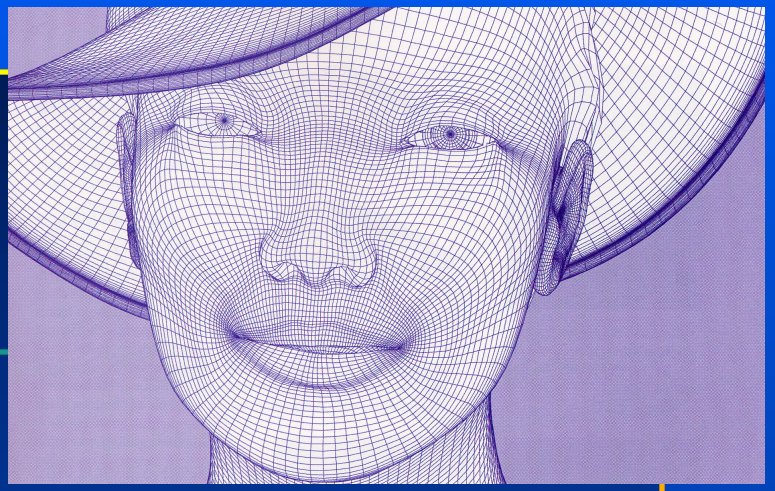
Modeling 1

Modeling

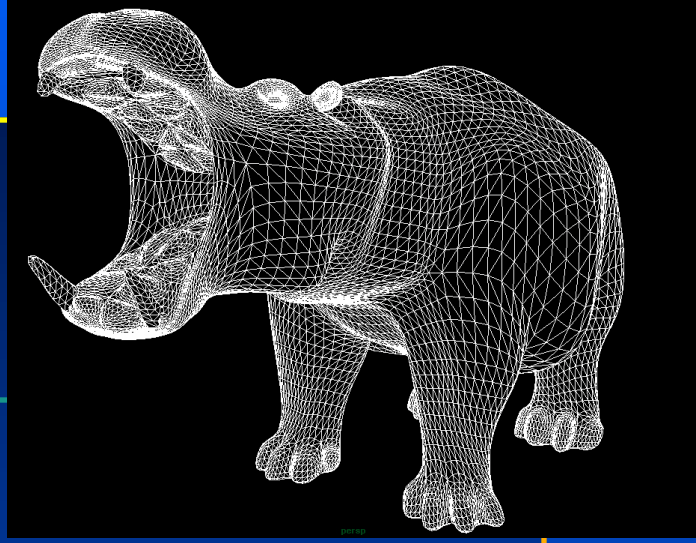


- Virtual world reproduce real world
- Real world reproduce virtual world
- Shape, physical properties, texture and color
- Rendered as
  - 2D image
  - 3D object (CAD-CAM)

Modeling Considerations

- Accuracy of representation
- Storage complexity
- Ease of manipulation
- Ease of rendering
- Level of detail?

