}_0$$

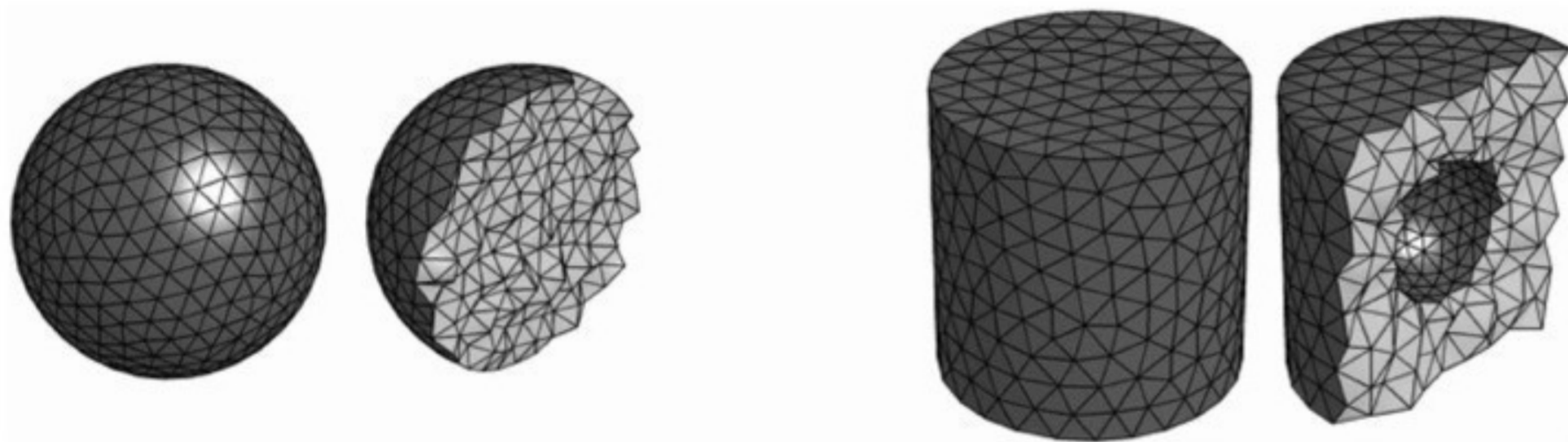
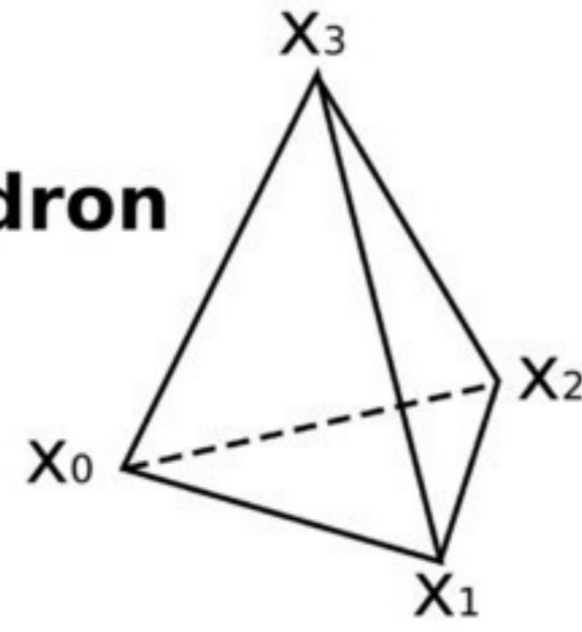
$$0 < u + v + w < 1$$

$$(u, v, w) \in [0, 1]^3$$

$$\mathbf{x} = \alpha\mathbf{x}_0 + \beta\mathbf{x}_1 + \gamma\mathbf{x}_2 + \delta\mathbf{x}_3$$

$$\alpha + \beta + \gamma + \delta = 1$$

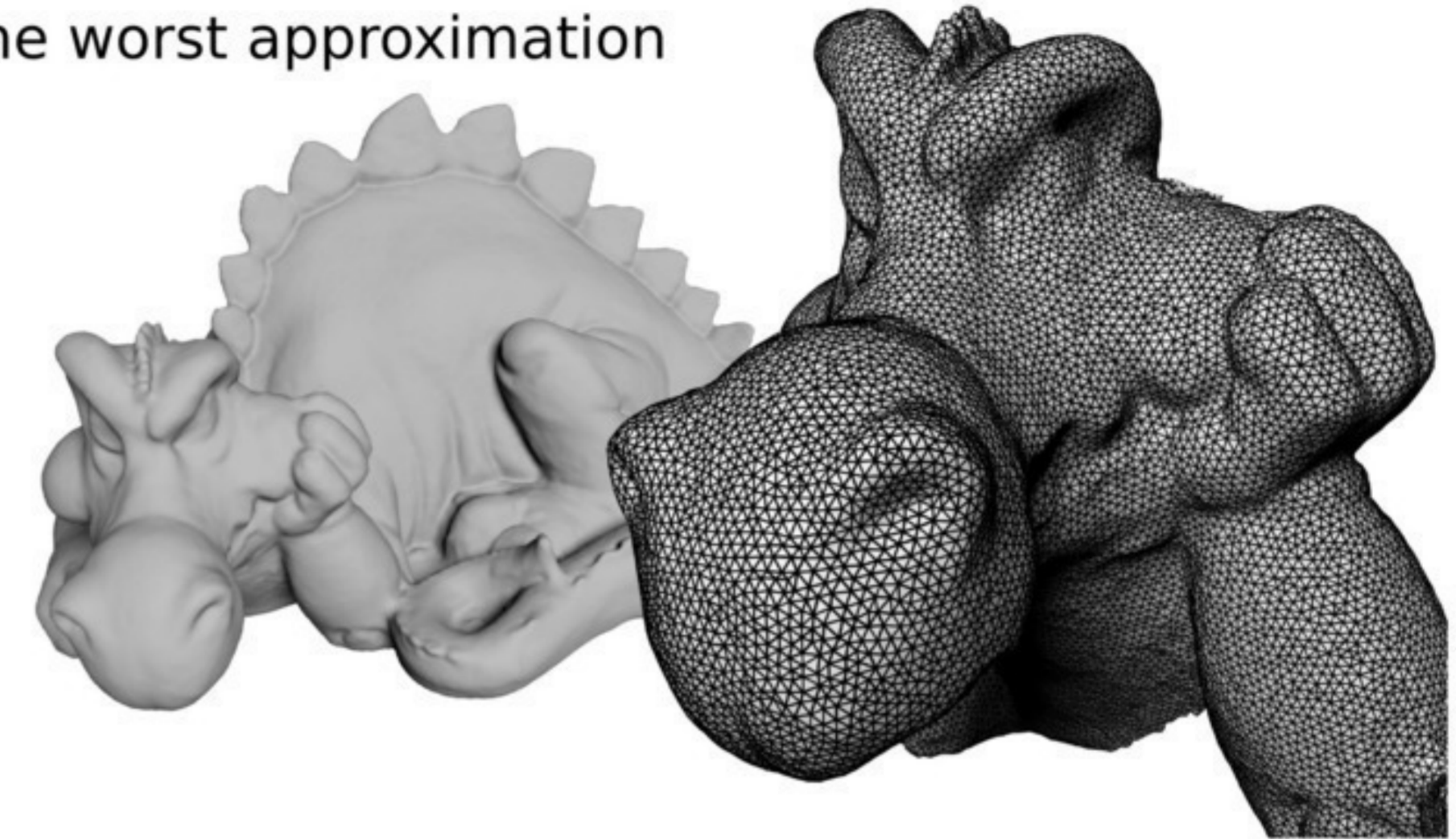
$$(\alpha, \beta, \gamma, \delta) \in [0, 1]^3$$



012

Mesh, conclusion

- + The simplest
- + The most versatile
- + The most common
- + Fast dedicated rendering
- The worst approximation



013

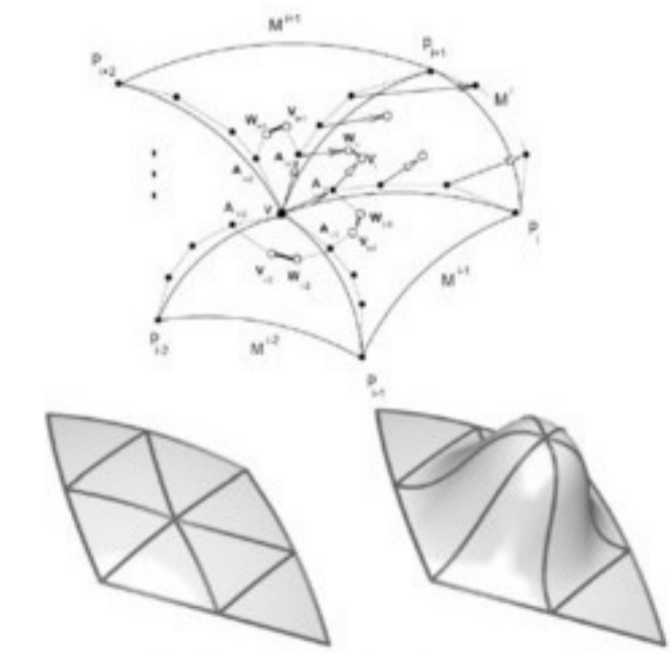
Explicit representation

Mesh
Parametric
 Subdivision

014

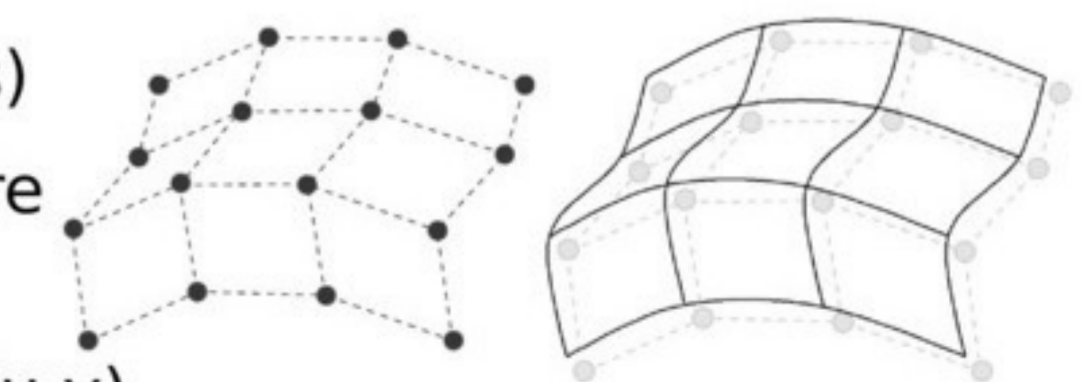
Parametric

- Order > 1
- Idea 1
 - Derivative data (smooth triangles)
 - Problem: Difficult to set-up
 => **Not so used**



[Yvart, Hahmann, 01-04]

- Idea 2
 - Add extra vertices (non triangular patches)
 - Problem: Patch structure
 - Common case: Rectangular patches (u,v)



=> **Very common**

015

Parametric : Spline Patches

.Patch (4x4) : bicubic polynomial

.Parametric surface C^2
(Continuous curvature)

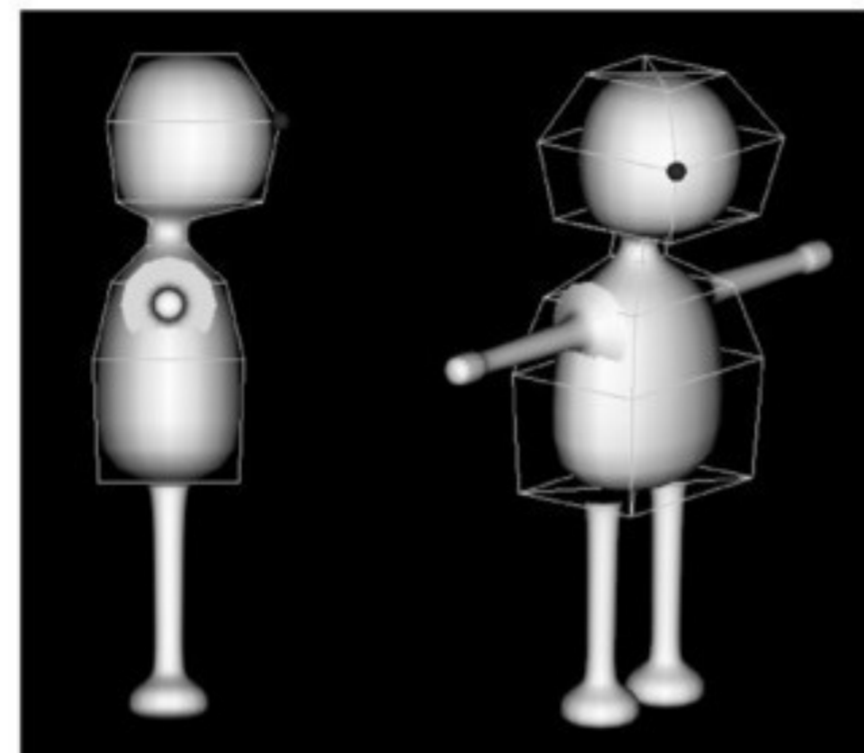
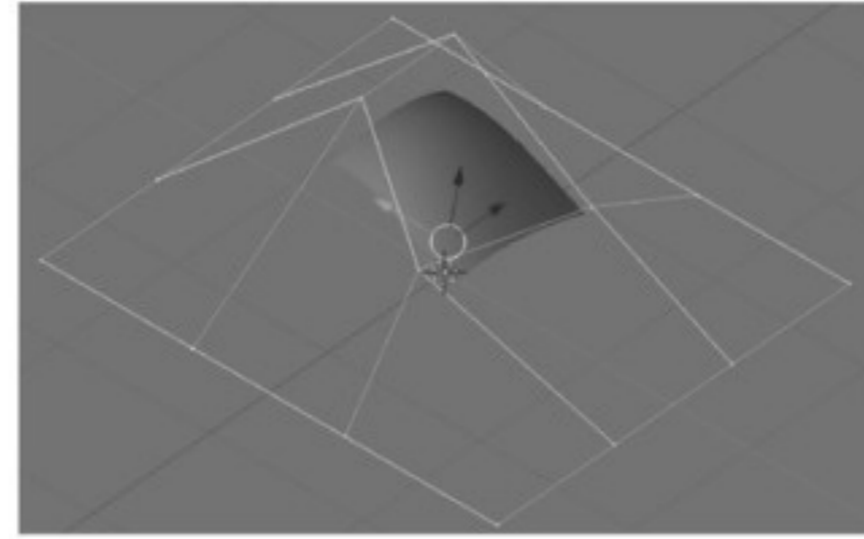
$$S(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 b_i(u) b_j(v) P_{ij}$$

(tensorial product surface)

