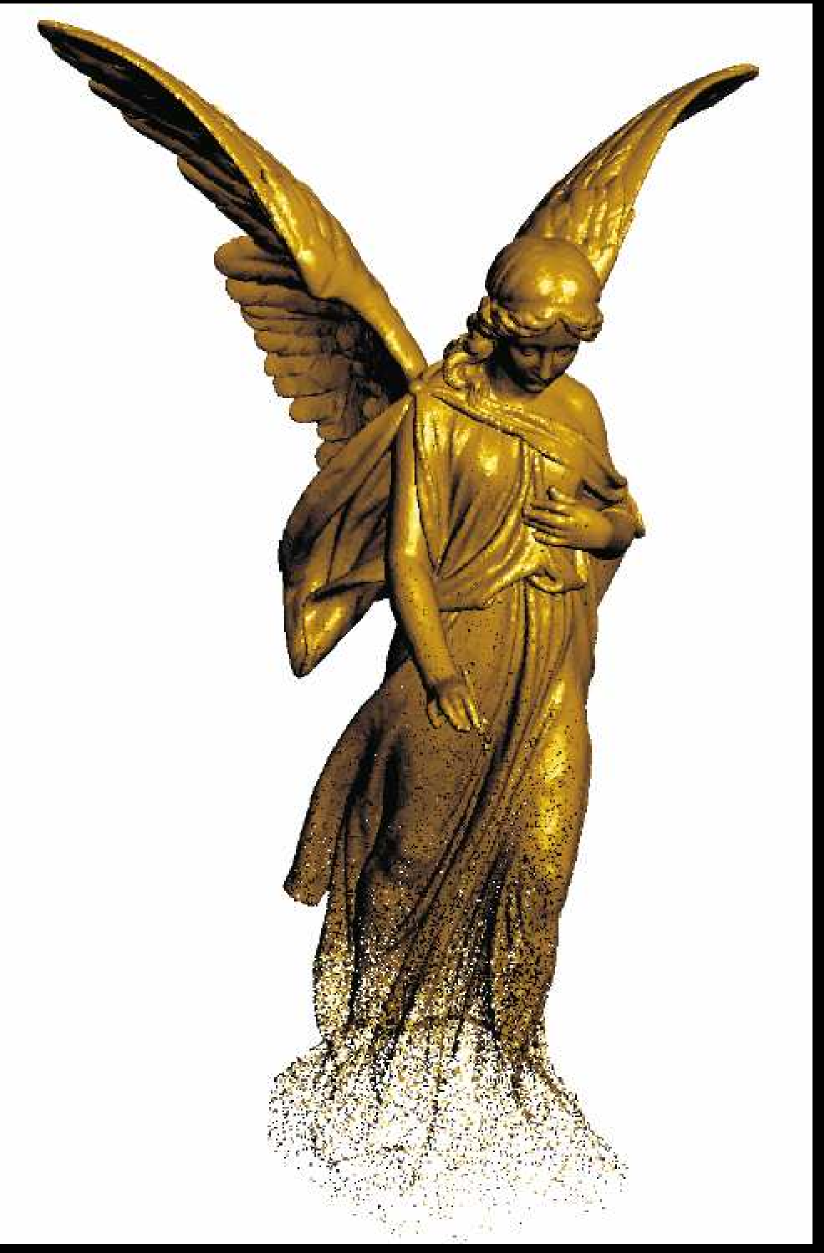
Point-based Modeling



Alexa et al., 2001

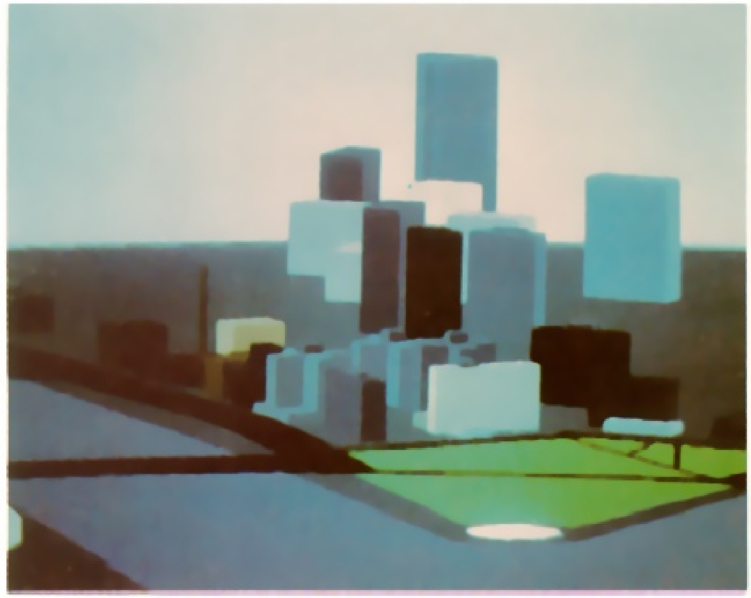


Figure 8: City of Pittsburgh from a distance.

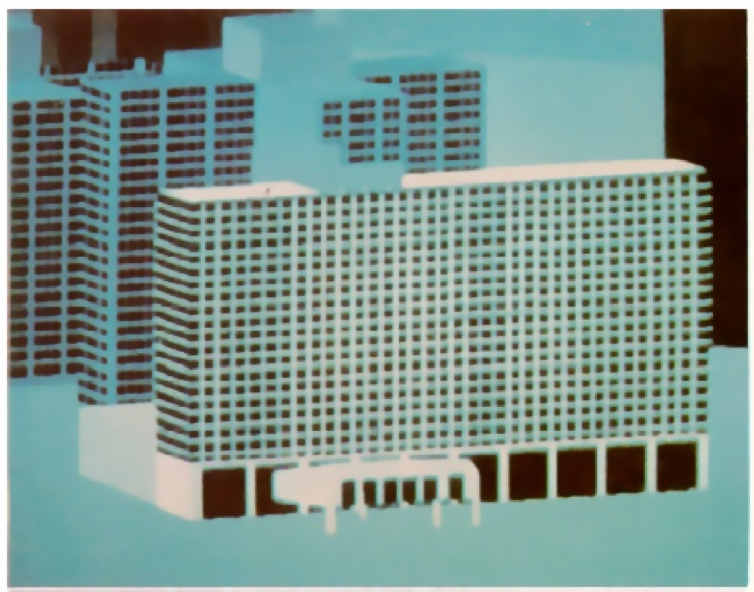


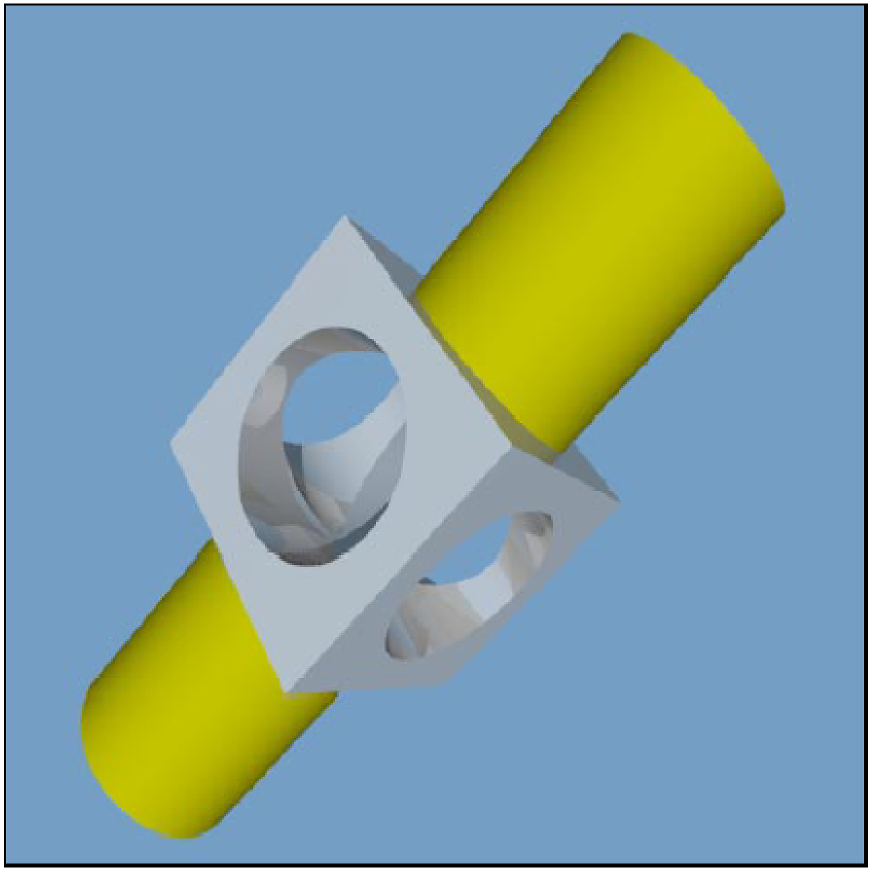
Figure 9: City of Pittsburgh close up.