Polygonal mesh



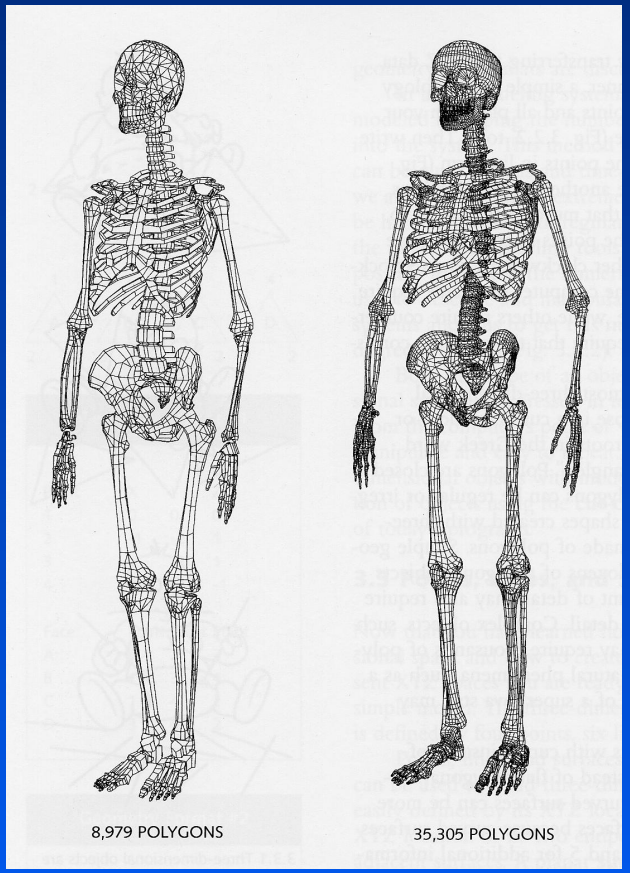
- Fast
  - Use of edge coherence makes interpolation easy
  - Hardware implementation
- Piecewise planar approximation: large number of polygons for complex curved surfaces
- Interactive sculpting/manipulation difficult
- Level of detail require sophisticated methodology
  - To prevent visual discontinuity between levels
- Texture mapping parameterization difficult

Explicit

- $P = ((X_1,Y_1,Z_1),(X_2,Y_2,Z_2),\ldots(X_n,Y_n,Z_n))$
- Shared vertices duplicated
- No representation of shared edges and vertices
- Must x-form each vertex and clip each edge
- Draw each edge twice

Pointers to vertex

- $V = ((X_1,Y_1,Z_1),\ldots(X_n,Y_n,Z_n))$
- $P_1 = (V_1,V_2,\ldots,V_n)$
- Must search to find polygons sharing edge
- Edge drawn twice



Pointers to edge

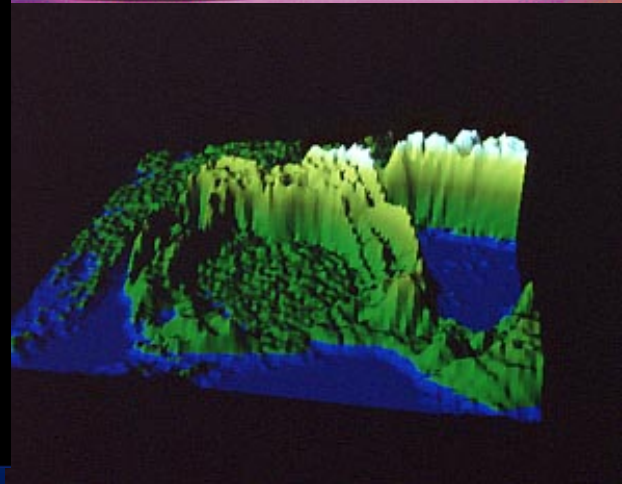
- $V = ((X_1, Y_1, Z_1), \ldots (X_n, Y_n, Z_n))$
- $E = (V_1, V_2, P_1, P_2)$ vertex list, polygon using the edge
- $P = (E_1, E_2, \ldots, E_m)$

Consistency checking

- Polygons closed
- All edges used once but not more than some max (e.g. 2)
- Each vertex referenced by at least two edges
- Polygon vertices planar