.Special case
(uniform knot vector)

$$S(u, v) = (u^3 u^2 u 1) M [P_{ij}] M^T (v^3 v^2 v 1)^T$$

$$M = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$



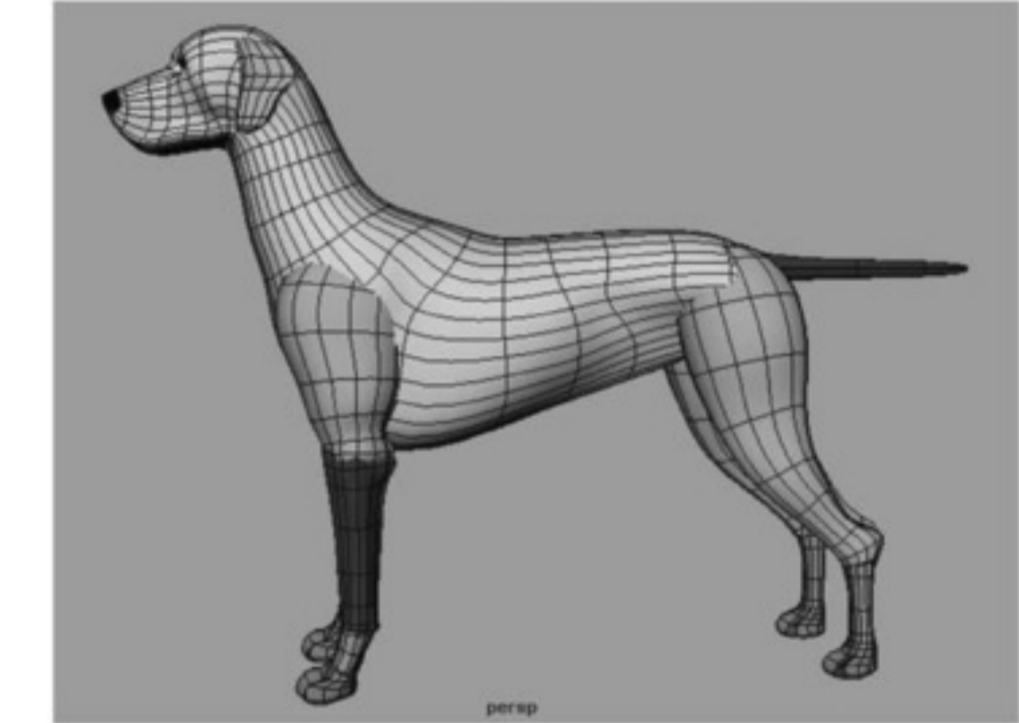
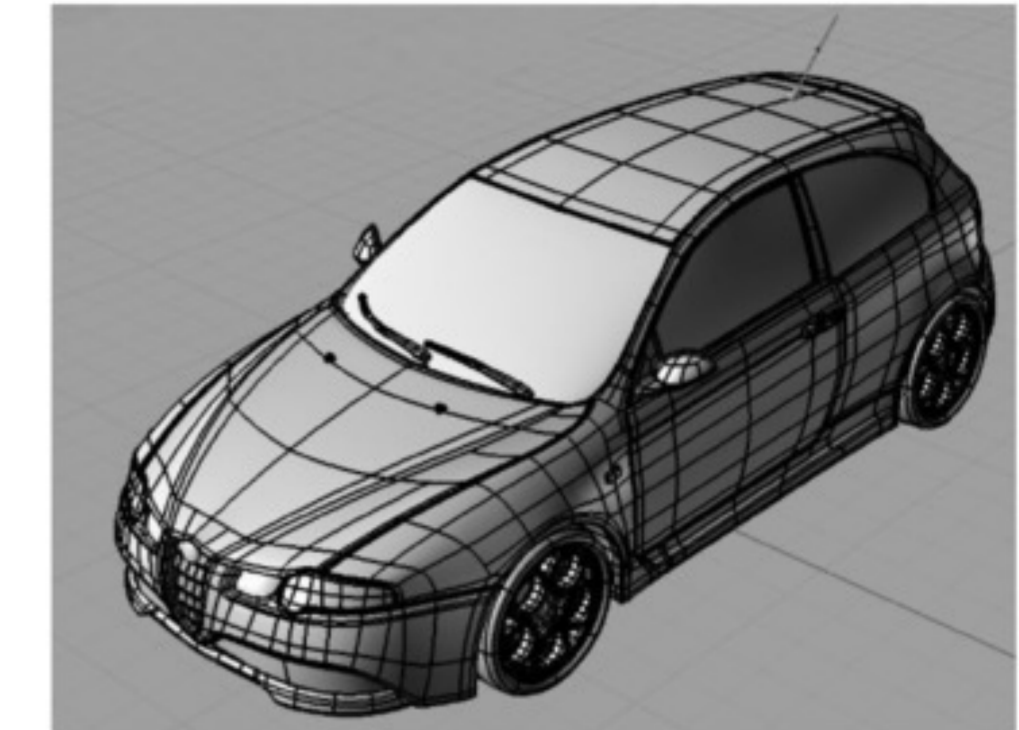
016

Parametric, Conclusion

+ Smooth Surface (CAD)

- Patch Structure

Manual modeling
Technical
Junction



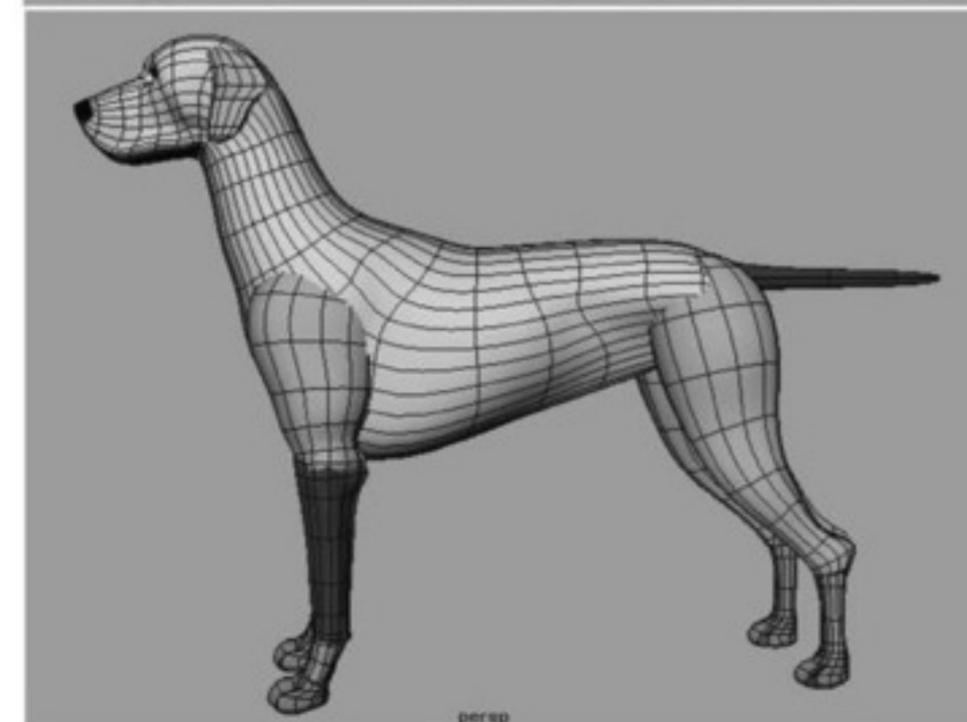
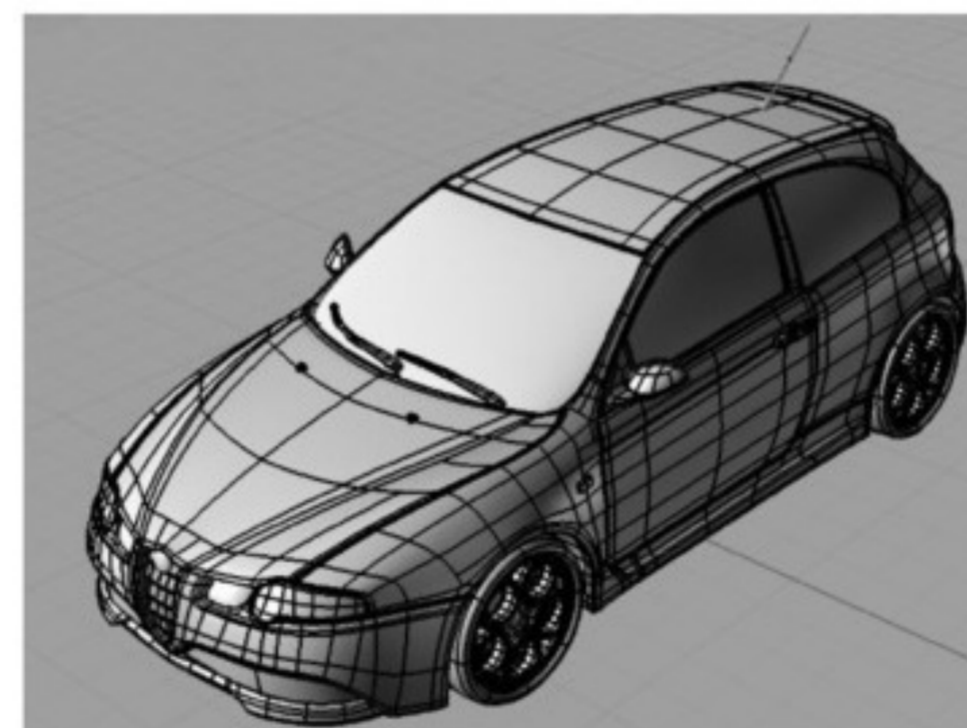
017

Parametric, Conclusion

+ Smooth Surface (CAD)

- Patch Structure

Manual modeling
Technical
Junction



018

Explicit representation

Mesh
Parametric
Subdivision

019

Subdivision surface

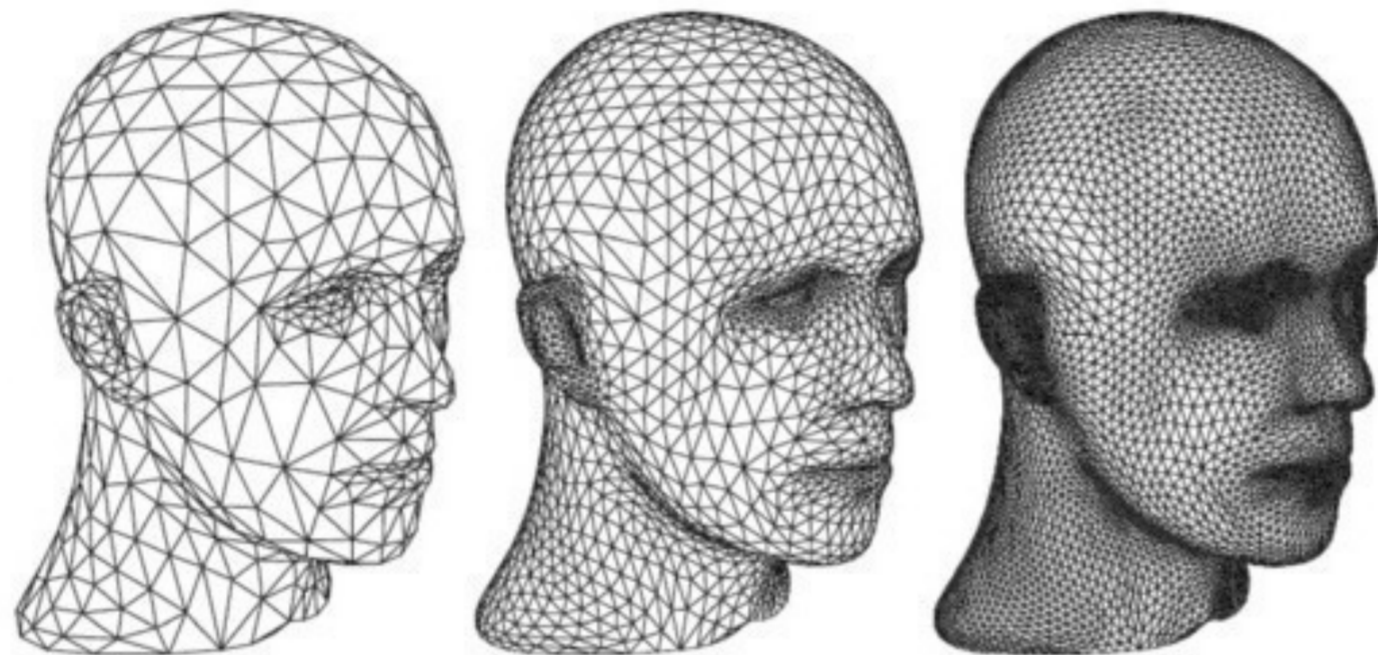
Answer the problem + Smooth surface
+ Local control
+ Arbitrary topology

.Principle in 1D (subdivision curve)



.Principle in 2D

2D subdivision mask



[Zorin, Schroeder, SIGGRAPH Course Notes 99]

020

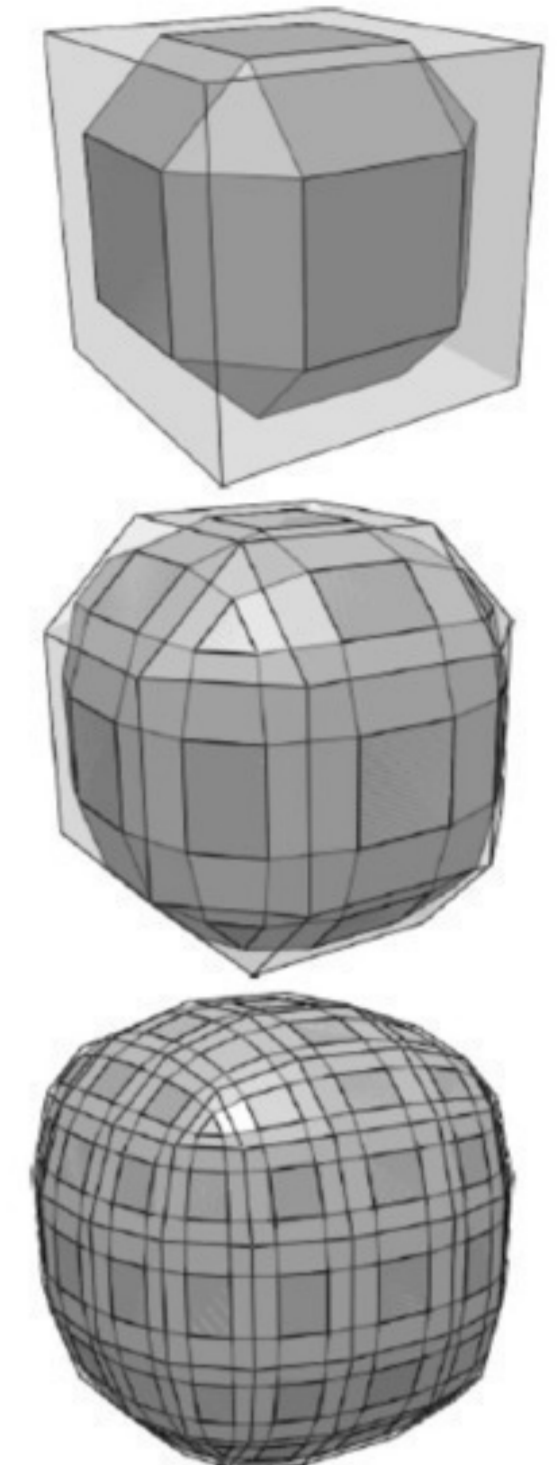
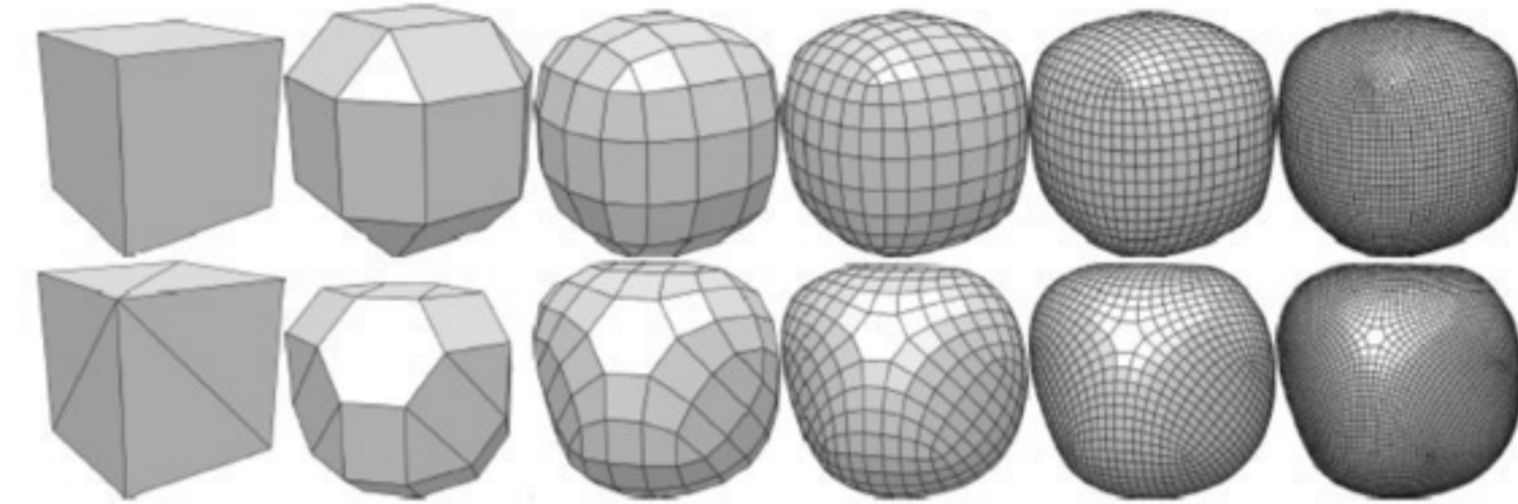
Subdivision surface, construction