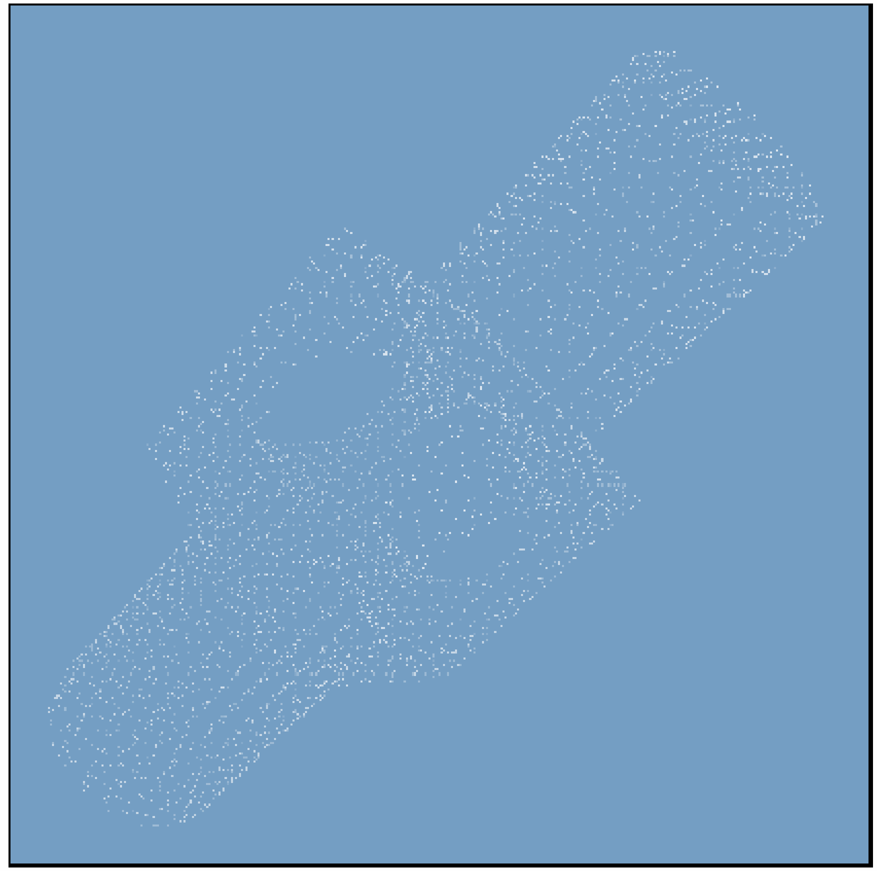
Rubin & Whitted, SIGGRAPH 1980

Navigating Point Clouds

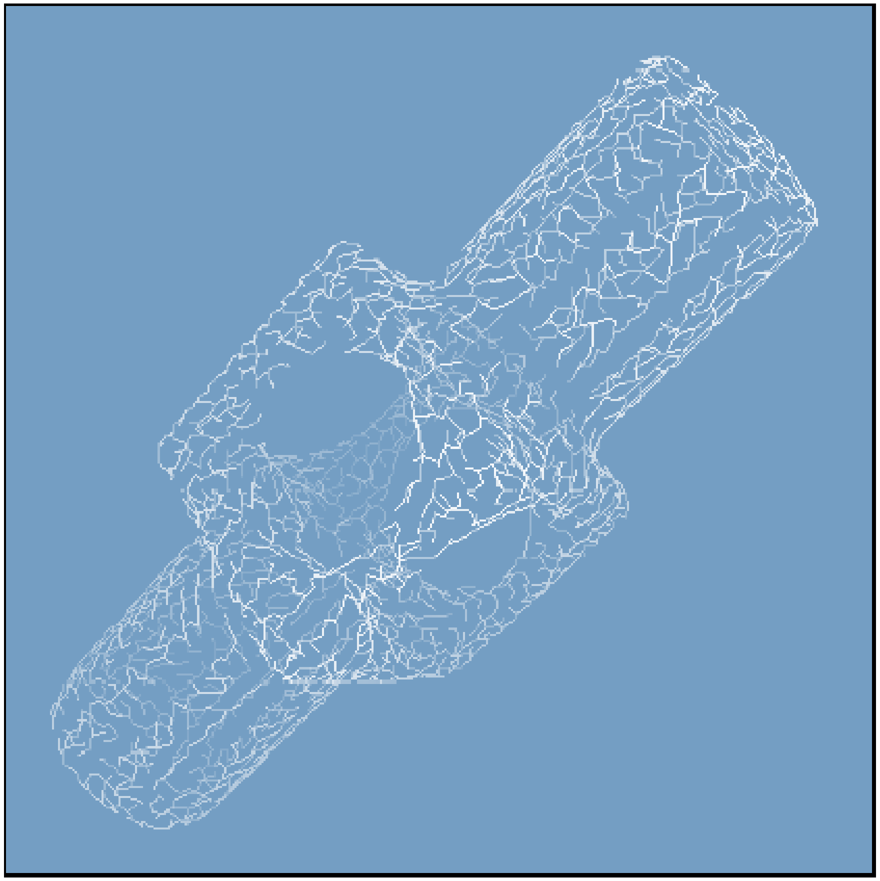
- Consistent normals?
- Inside and outside?
- Signed distance?

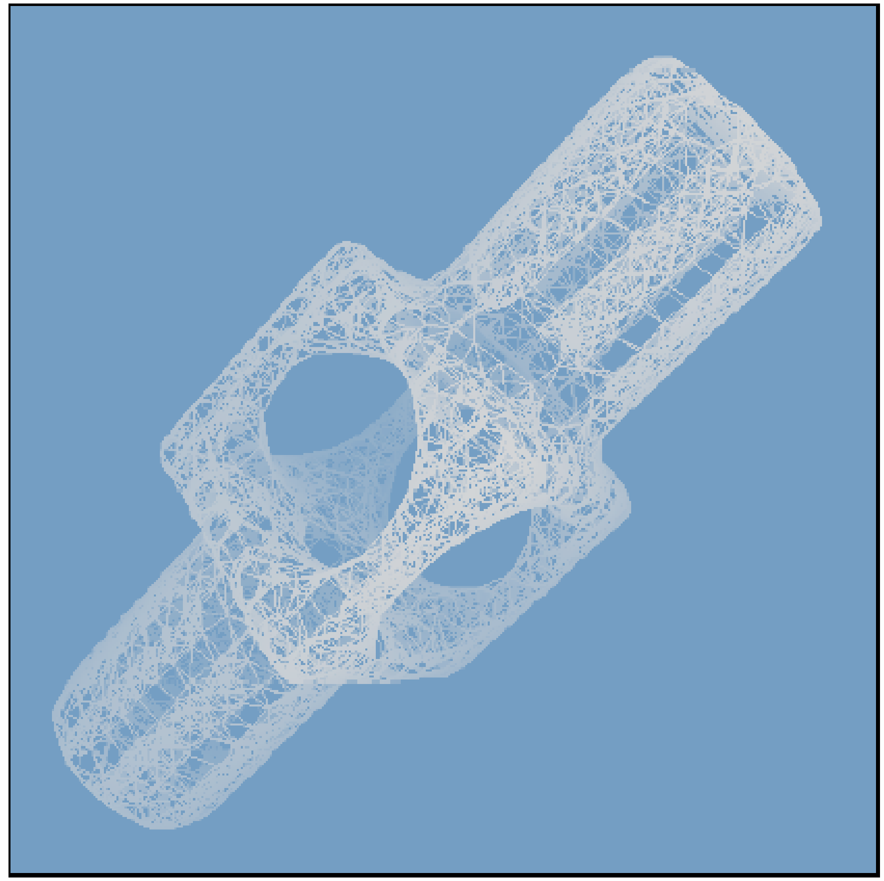
From [Hoppe et al., 1992]