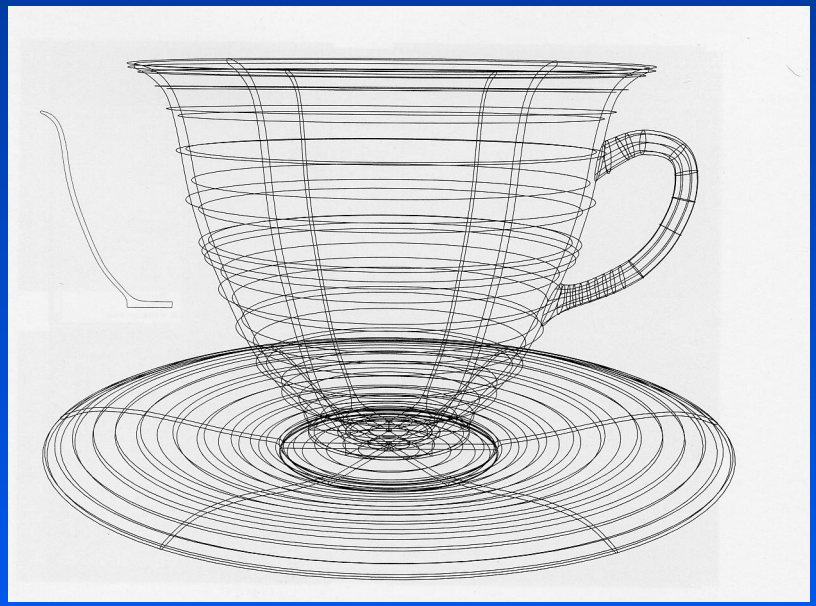
Creating Polygonal Objects

- Digitized from real models
  - Manual
  - Laser ranger, Kinect, etc.
- Construct using modeling package
  - Mathematical generation e.g. sweeps
- From mathematical description
- Algorithmic (e.g. fractals)

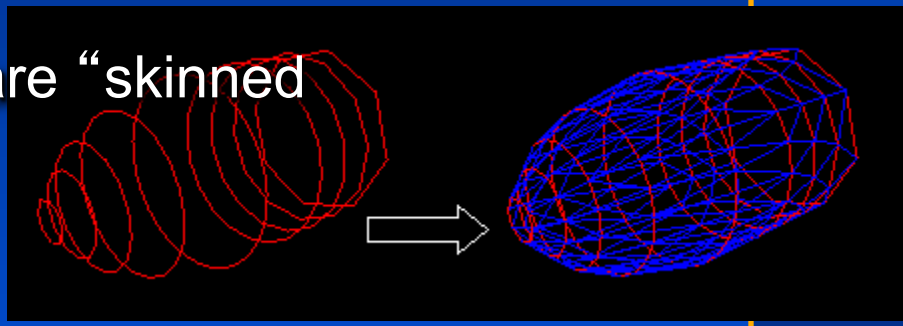


Mathematical generation: Sweeps

- 2-D forms created by drawing a 2-D geometric shape (profile)
- Sweeping to create 3-D form
- Revolution: rotate profile about an axis
  - Like a lathe or potter's wheel



- Extrusion: move profile along a straight or curved path to make 3-D form
  - Like “Play-Doh”
- Lofting: series of cross sections that are “skinned over”
  - Like the wings of an airplane

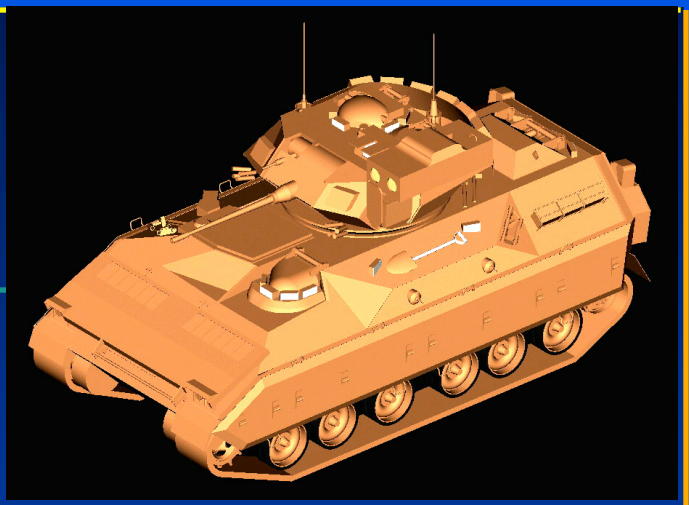


- Need to generate more polygons in areas of high curvature
- Orientation of the polygon along a path