Step 0: Control polygon P^0
Step 1: Subdivision $P^0 \rightarrow P^1$

...
Step i : Subdivision $P^{i-1} \rightarrow P^i$

...
Limit surface $S = \lim_{i \rightarrow +\infty} P^i$

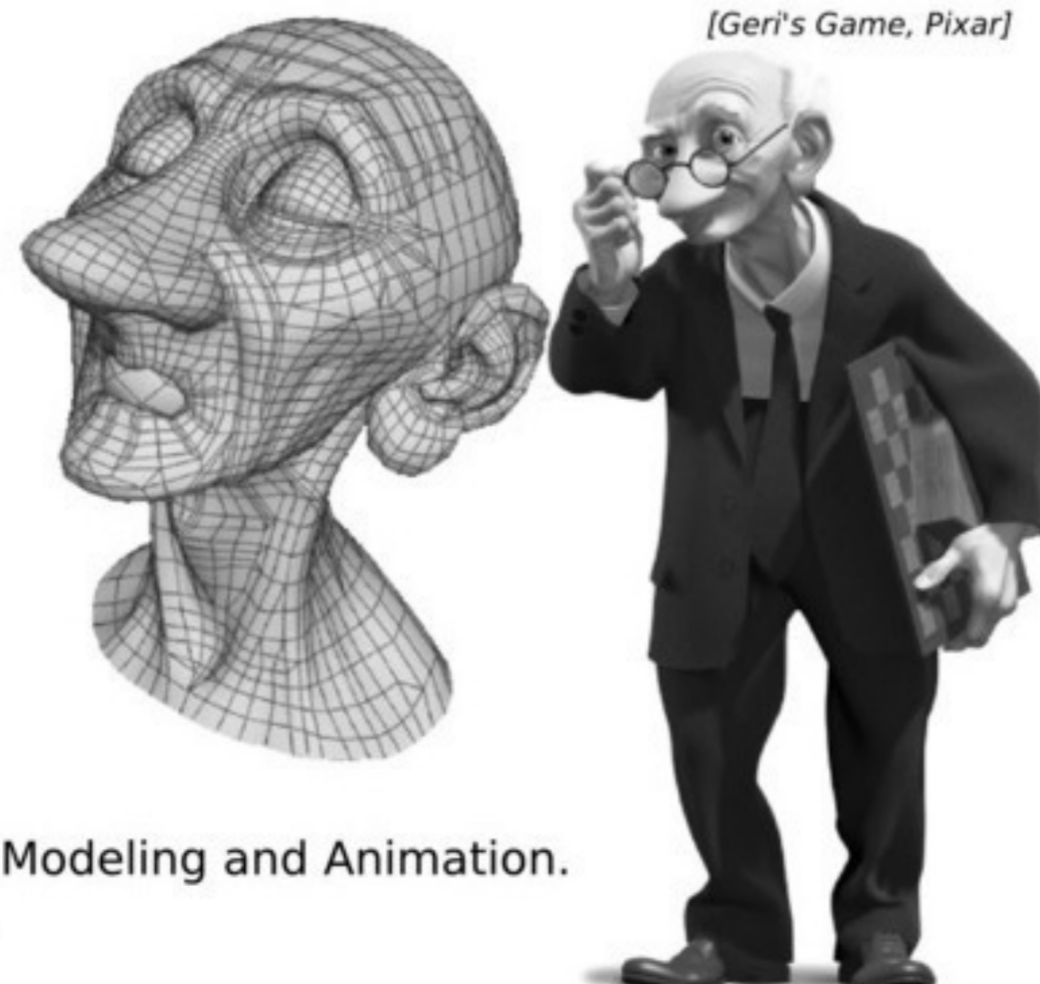
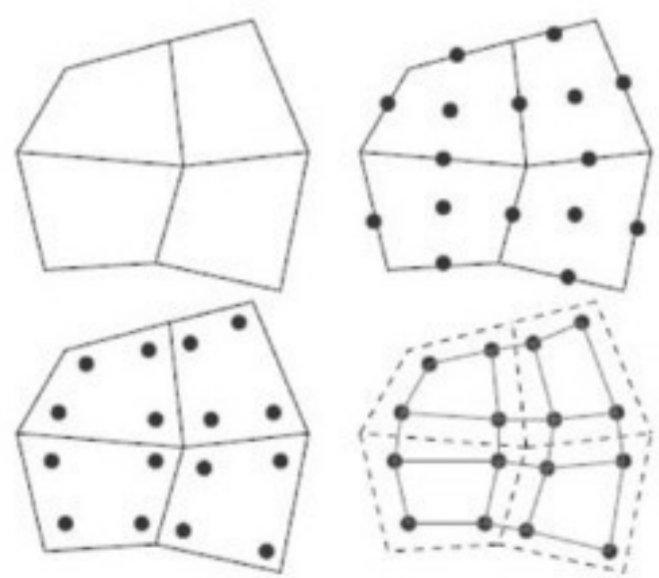
S can be C^2 almost everywhere



021

Subdivision scheme

Loop: triangles, C^2 alm. ev., approximation
Catmull Clark: quads, C^2 alm. ev., approximation
Doo-Sabin (corner cutting): quads, C^2 alm. ev., approximation
Butterfly: triangles, C^1 alm. ev., interpolation.
Kobbelt: triangles, C^2 alm. av., approximation



[Geri's Game, Pixar]

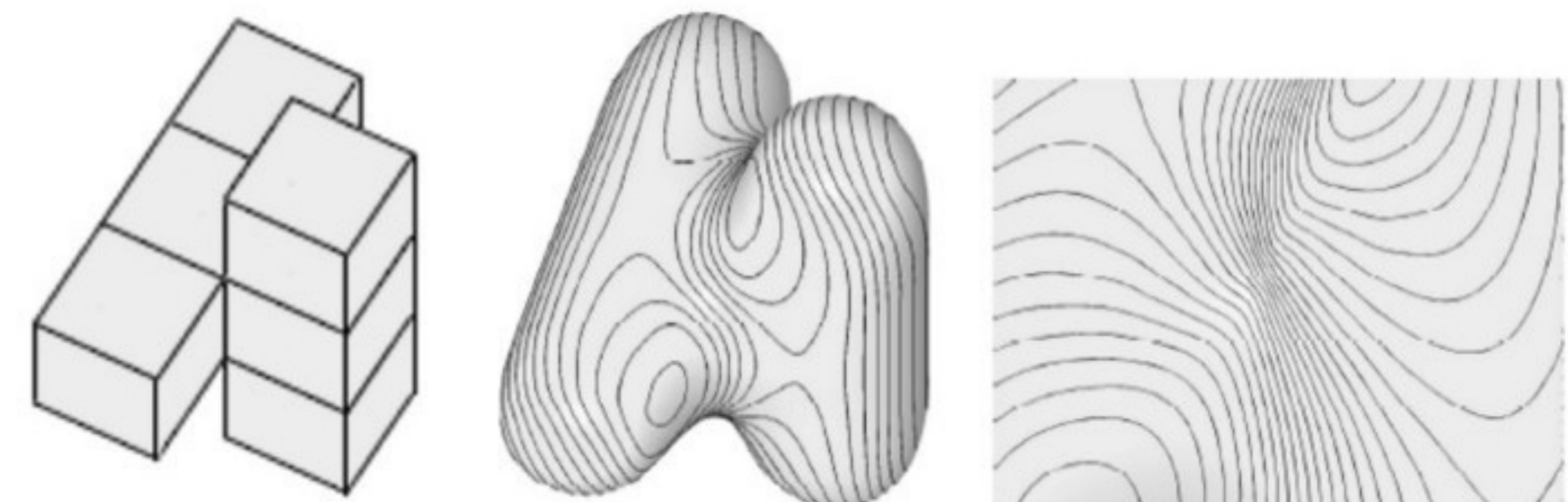
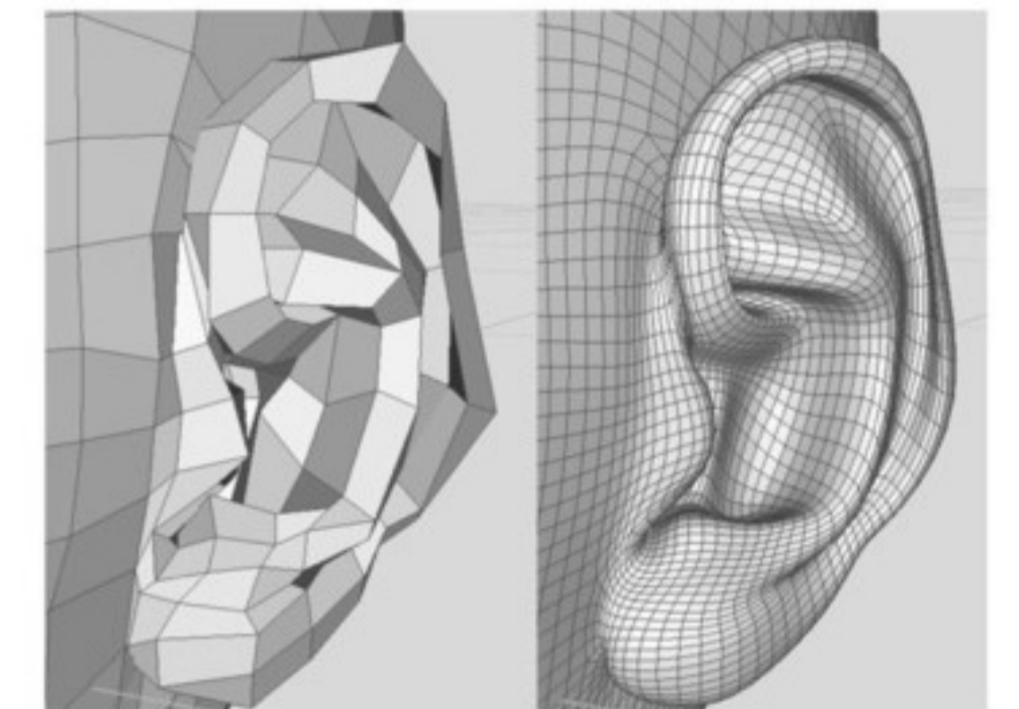
D. Zorin, P. Schroeder. Subdivision for Modeling and Animation.
ACM SIGGRAPH Course Notes, 1999.

022

Subdivision surface, conclusion

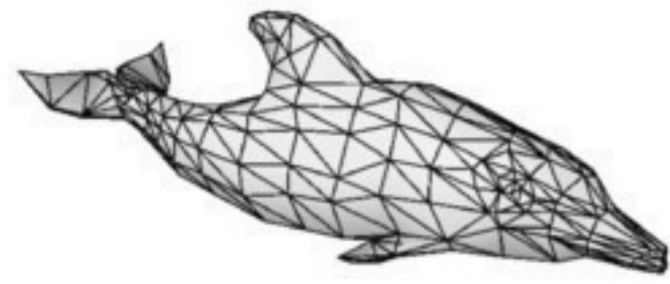
+ Arbitrary topology
+ Smooth surface

- Control of the final shape
- Extraordinary vertices



023

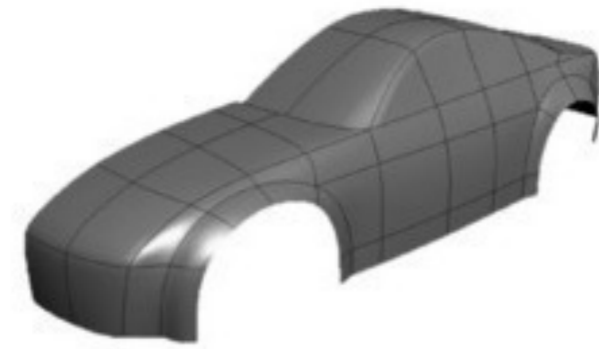
BRep: Comparison



Mesh

- + Simple
- + Generic
- + Automatic generation
- Non differentiable
- Bad approximation
- Manipulation

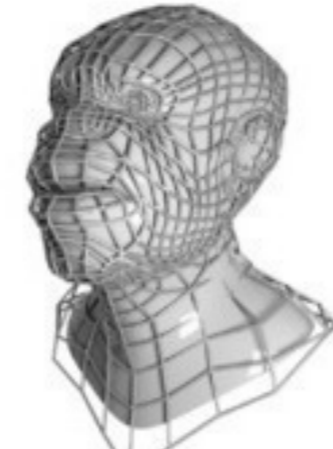
Graphics, Computation.
Maya, 3DStudio, Blender



Parametric

- + Smoothness
- + Parametrization information
- Technical (math. model)
- Patch structure
- Manual generation

CAD, (Graphics)
Rhino, Catia, Autodesk



Subdivision

- + Smooth appearance
- +/- No Patch, Extr. vertices
- No information on limit surface

Graphics, (CAD)