(a) Original CSG object

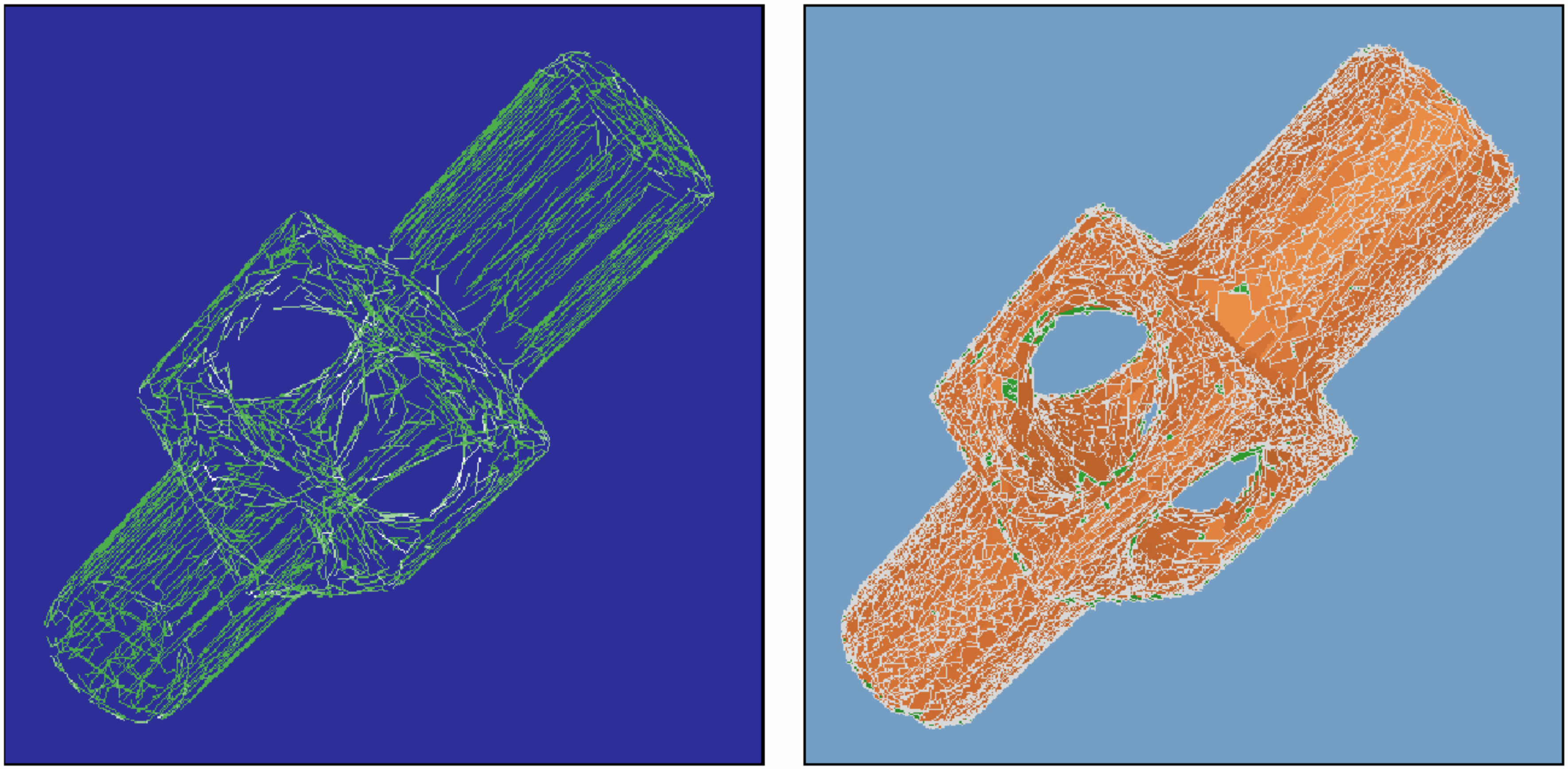

(b) Sampled points ($\mathbf{x}_i$) (n = 4102)

From [Hoppe et al., 1992]


(c) EMST of tangent plane centers $\mathbf{o}_i$


(d) Riemannian Graph over $\mathbf{o}_i$

From [Hoppe et al., 1992]



(a) Traversal order of orientation propagation

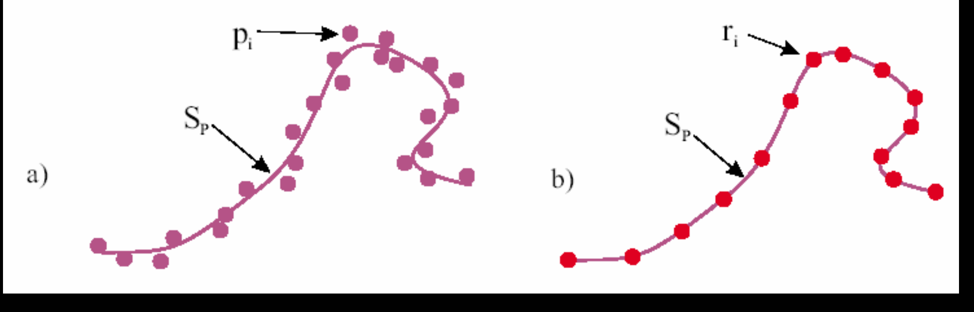
(b) Oriented tangent planes ($Tp(\mathbf{x}_i)$)

Rendering Point-based Models

- Surfels, or surface elements
- Splatting
  – one surfel maps to many pixels
- Two demos:
  – QSplat [Rusinkiewicz et al, 2000]
  – Surfels… PointShop
- Sampling details … later

Modeling with Point Set Surfaces

- Point set surfaces:
  - implicit connectivity info
  - no fixed continuity class
- Moving Least Squares (MLS) surfaces
  - High quality C^∞ surfaces
  - Noise & redundancy reduction
  - Progressive representations, etc.



Alexa et al., 2001

References

- David Levin, **The approximation power of moving least-squares**, *Mathematics of Computation*, 76(224), 1998.
  - David Levin, Mesh-independent surface interpolation, To appear in "Geometric Modeling for Scientific Visualization" Edited by Brunnett, Hamann and Mueller, Springer-Verlag, 2003.

- M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin and C. T. Silva, **Point Set Surfaces**, *IEEE Visualization 2001*. pp. 21-28, 2001.