Bi-cubic parametric patches

- Specified by mathematical formula
- Can generate any point on the surface
- Expensive to render
- Easier to manipulate using control points
- For representing arbitrary surfaces, still an approximation (it is an exact representation of itself)
- (More on this approach next chapter)

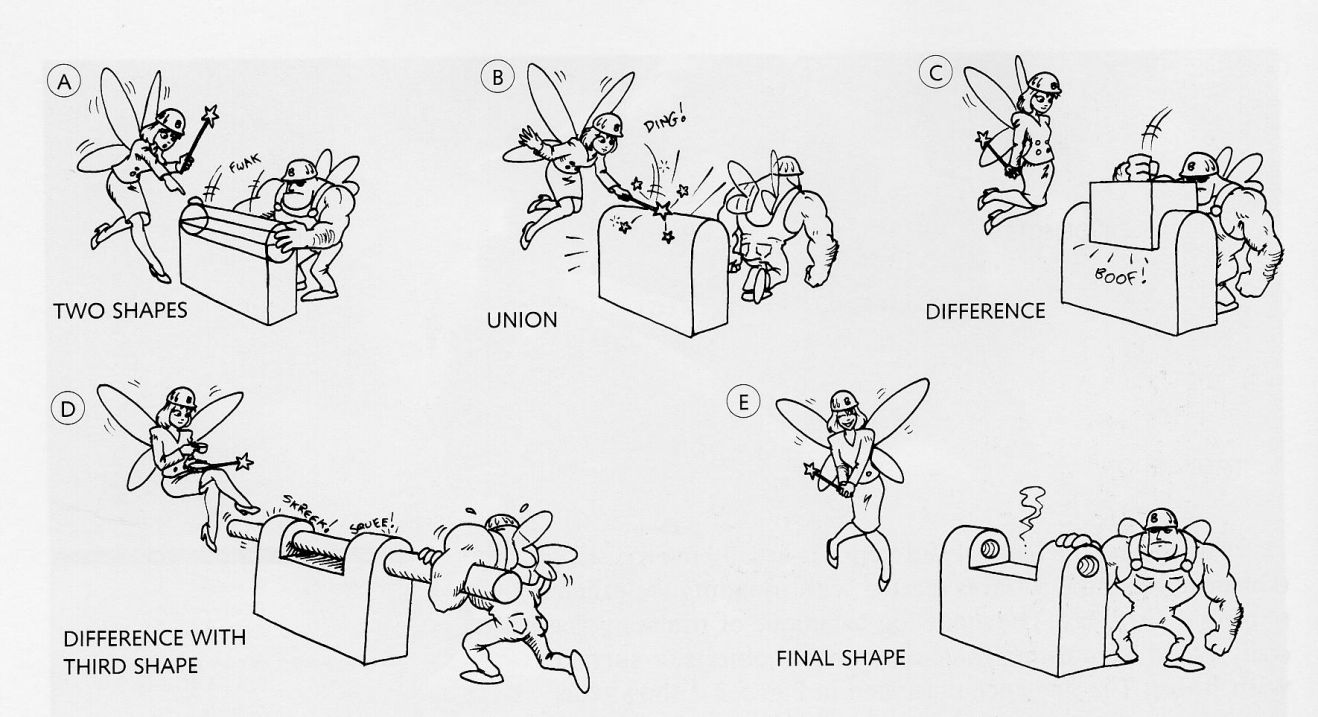
Constructive Solid Geometry (CSG)