024

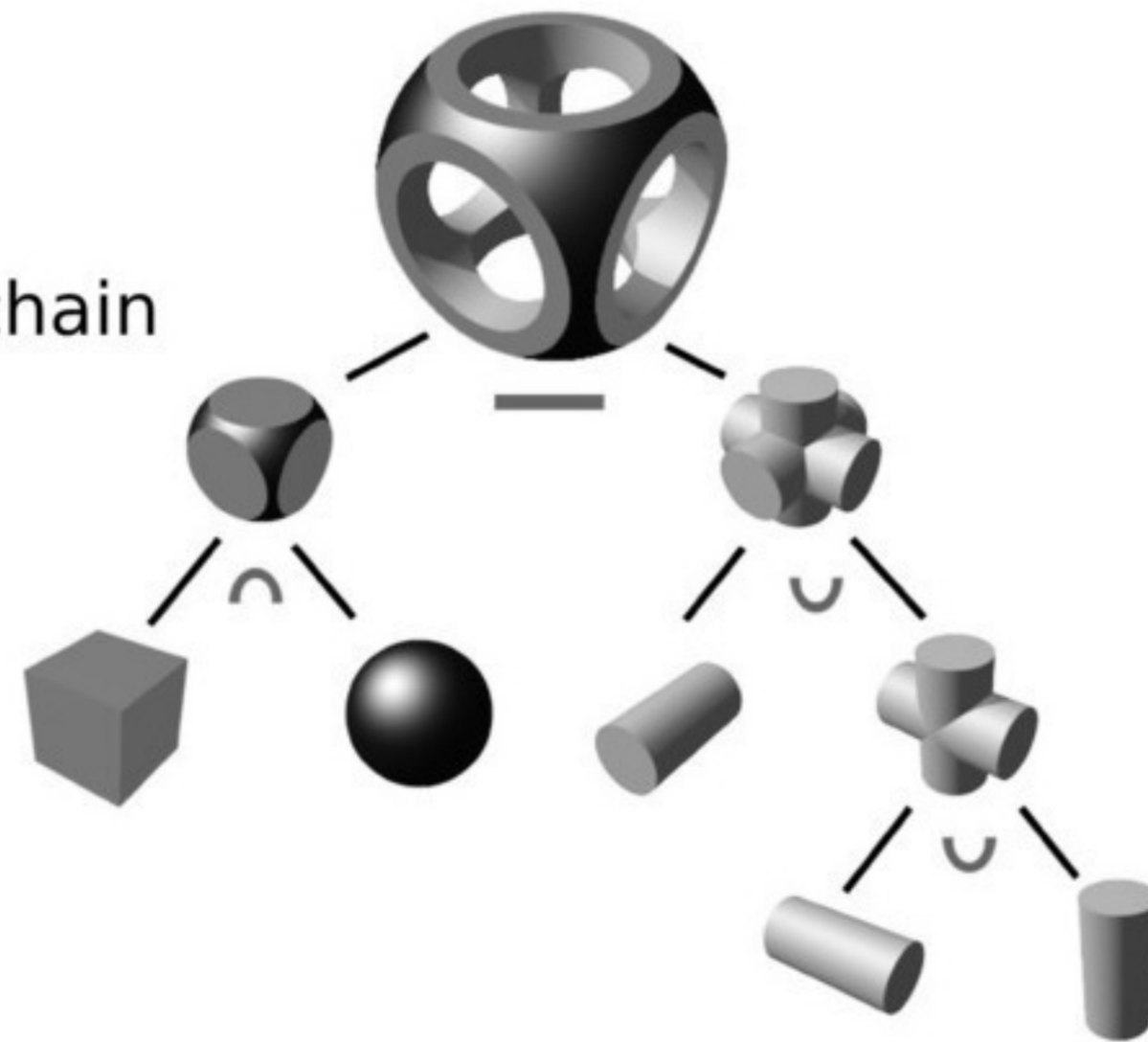
Explicit : CSG

025

CSG: Constructive Solid Geometry

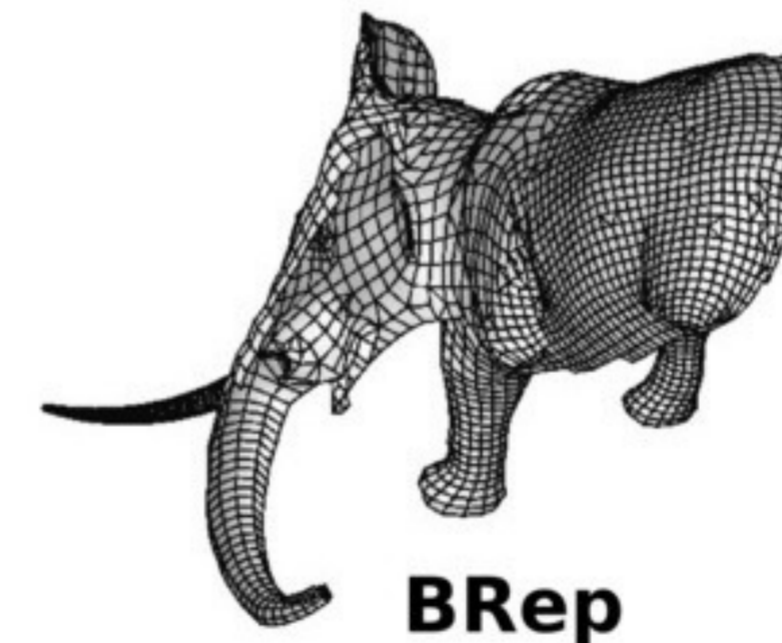
CSG = Assembling primitives using boolean operations

- Solid: Interior/Exterior
- Modeling an assembly chain
use in CAD :
Solid Works,
AutoCAD,
Catia,
PovRay.



026

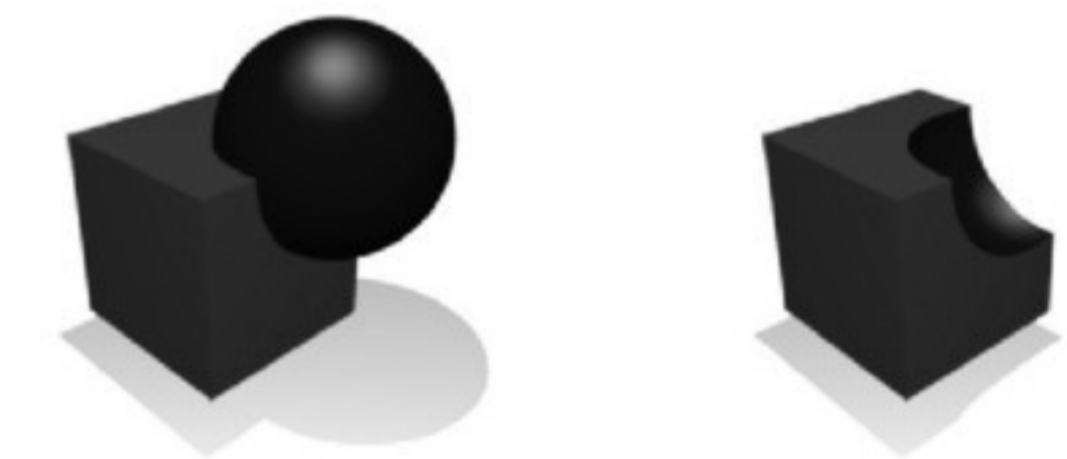
BRep VS CSG



BRep

- + Modeling complex object
- Approximation
- Surface only
- Discretization

Arbitrary discrete surfaces
(Graphics, Computation, CAD)



CSG

- + Exact
- + Constructive method
- Limited possibilities
- Tedious for complex shapes
- Non unique construction

Simple exact shapes
(CAO)

027

Implicit

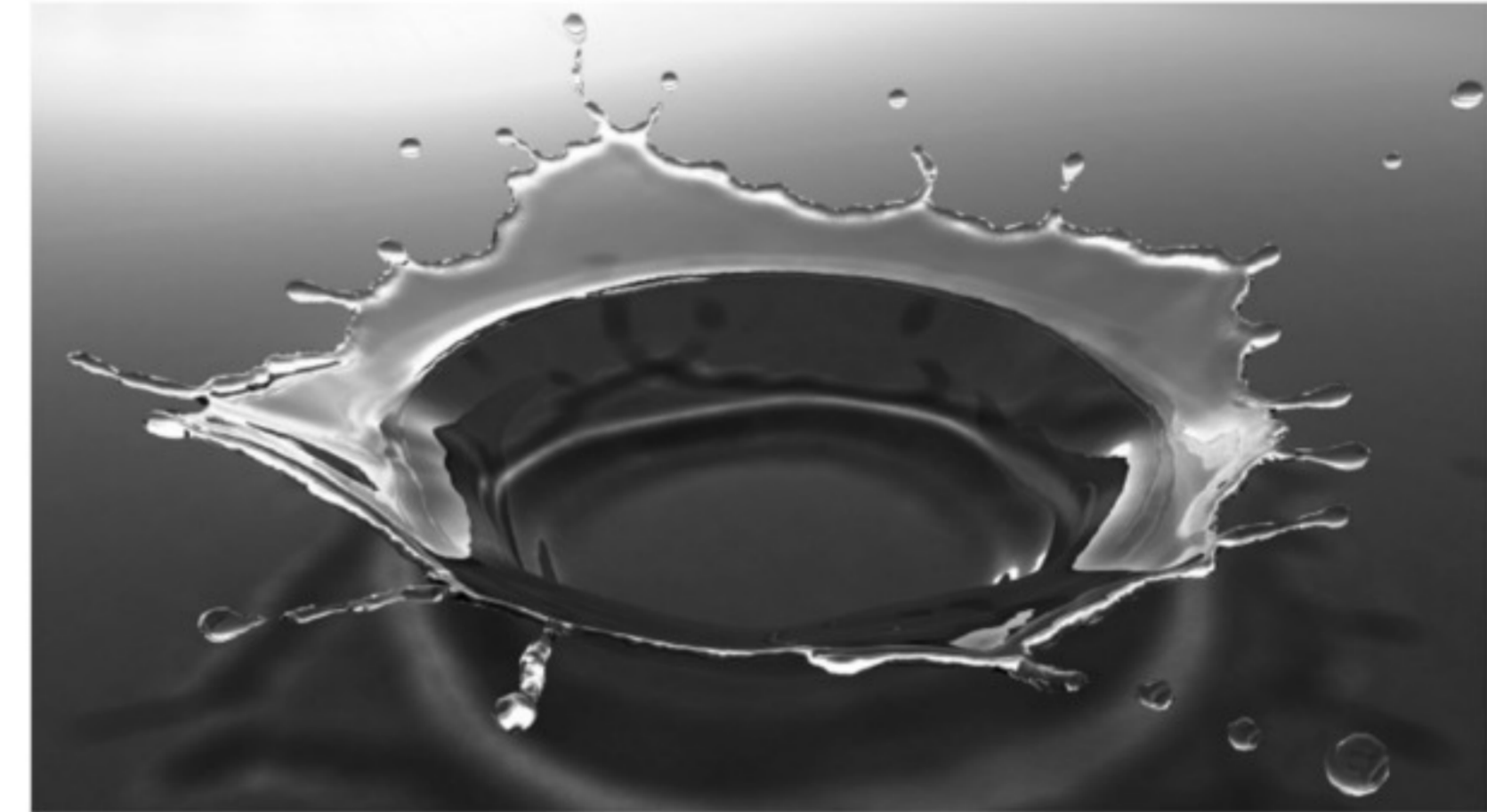
028

Implicit Modeling

Why using implicit modeling: **Topology changes**

$$S = \{\mathbf{p} \in \mathbb{R}^3 \mid \psi(\mathbf{p}) = 0\}$$

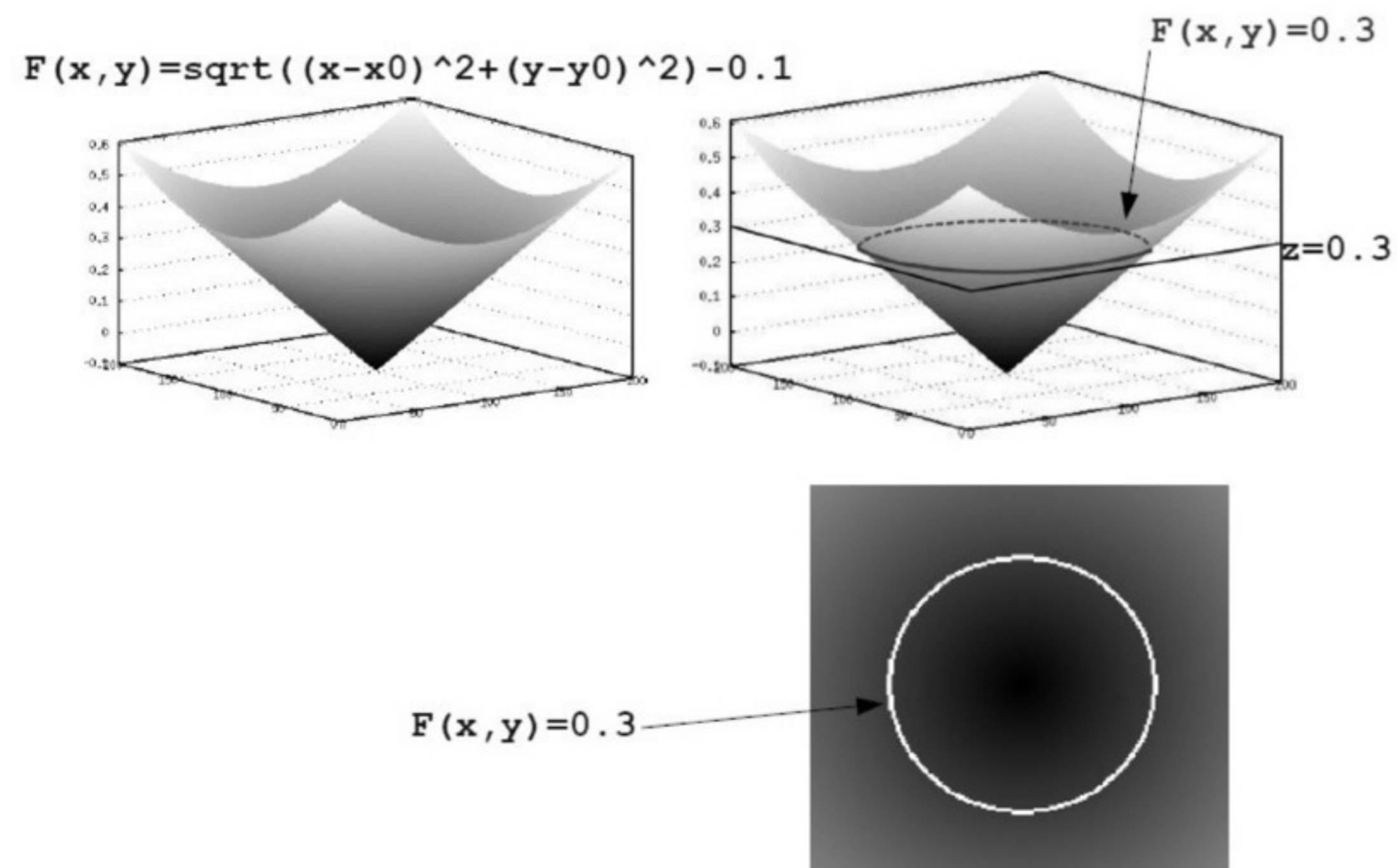
$$S = \psi^{-1}(0)$$



[Thuereu, Wojtan, Gross, Turk, SIGGRAPH 2010]