Details: Moving Least Squares (MLS) Approximations

- MLS: Nonlinear projection procedure
- Goal project a point "r" onto surface
- Two steps:
  1. Define local reference domain (plane)
  2. Build local polynomial map

MLS

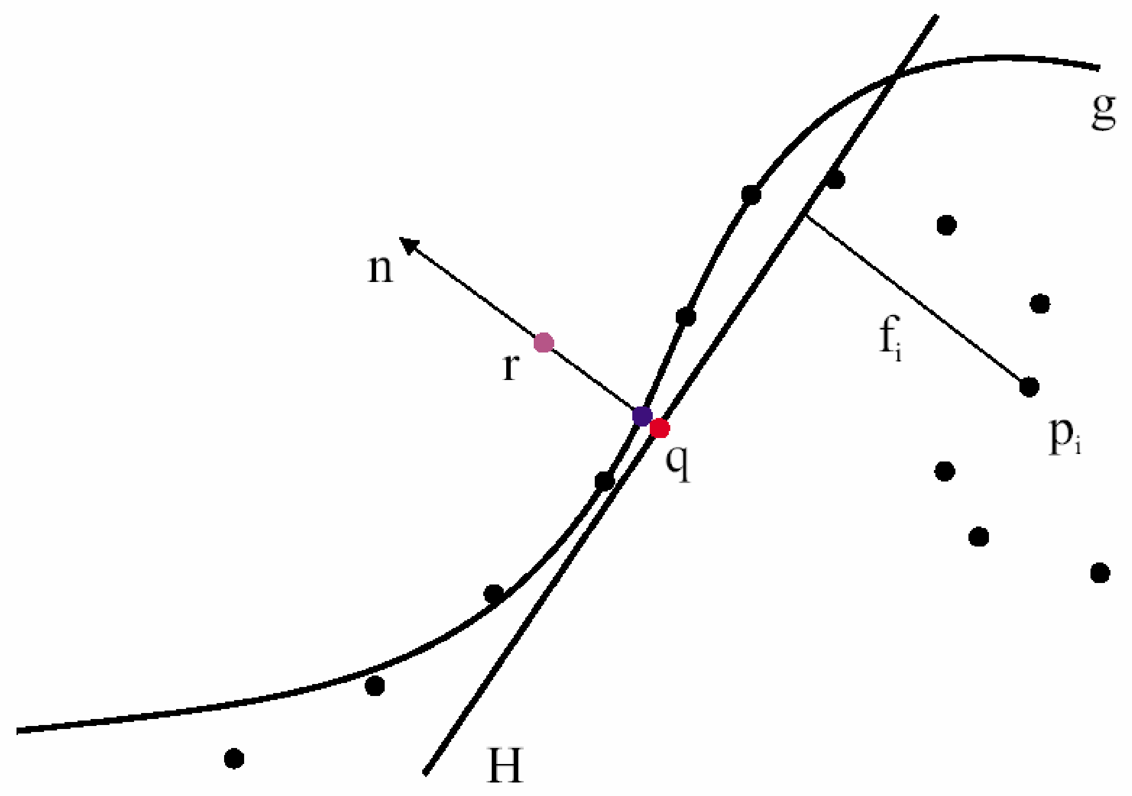


Figure 3: The MLS projection procedure: First, a local reference domain H for the purple point r is generated. The projection of r onto H defines its origin q (the red point). Then, a local polynomial approximation g to the heights f_i of points p_i over H is computed. In both cases, the weight for each of the p_i is a function of the distance to q (the red point). The projection of r onto g (the blue point) is the result of the MLS projection procedure.

Smoothing and the "h" parameter

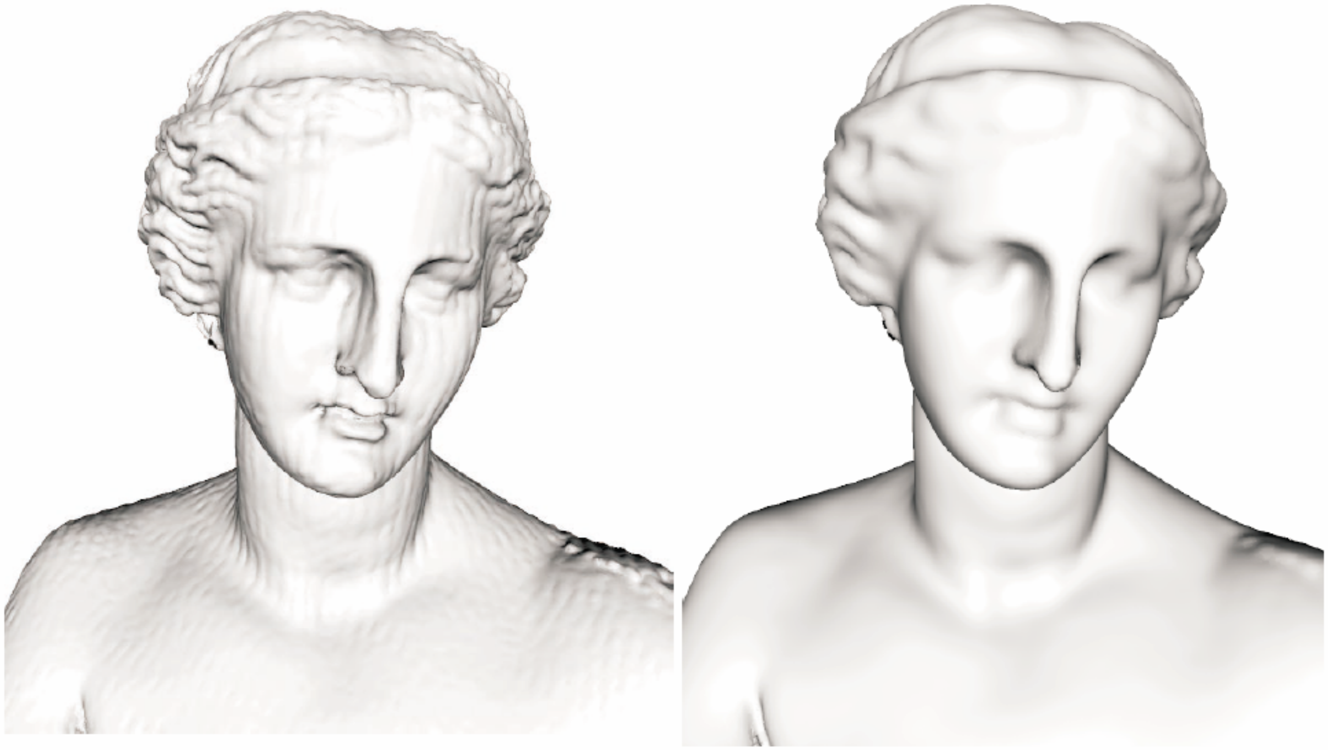
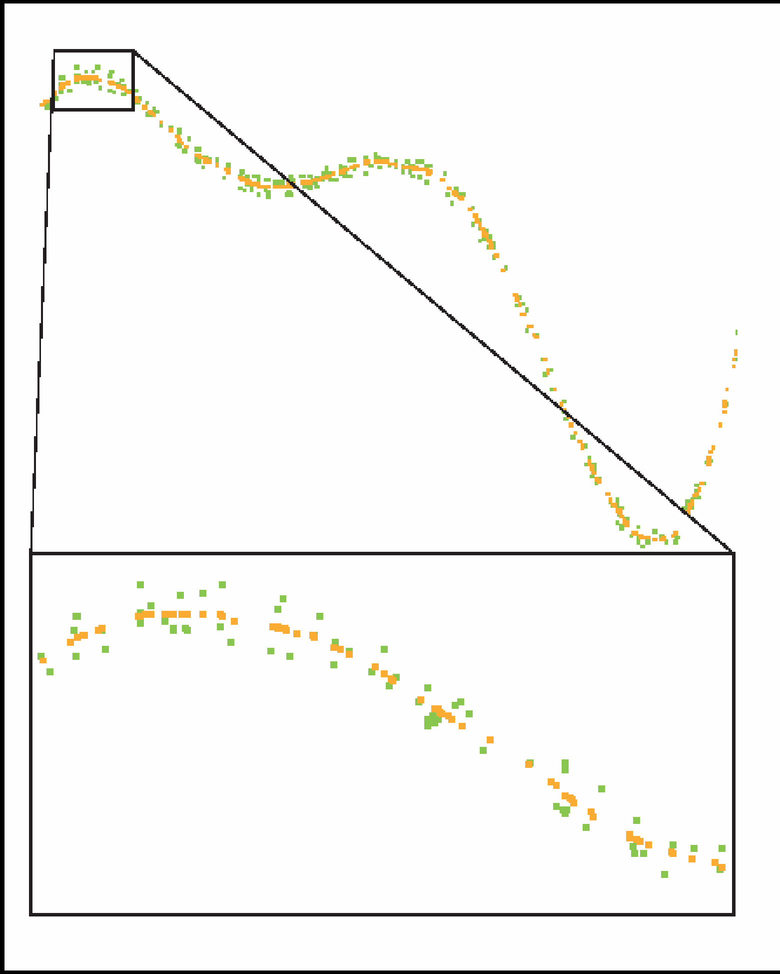