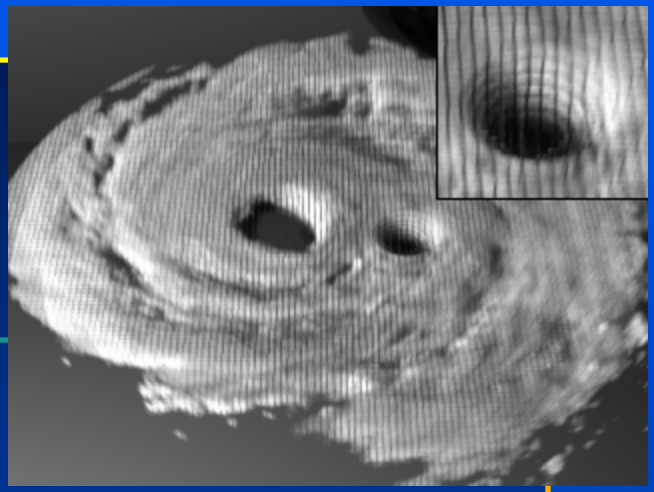
- Used for CAD/CAM, structural requirements
- Composed of operations performed on geometric primitives
  - Representation is also a record of the construction process
- Solid (as opposed to boundary representation)
- Usually converted to polygon for rendering
- Boolean operations on primitive shapes
  - Represented as a binary tree

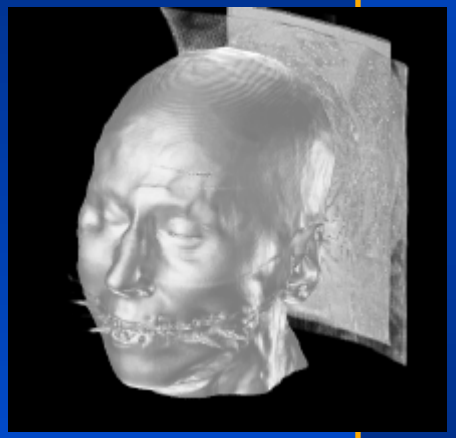
Boolean operations

- Operations on solid shapes to create other shapes
  - Intersection, union, minus, etc.
- Used both as a user interface and as representational structure
- Can easily change shapes by changing binary tree



Spatial subdivision (volumetric representation)



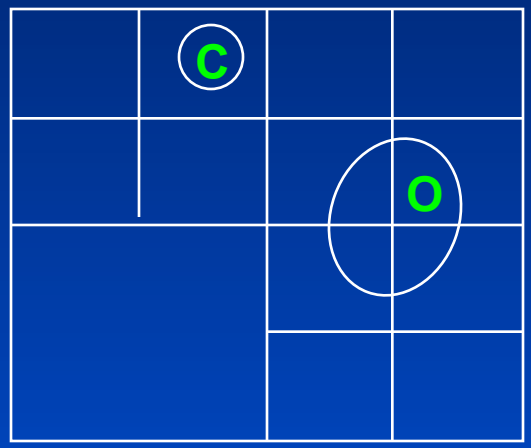


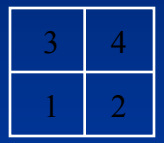
- Subdivide space into voxels
- Each voxel empty or contain part of object
- Can be represented hierarchically (e.g. octree) or as uniform subdivision
- Can be ray-traced efficiently
  - Basically DDA: line converted to voxels
  - (More on this later)
- Often seen in medical applications
  - Imaging hardware generates set of voxels
- Other objects can be converted to voxel representation