029

Implicit Modeling in 2D

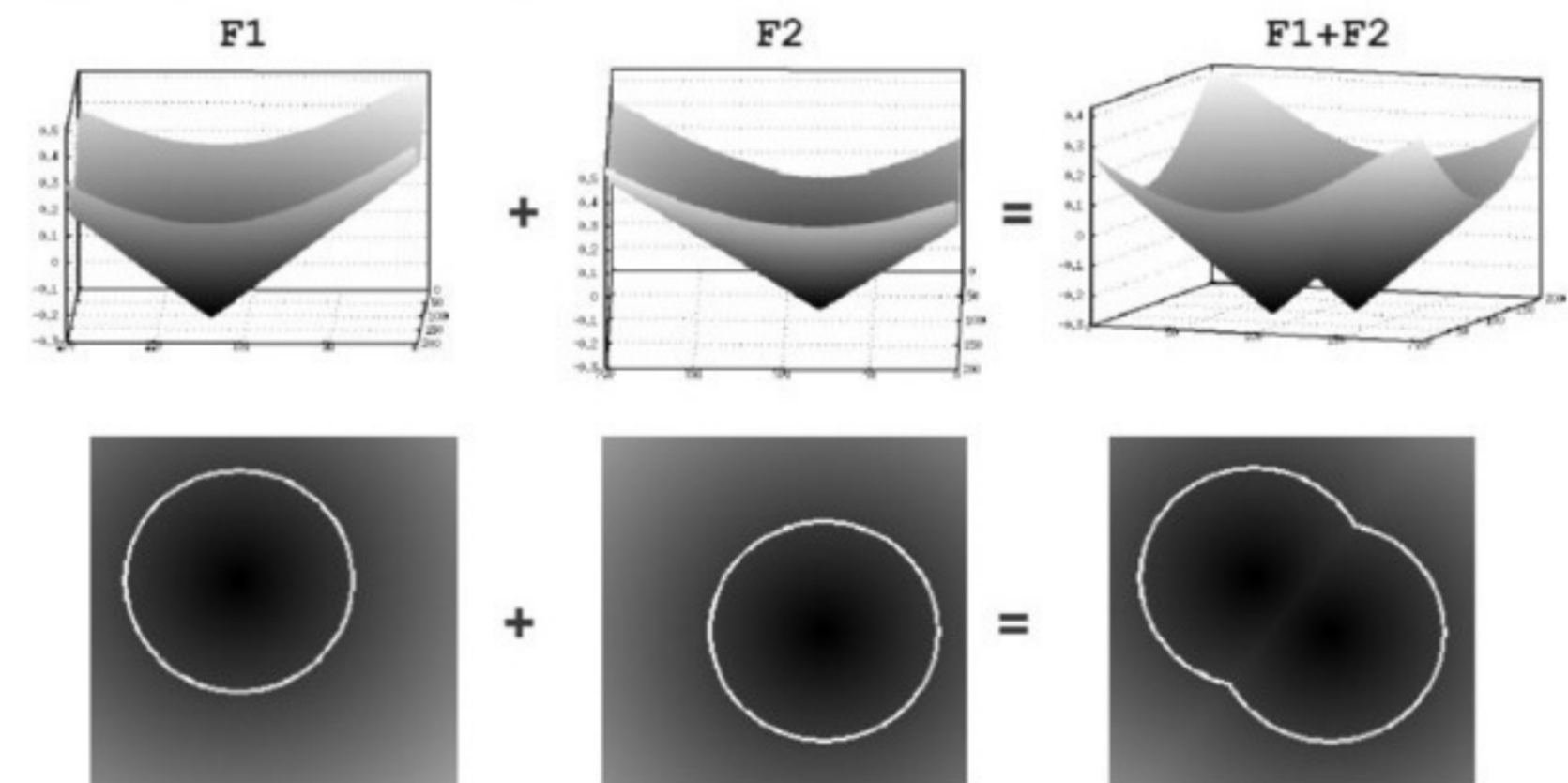


030

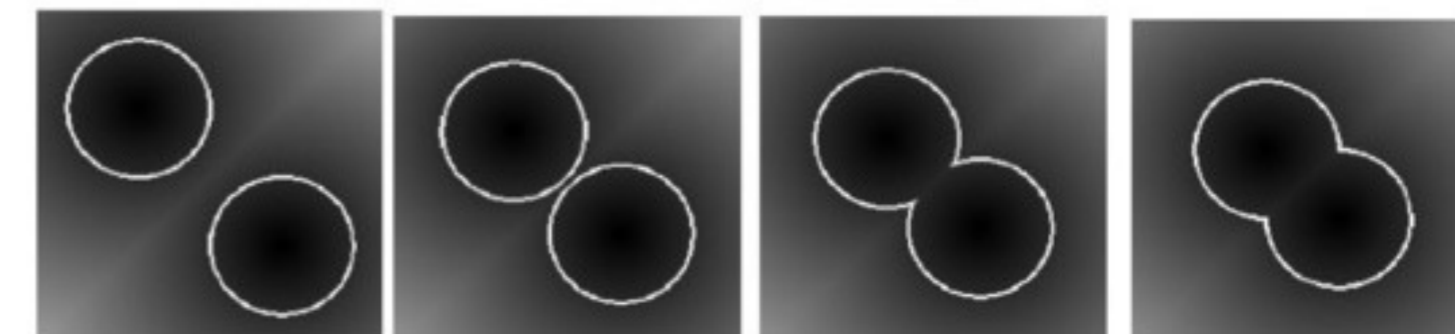
Implicit Modeling in 2D

$$F_1(x, y) = \sqrt{(x-x_1)^2 + (y-y_1)^2} - 0.1$$

$$F_2(x, y) = \sqrt{(x-x_2)^2 + (y-y_2)^2} - 0.1$$



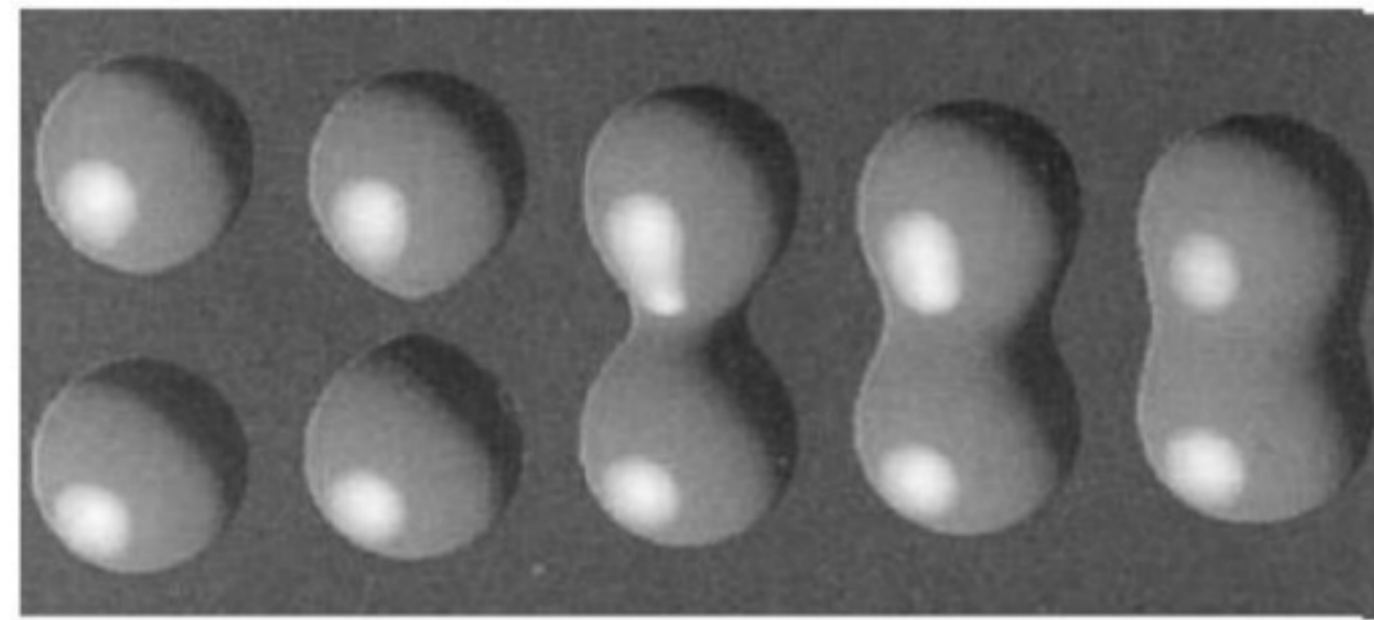
position of center: different topology



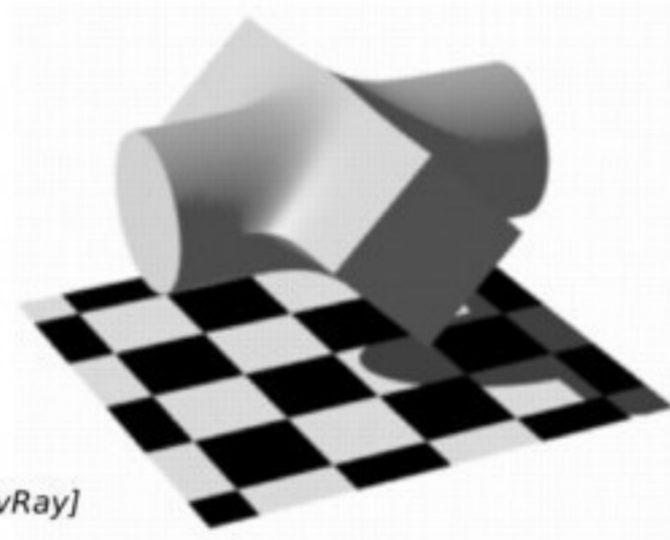
031

Implicit Modeling in 3D

$$\psi(x, y, z) = 0$$



[Wyvill, McPheeters, Wyvill, The Visual Computer, 1986]

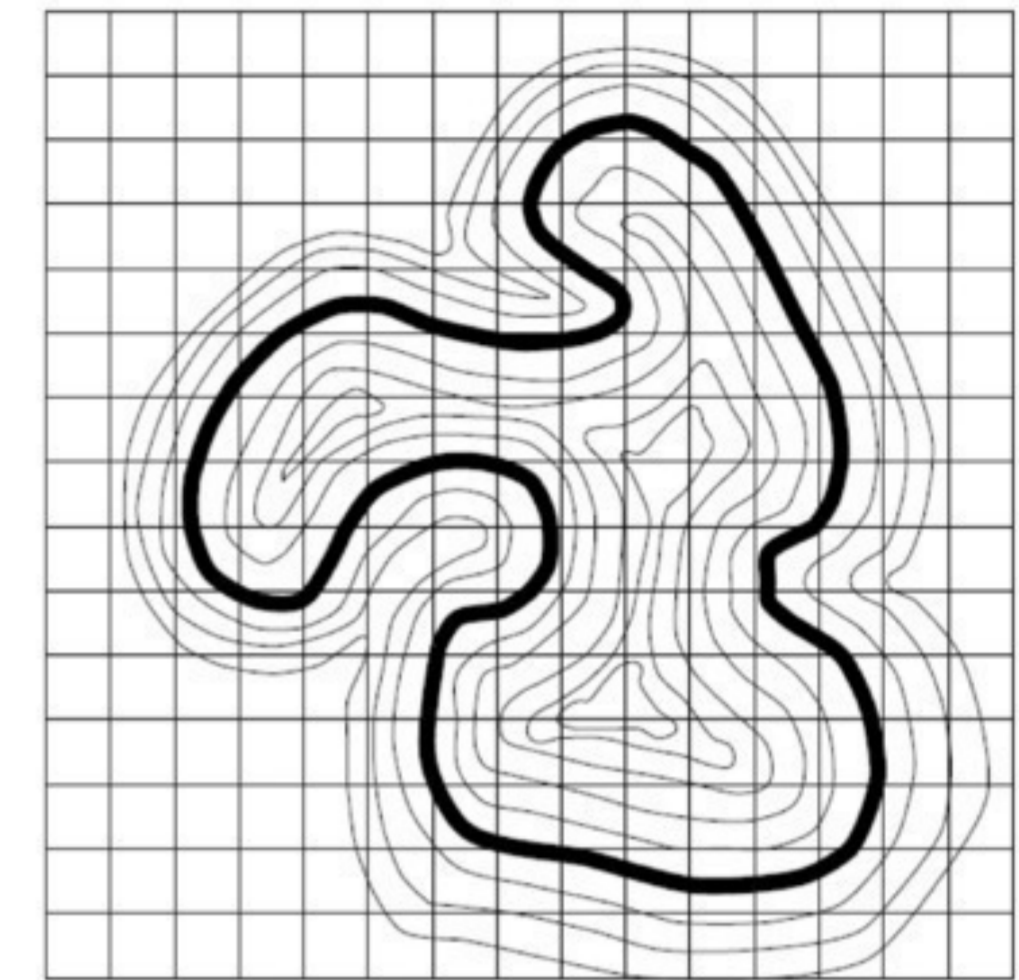


[PovRay]

032

Implicit Function Encoding

How do we encode the field function ?



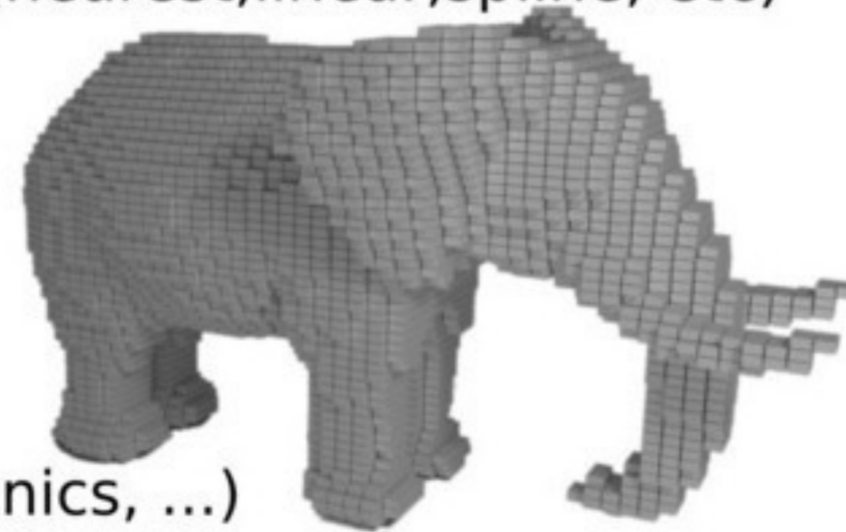
[Sethian]

033

Voxel Grid

- Spatial discretization in voxels
- Store each value in the grid $\psi(k_x, k_y, k_z)$
- Access value using interpolation (nearest, linear, spline, etc)

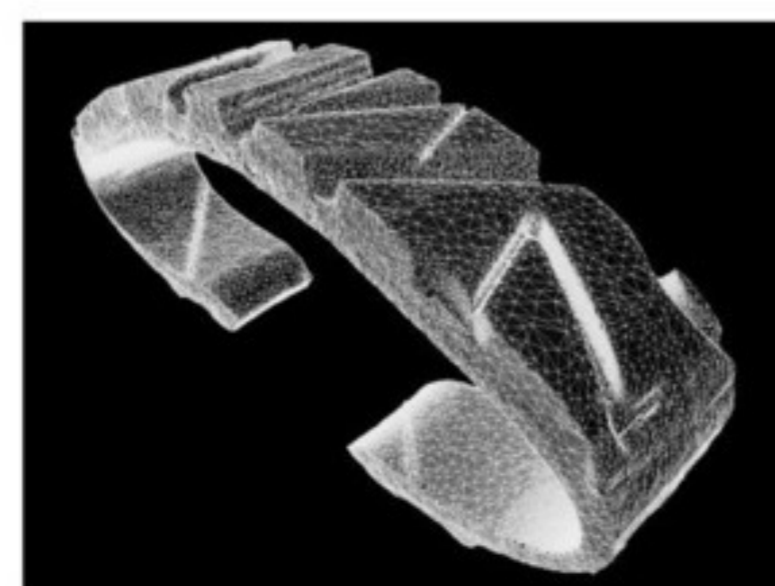
- + Very general
- Memory storage
(1024^3 voxels => 8Go RAM)



Scanner imaging (medical, mechanics, ...)