Figure 4: The effect of different values for parameter h. A point set representing an Aphrodite statue defines an MLS surface. The left side shows an MLS surface resulting from a small value and reveals a surface structure resulting from the wood carving. The right side shows a larger value for h, smoothing out small features or noise.

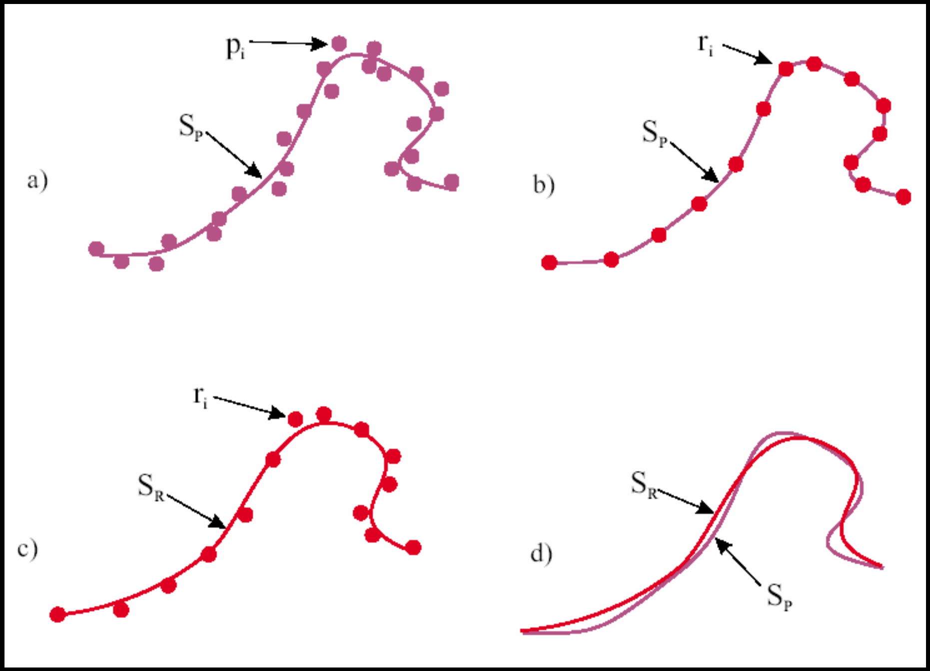
Noise Reduction

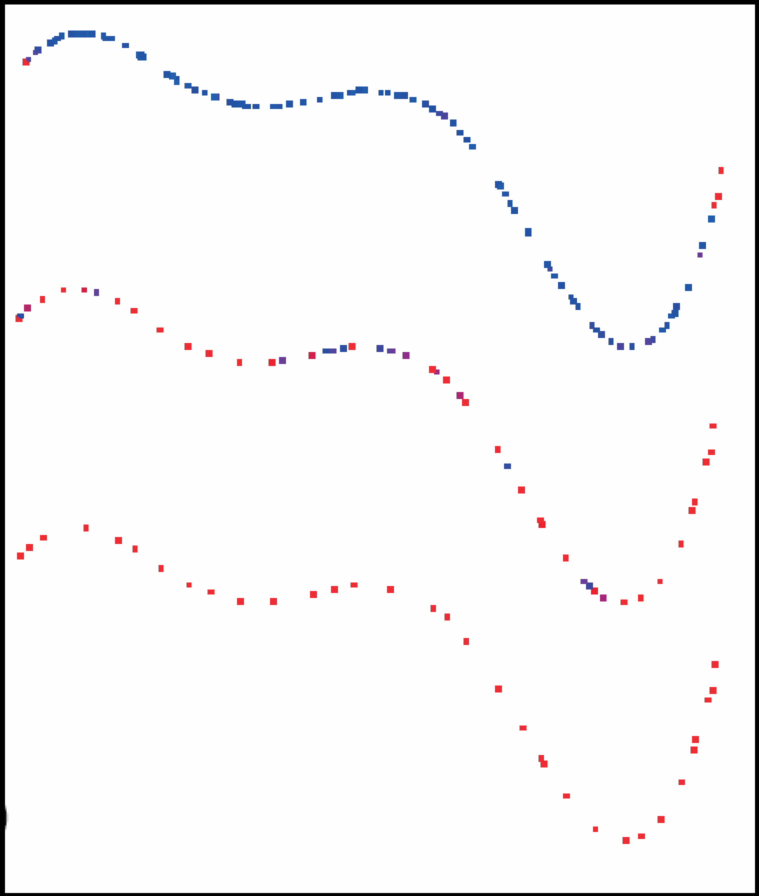
- MLS projects points onto smooth MLS 2-manifold



Down-sampling

- Useful for compact building models
- Process similar to polyhedral simplification





Up-sampling

- Refine point set to achieve desired density
- Using local approximation of Voronoi diagram

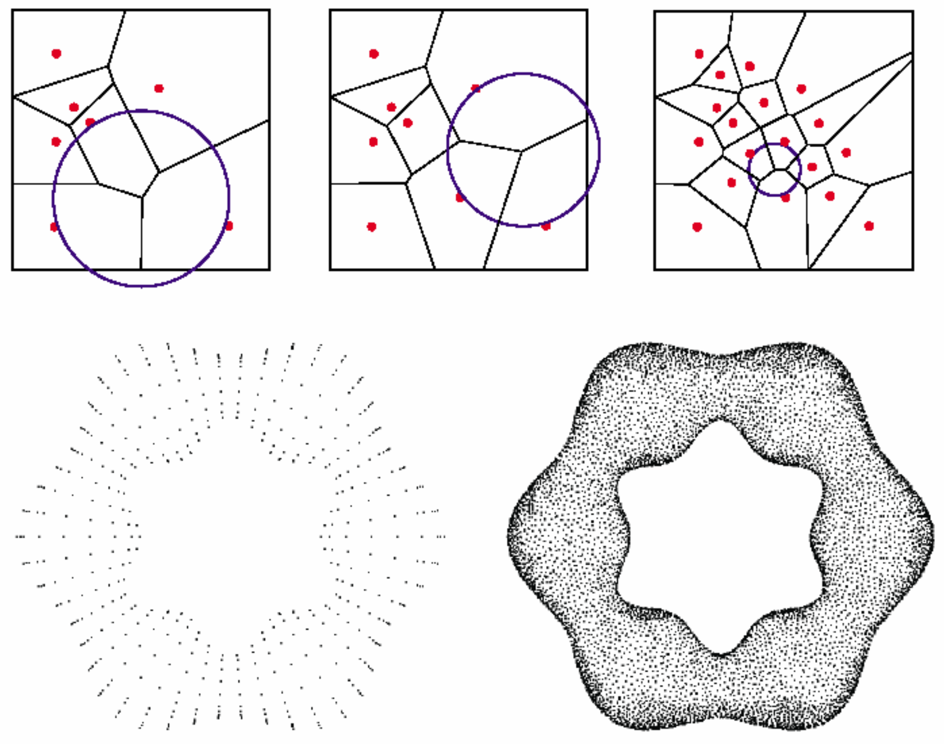


Figure 7: The up-sampling process: Points are added at vertices of the Voronoi diagram. In each step, the vertex with the largest empty circle is chosen. The process is repeated until the radius of the largest circle is smaller than a specified bound. The wavy torus originally consisting of 800 points has been up-sampled to 20K points.

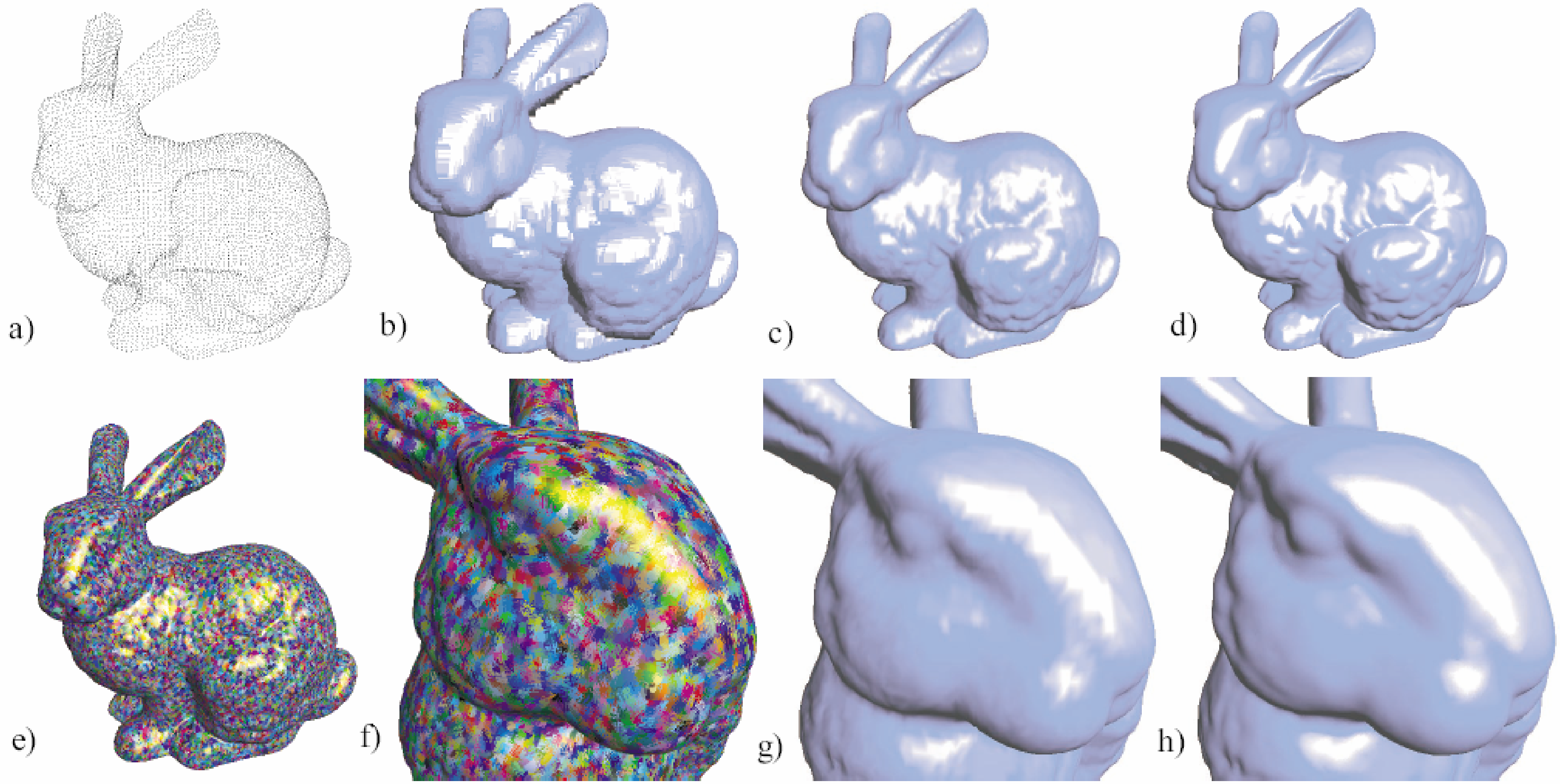


Figure 9: The Stanford Bunny: The points defining the bunny are depicted in (a) (some points are culled). Points are splatted in (b) to satisfy screen space resolution. Note the difference of a piecewise linear mesh over the points (c) and close-up in (g) to the rendering of non-conforming polynomial patches (d) and (h). The patches are color-coded in (e) and (f).

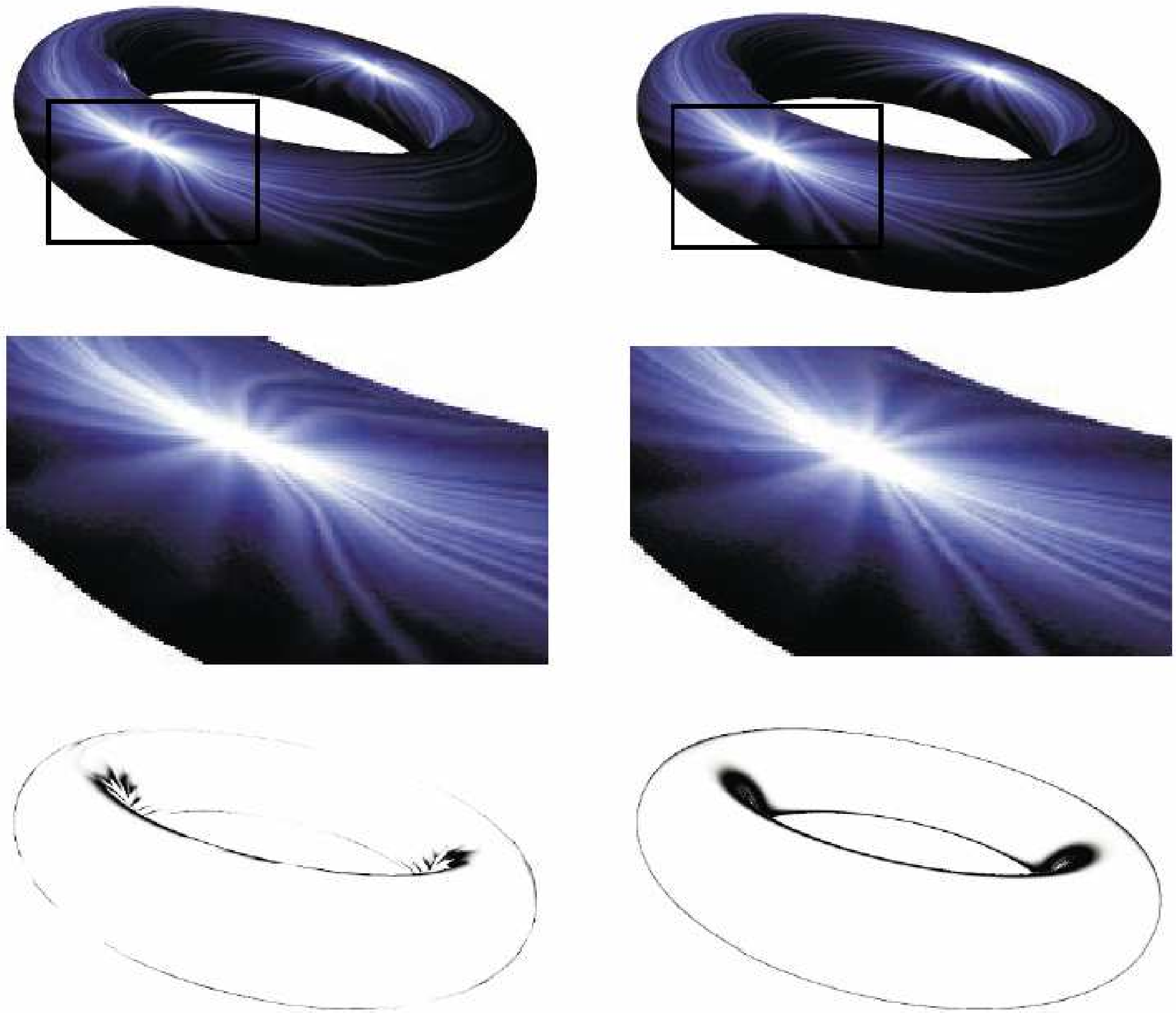


Figure 10: Comparison of mesh rendering with our technique with environment mapping. The left column shows renderings of a mesh consisting of 1000 vertices. The right column shows our technique using the vertices as input points. The environment maps emphasize the improved normal and boundary continuity.