[Matlab]



[Digisens]

034

Parametric

Skeleton based

- Blobs

$$\psi_i(\mathbf{p}) = e^{-\frac{\|\mathbf{p} - \mathbf{p}_i\|^2}{\sigma^2}}$$

- Metaballs

$$\psi_i(\mathbf{p}) = \sum_k \omega_k \|\mathbf{p} - \mathbf{p}_i\|^k$$

- Convolution

$$\psi_i(\mathbf{p}) = \int_{\mathbf{q} \in S_i} \omega(\mathbf{q}) h(\|\mathbf{p} - \mathbf{q}\|) d\mathbf{q}$$



[Sherstyuk, 98-99]

Final field:

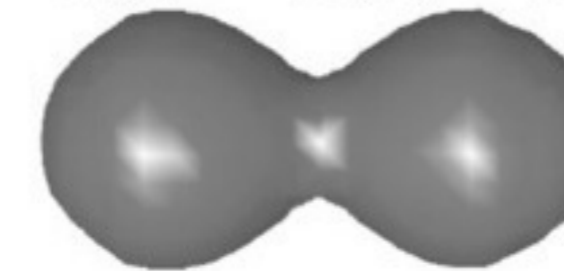
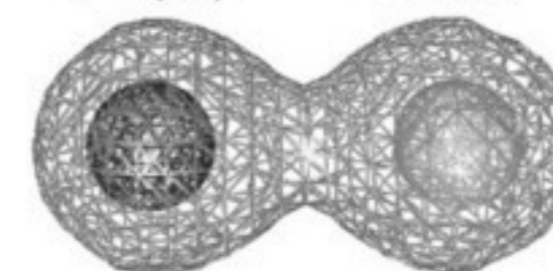
Sum of the contribution

$$\psi(\mathbf{p}) = \sum_i \psi_i(\mathbf{p})$$

$$\psi_1(\mathbf{p})$$

$$\psi_2(\mathbf{p})$$

$$\psi(\mathbf{p}) = \psi_1(\mathbf{p}) + \psi_2(\mathbf{p})$$

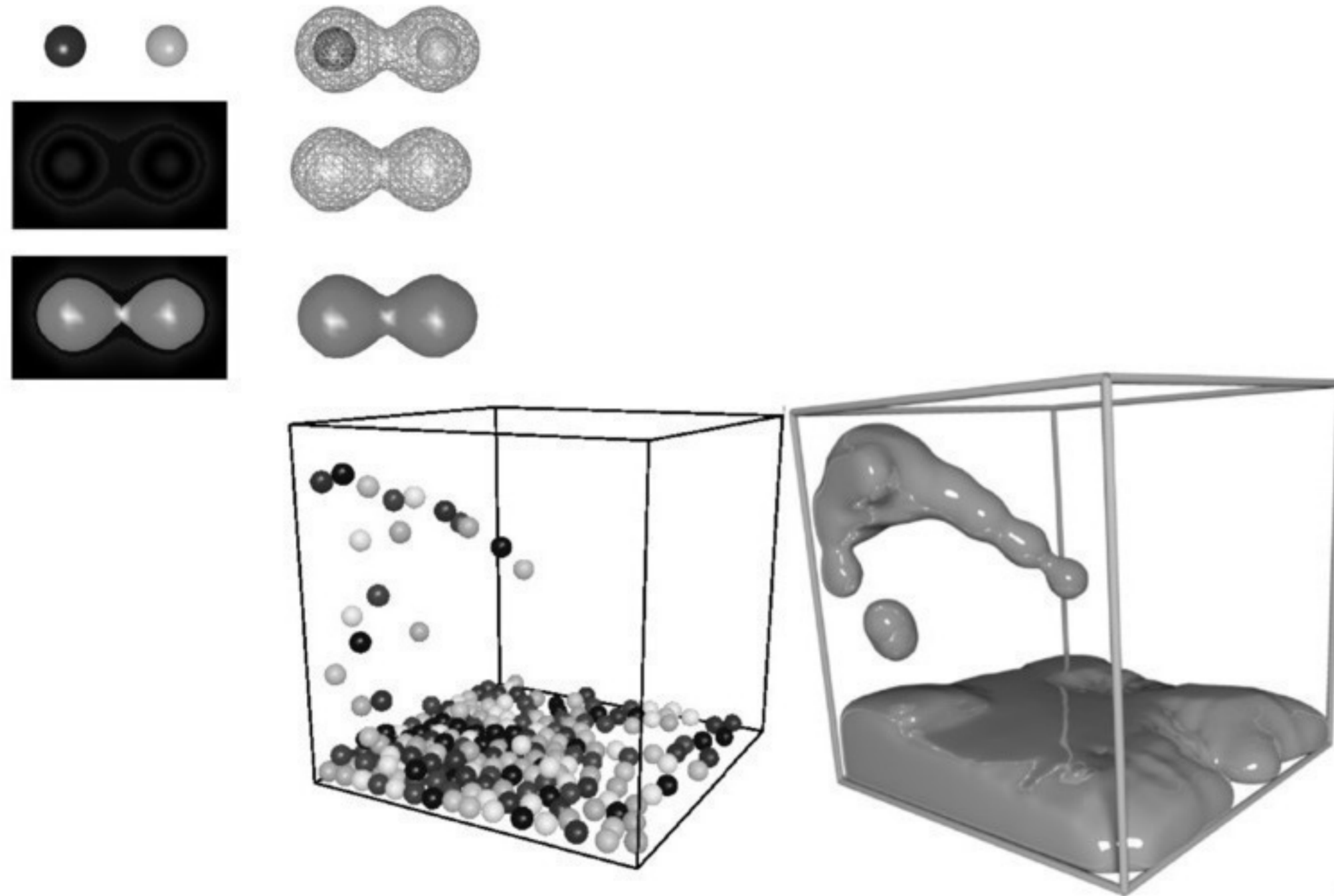


Direct manual control.
=> use for Graphics

035

Parametric

Skeleton based: Application to fluid simulation



036

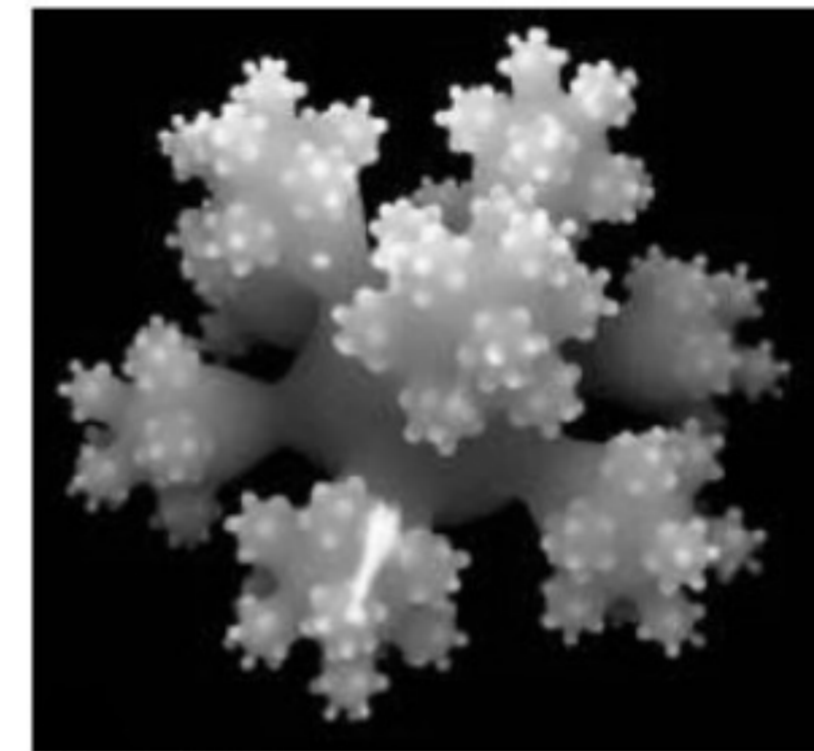
Parametric

Function optimization/fitting

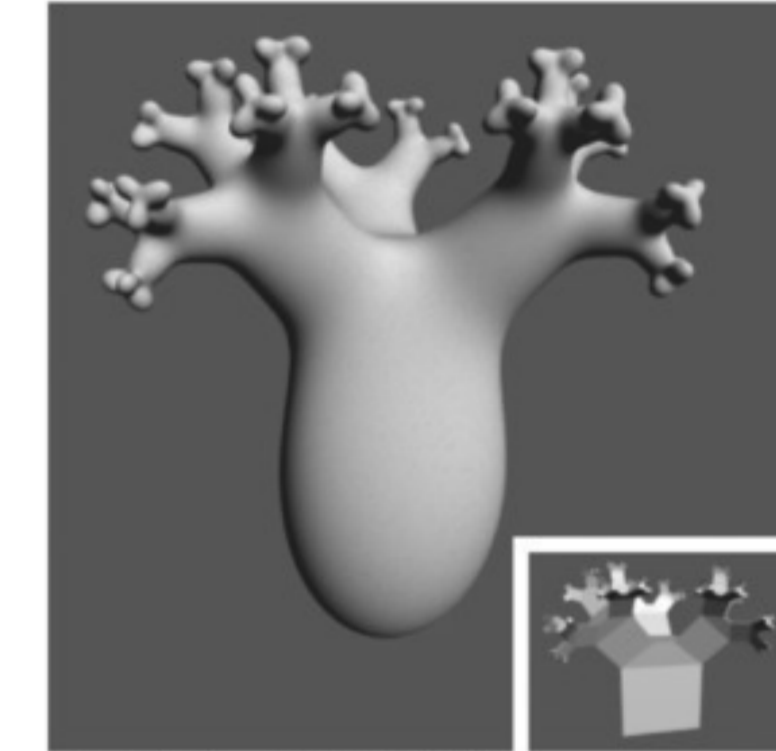
- Splines
- RBF

+ More general than skeleton based
- Requires a minimization, no direct control

=> mostly medical applications



[Sherstyuk, 98]

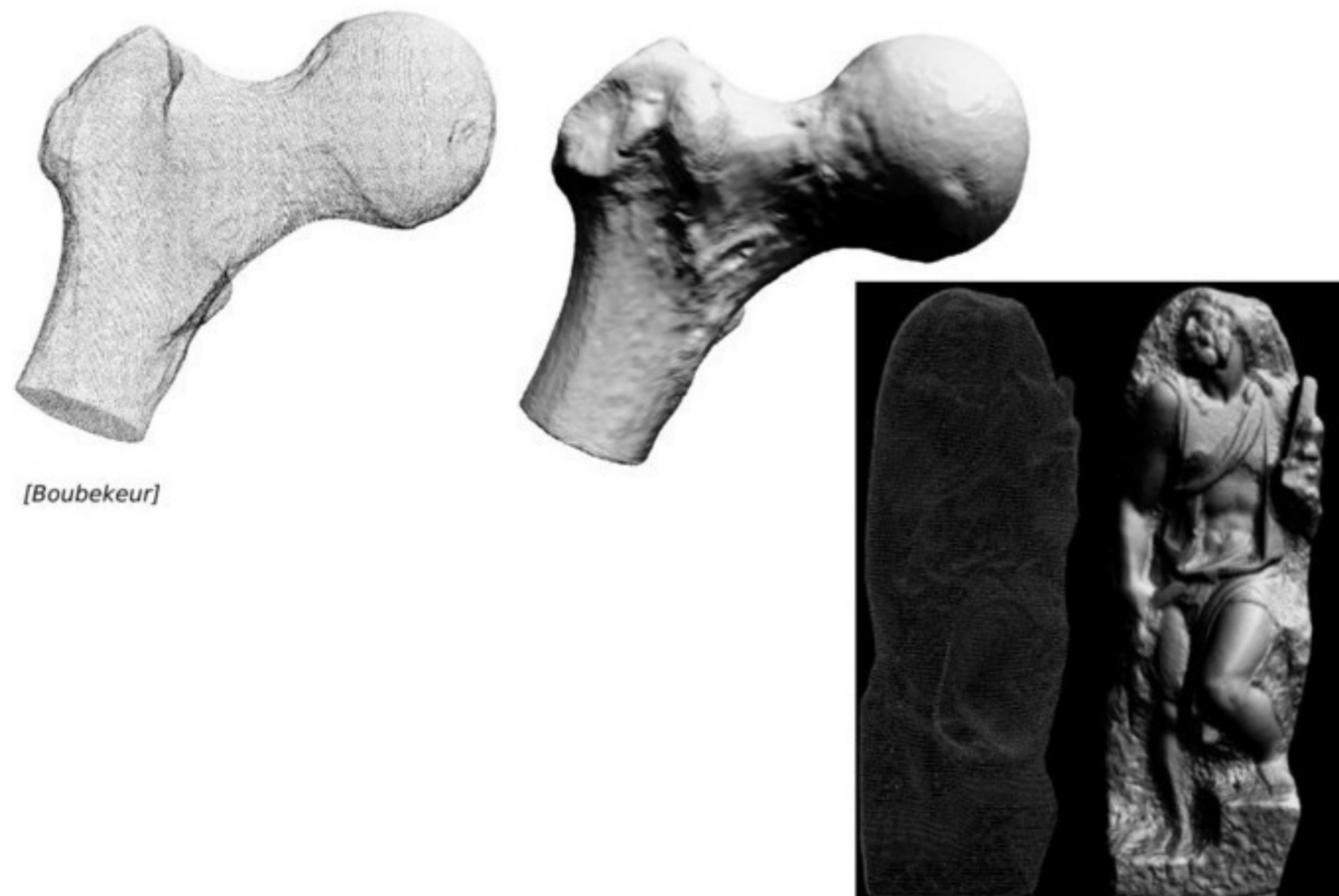


[Turk, O'Brien, SIGGRAPH 02]

037

Points set

Input = discrete points (+ normals)
Data from scanners



[Boubekeur]

038

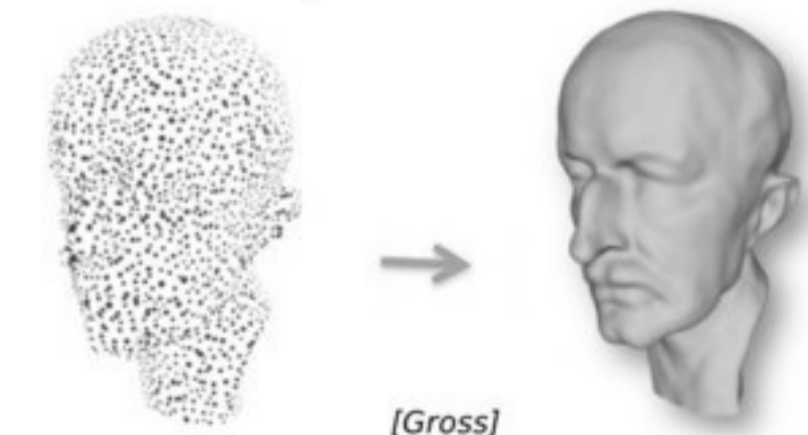
Points set

Moving Least Squares (MLS)