Surfel Sampling using LDC Trees

Reference:

Hanspeter Pfister, Matthias Zwicker, Jeroen van Baar, Markus Gross.
Surfels: Surface Elements as Rendering Primitives
Proceedings of ACM SIGGRAPH 2000. pp. 335-342, 2000.

Some slides from:

IEEE Visualization 2001 Tutorial
Point-Based Computer Graphics and Visualization
Instructor: Hanspeter Pfister (pfister@merl.com)

Surfel Sampling using LDC Trees

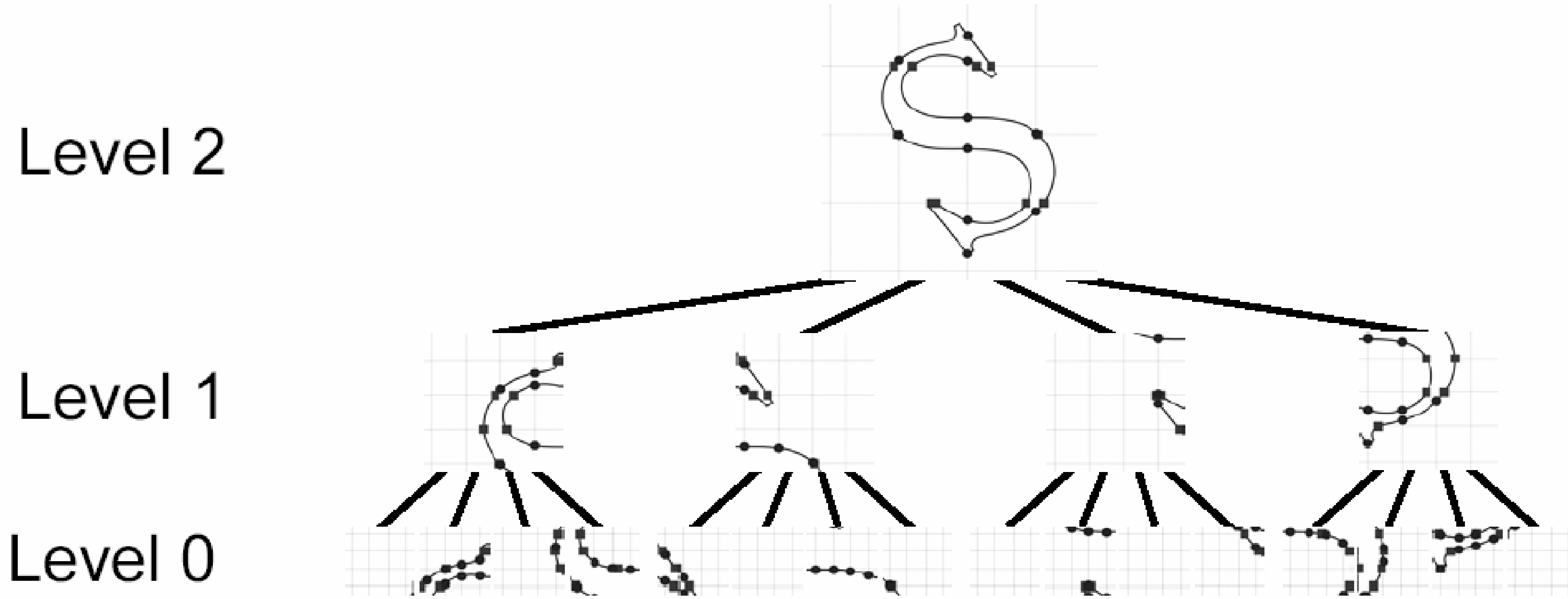
- Need guarantee on sampling density
  - Low dispersion sampling
- Pseudo image space approach
- LDC Tree…

Data Structure - LDC Tree

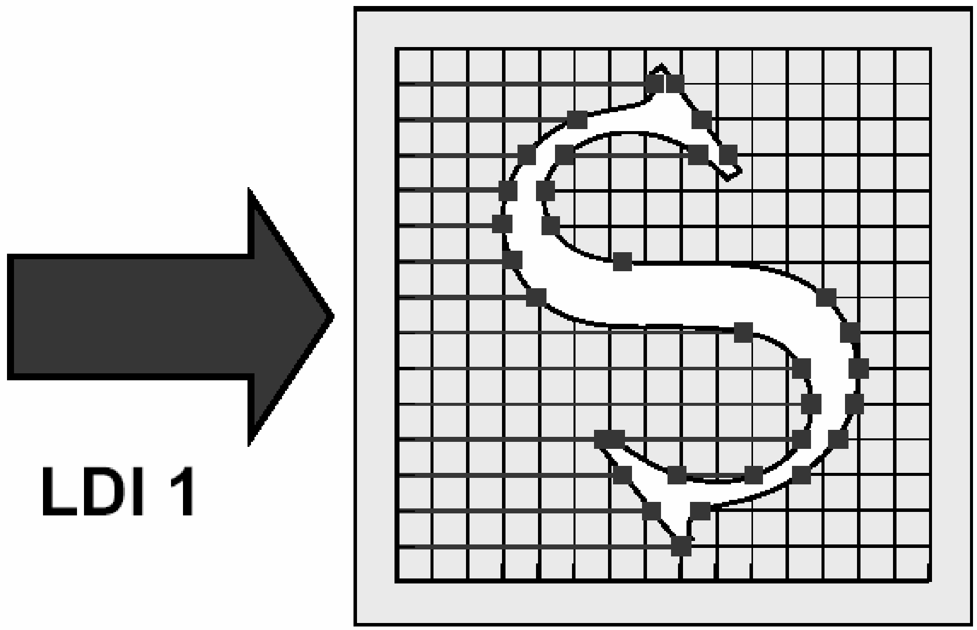
- LDC tree stands for layered depth cube tree.
- It is based on the concept of layered-depth images (LDIs).
  - [Shade et al., 1996]
- The LDC consists of 3 orthogonal LDIs.
  - [Lischinski et al., 1998]
- The LDI tree extends LDIs towards a hierarchy.
  - [Chang et al., 1999]

Data Structure - LDC Tree

- Hierarchical octree-like data structure for progressive transmission and rendering.

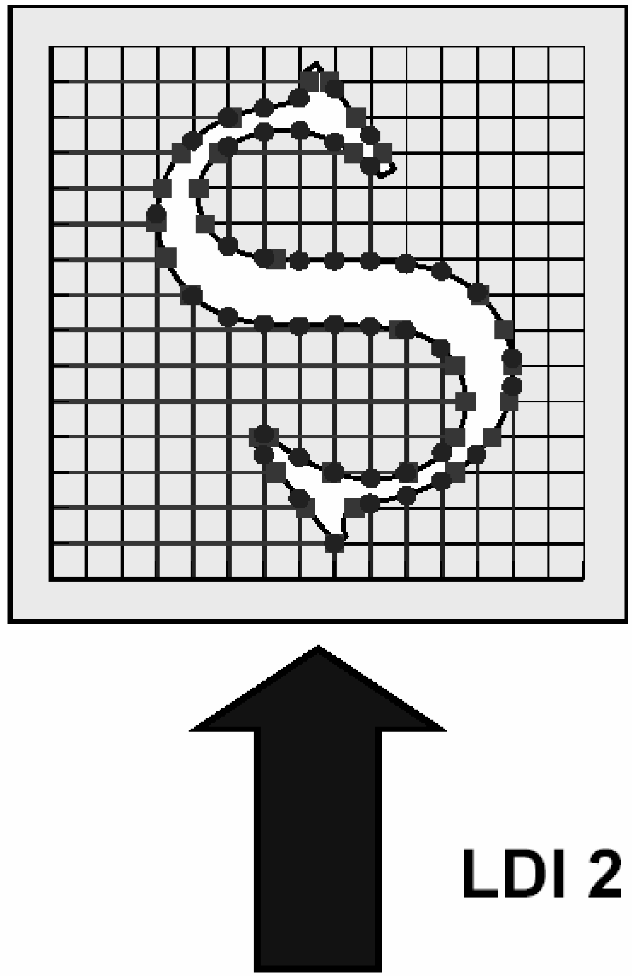


Sampling - 2D Example



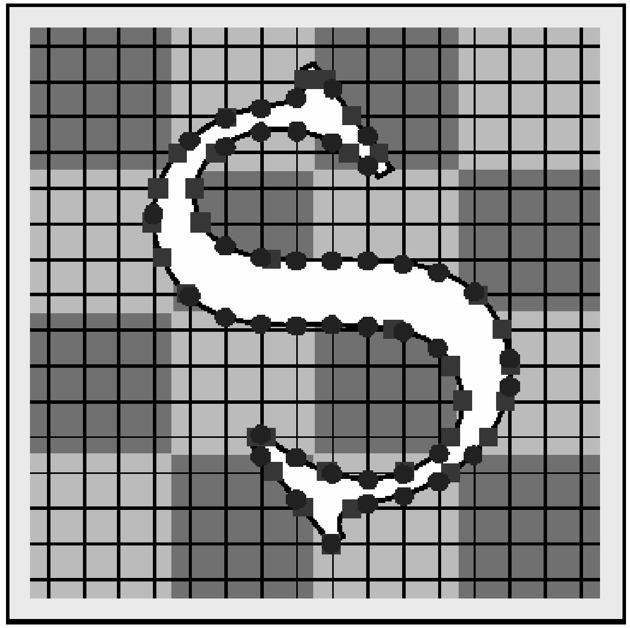
Object sampling is based on the expected image resolution.

Sampling - 2D Example



LDC Tree

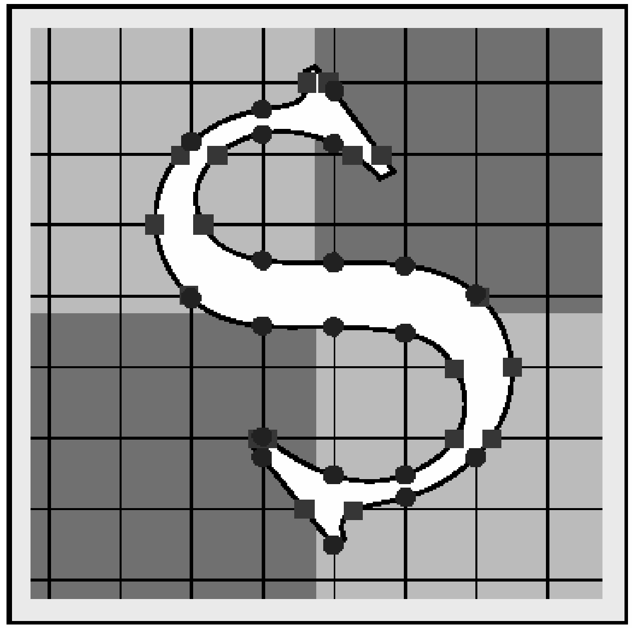
Level 0



To construct the tree, the LDC is partitioned into square blocks of size ~16 LDI pixels.
These blocks are stored at the leaves (level 0).

LDC Tree

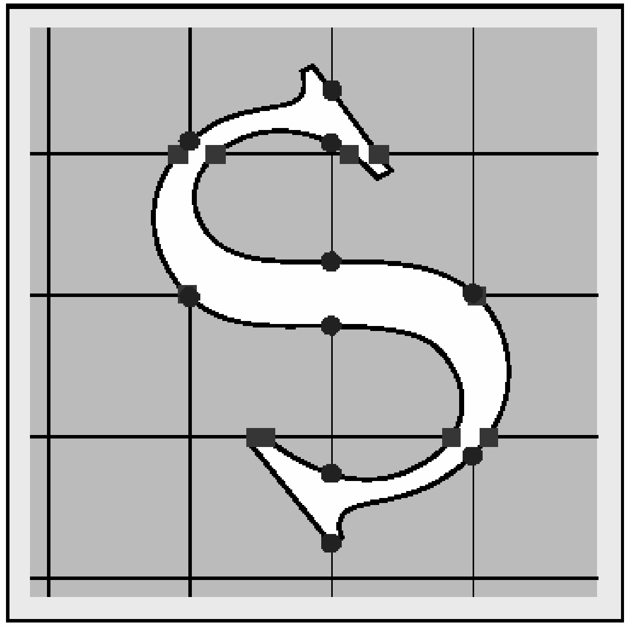
Level 1



Lower resolution levels of the LDC tree are constructed by subsampling the LDIs.

LDC Tree

Level 2



Surfels at the lower resolution levels are also present in the highest resolution level.

"3-to-1 Reduction"

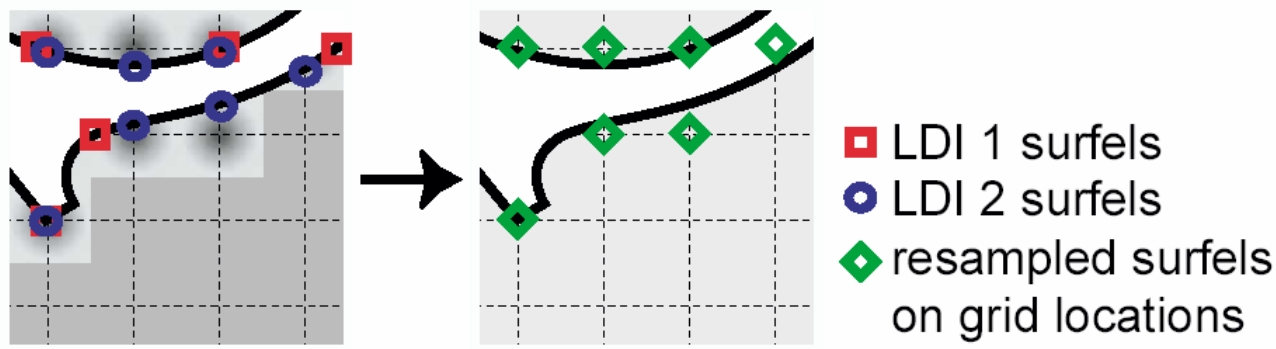


Figure 6: *3-to-1 reduction example.*

- Reduce LDC to one LDI
  - Name "3-to-1 reduction" due to rendering speedup
- Nearest neighbor interpolation
  - Quantized positions, normals (look-up table), materials, …
- Better surfel density

Rendering

- Traverse tree top to bottom.
- Visibility culling of blocks outside the viewing frustum.
- Forward projection of surfels from object to screen space using incremental forward warping. [Grossman, Dally 99]
- Visibility cones to cull blocks with backfacing surfels (computed during preprocessing).