Goal: Find f , smooth function such that

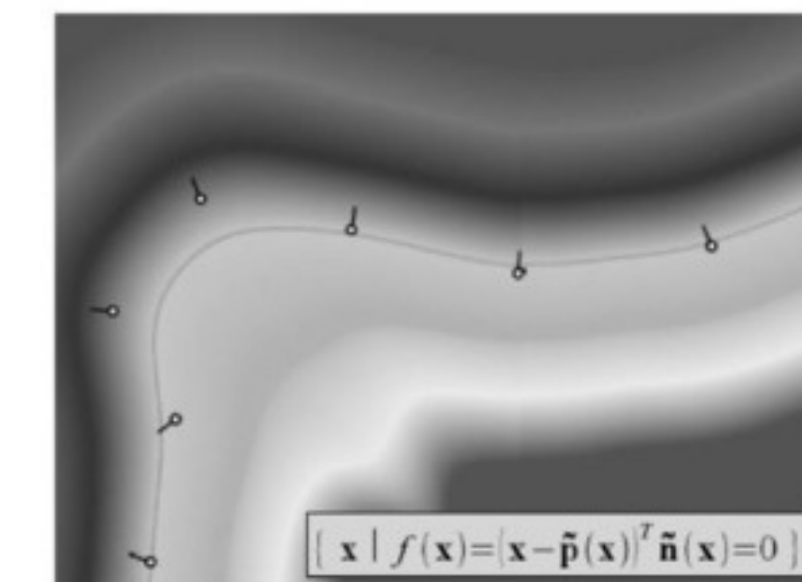
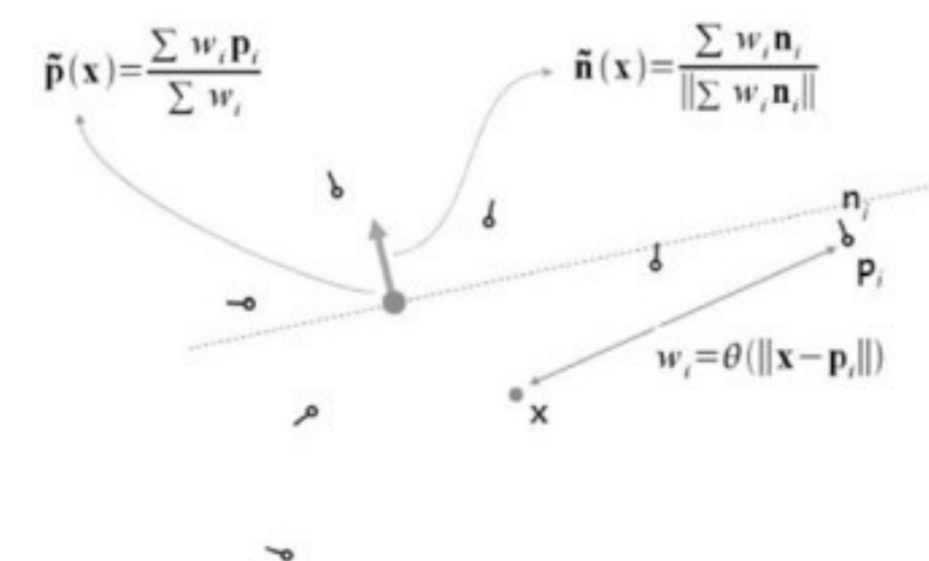
$$f = \operatorname{argmin} \left(\sum_i \psi(\|\mathbf{p} - \mathbf{p}_i\|) (f(\mathbf{p}) - f(\mathbf{p}_i))^2 \right)$$

+ Smooth approximating functions
- Minimization



[Gross]

Application to noisy data



$$[\mathbf{x} \mid f(\mathbf{x}) = \mathbf{x} - \tilde{\mathbf{p}}(\mathbf{x}) \mid \tilde{\mathbf{n}}(\mathbf{x}) = 0]$$

039

Points set

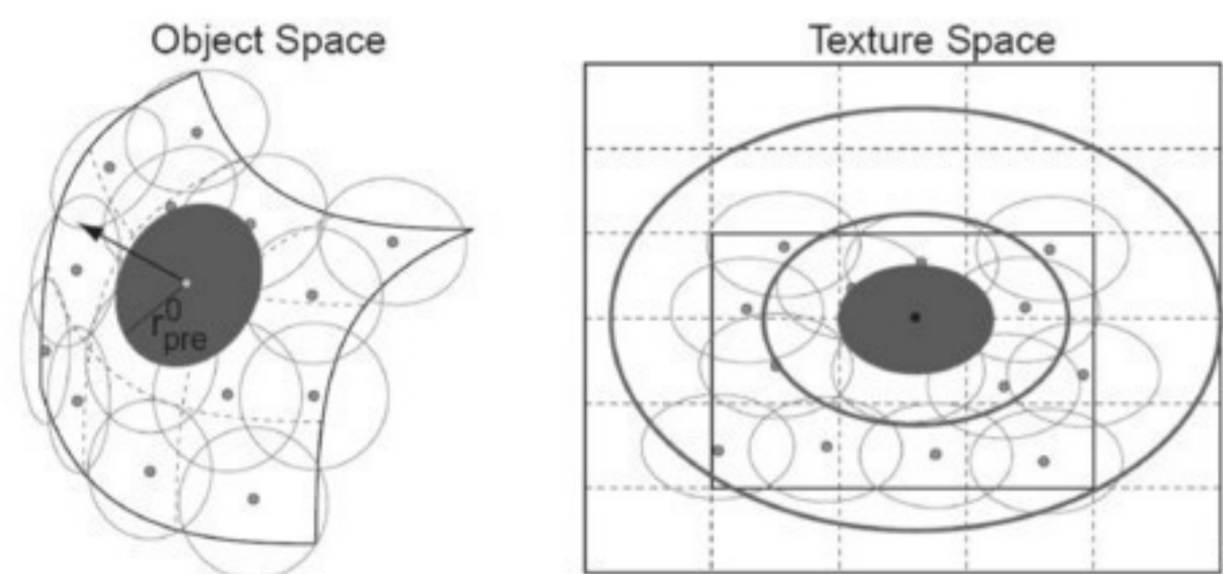
Surfels

Goal: Render a continuous surface from simple primitives

$$f = \operatorname{argmin} \left(\sum_i \psi(\|\mathbf{p} - \mathbf{p}_i\|) (f(\mathbf{p}) - f(\mathbf{p}_i))^2 \right)$$

- + Fast rendering
- No underlying surface

Application to big data



[Pfister, Zwicker, van Baar, Gross, SIGGRAPH 00]



[Zwicker]

040

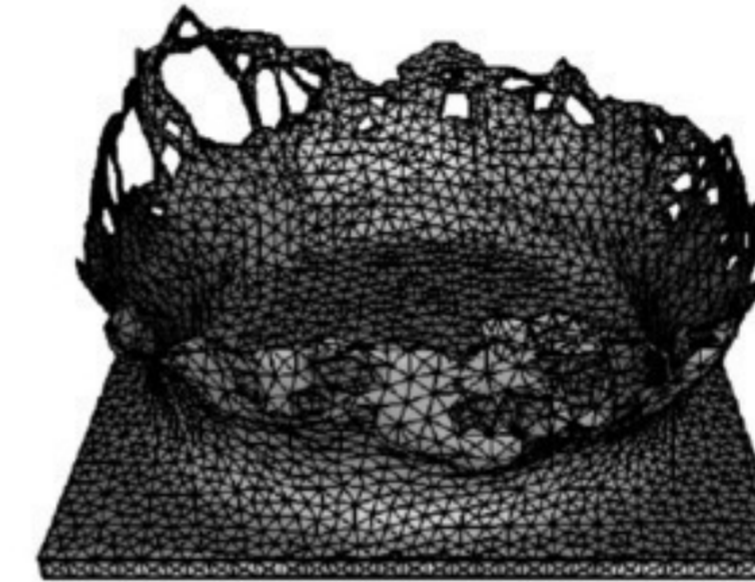
Implicit VS Explicit

- + Arbitrary topology
- + Shape blending

- Manipulation
- Memory cost
- Rendering time
- Surface details



[Hornus, Angelidis, Cani, Vis. Comp. 03]



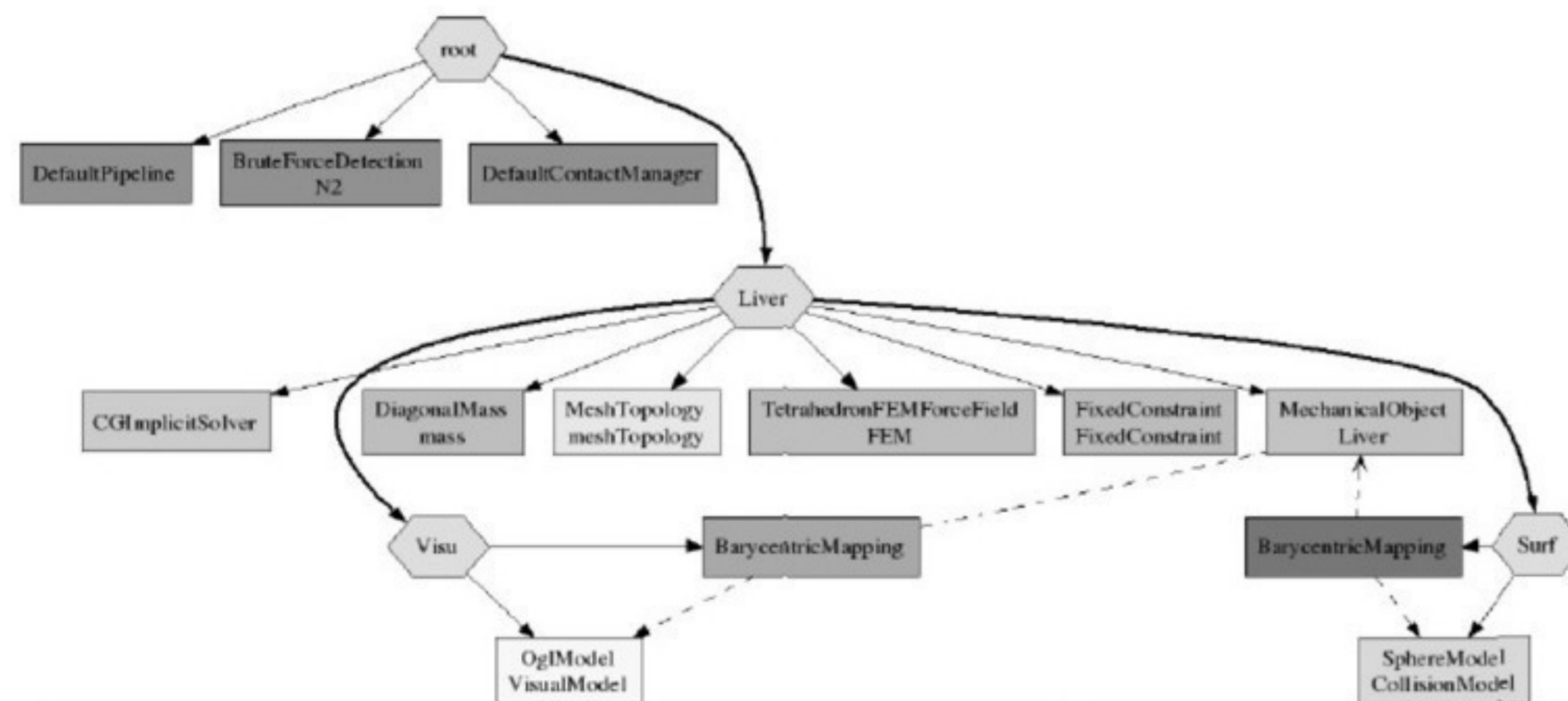
[Broshu, Batty, Bridson, SCA 09]



[Ohtake, Belyaev, Alexa, Turk, Seidel, SIGGRAPH 03]

041

Interaction between representations



[SOFA, Inria]

042

Fractals

043

Fractals

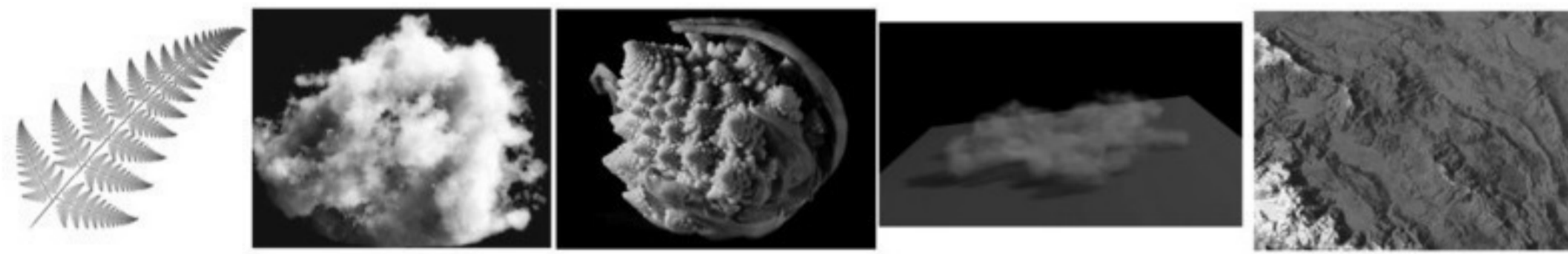
Principles:

Recursive deformations converging toward a complex shape

Use:

Modeling complex objects from a few simple rules

Application: Graphics (procedural modeling)



044

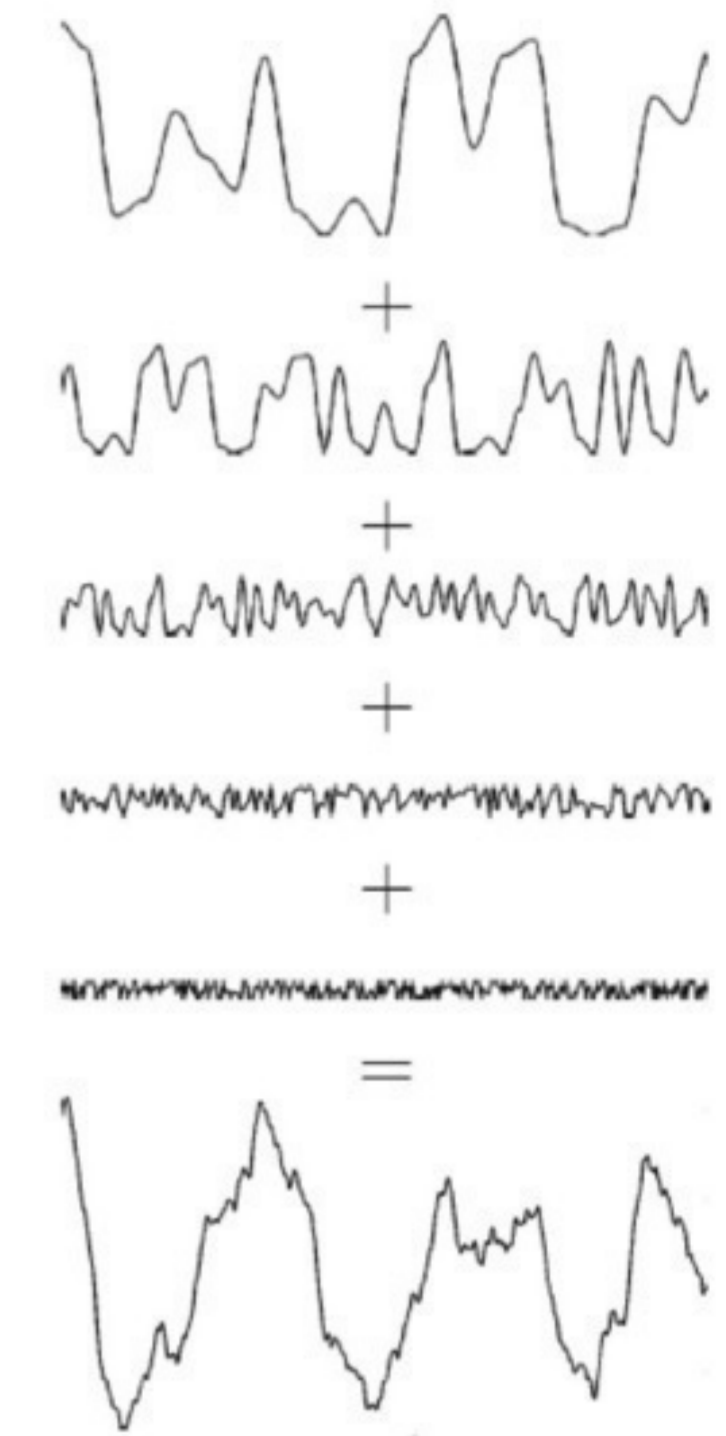
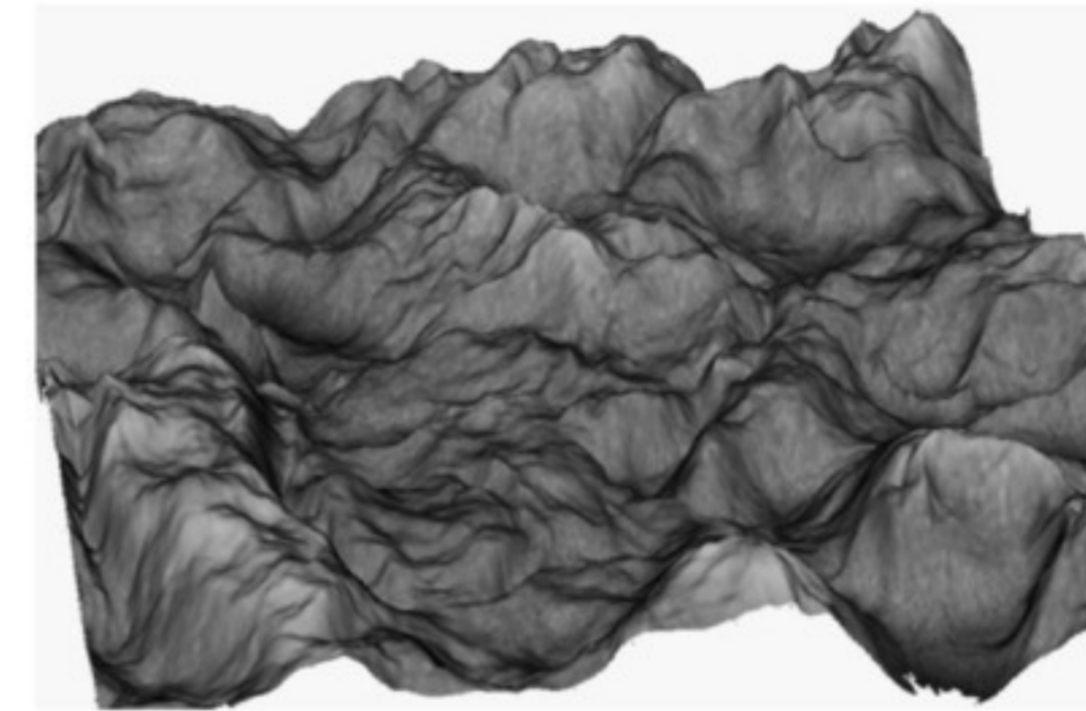
Perlin noise

$$f(x) = \sum_{k=0}^N \alpha^k f(\omega^k x)$$

N : Octave

α : Attenuation ($1/\alpha$: persistence)

ω : Frequency gain

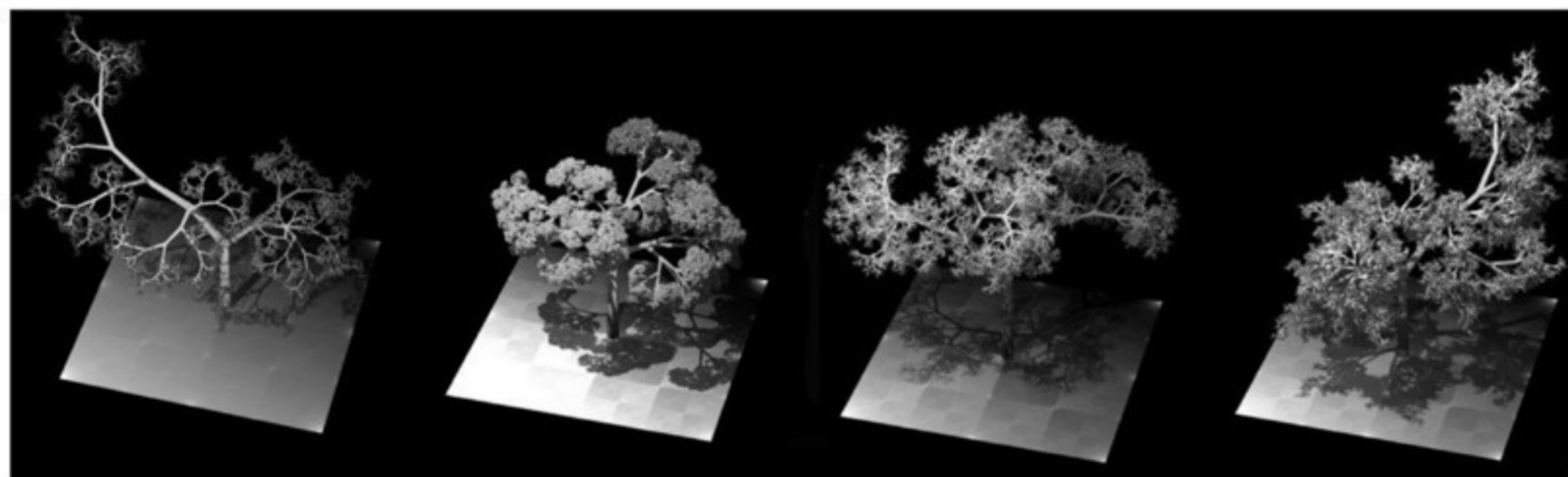
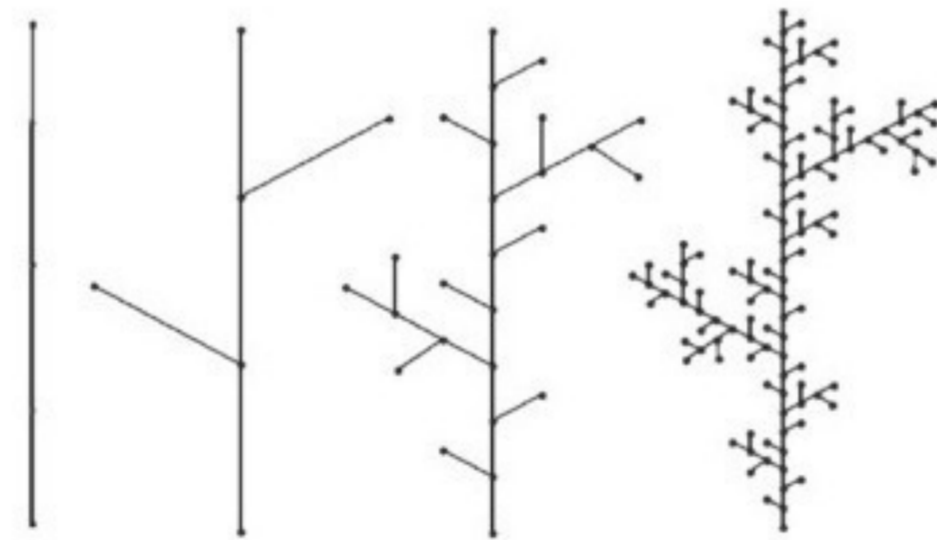


045

L-System

Grammar:

$F [+F] F [-F] F, \theta = 60^\circ$

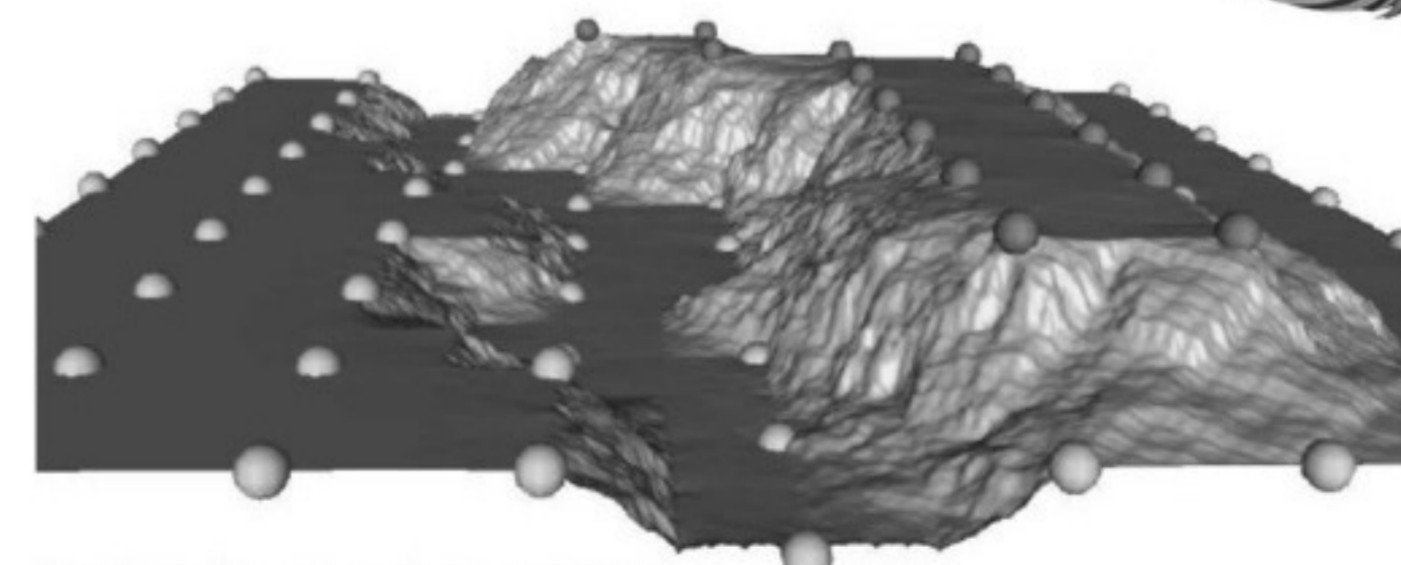
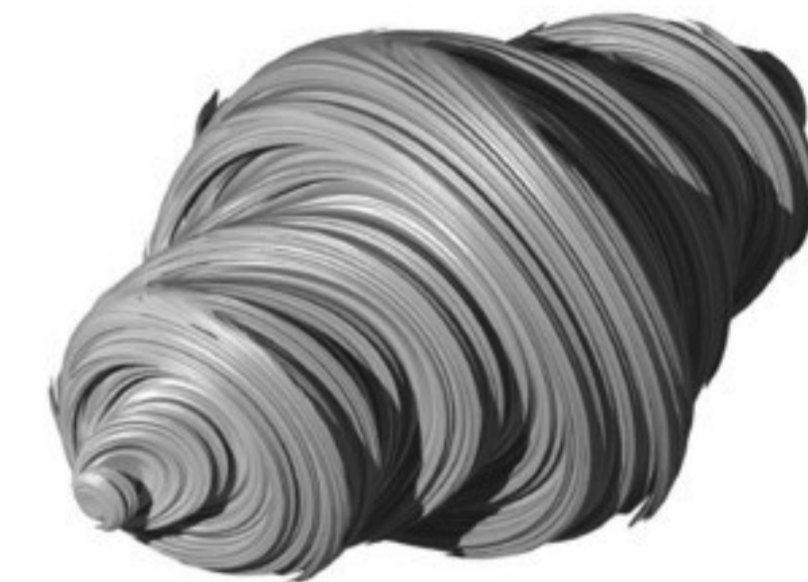


046

Fractals, conclusion

+ Complex objects from simple rules
+ Natural aspect

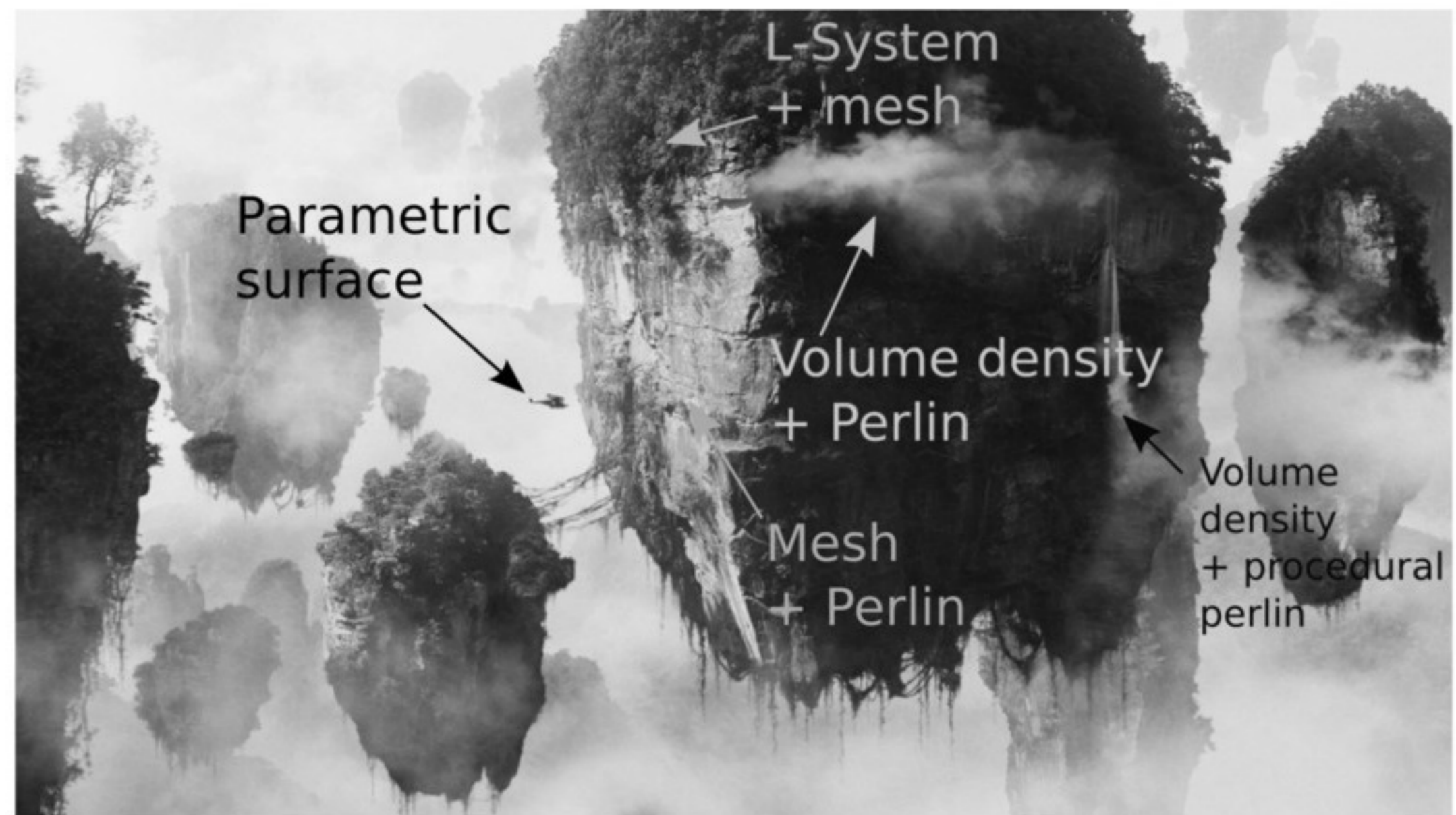
- Control



[Hnaidi, Guérin, Akkouche, Fractals 2010]

047

Various modeling



[Avatar]