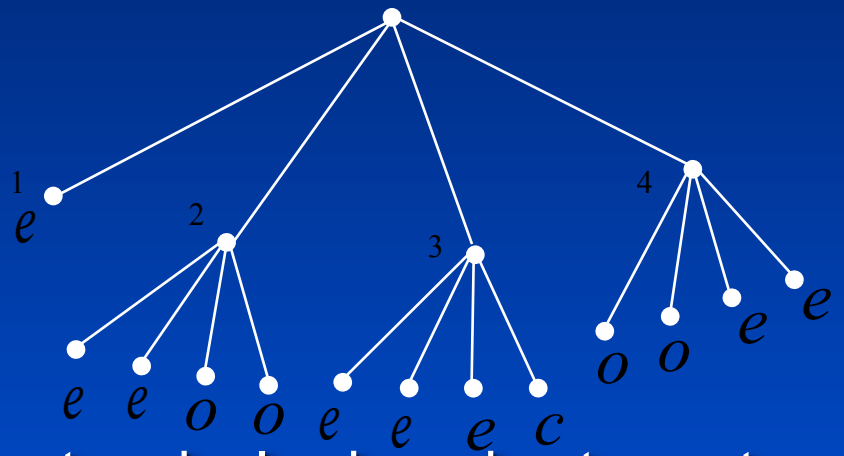
Octree

- Hierarchical representation of volume
- Can represent a volumetric object itself or be used to hold information about distribution of objects (any representation) in space
- Any region containing an object or part of object subdivided – until small enough

Quadtree (for 2D space)
Used to illustrate Octree







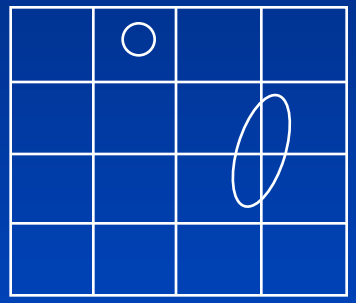
- As ray traverse space, go from one terminal sub-region to next
- As we decrease size of terminal voxel, “tight fit” but more terminal voxels
- Adoptable to scenes with varying density

BSP Tree

- Each node represent a dividing plane that separate space into two half spaces
- Any object on one side of dividing plane cannot intersect objects on the other side
- If the camera is located on one side of the plane, then objects in that plane can be considered “closer” to the camera than objects on the other side
  - Often used for hidden surface algorithm
- Dividing plane can be aligned with axes or with arbitrary orientation
- Arbitrary orientation can be used to adaptively (and efficiently) divide the spaces

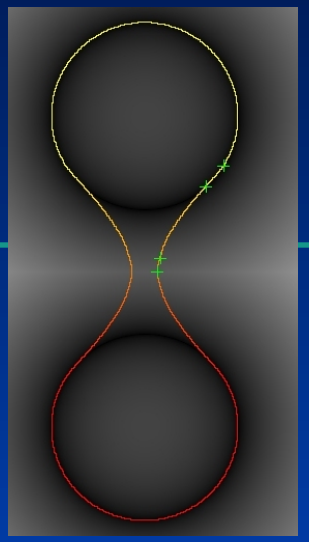
Uniform space subdivision

- SEADS (Spatially Enumerated Auxiliary Data Structure)

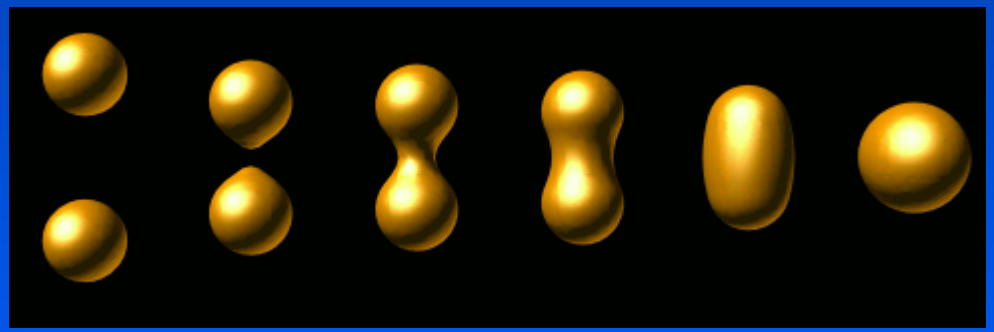


- More voxels to traverse than octrees but easier to determine neighboring voxels: DDA (Digital Differential Analyzer)
- Tradeoff resolution of grid and number of intersection tests w/ objects
- Can combine w/ octree: within a level use DDA

Implicit Surfaces



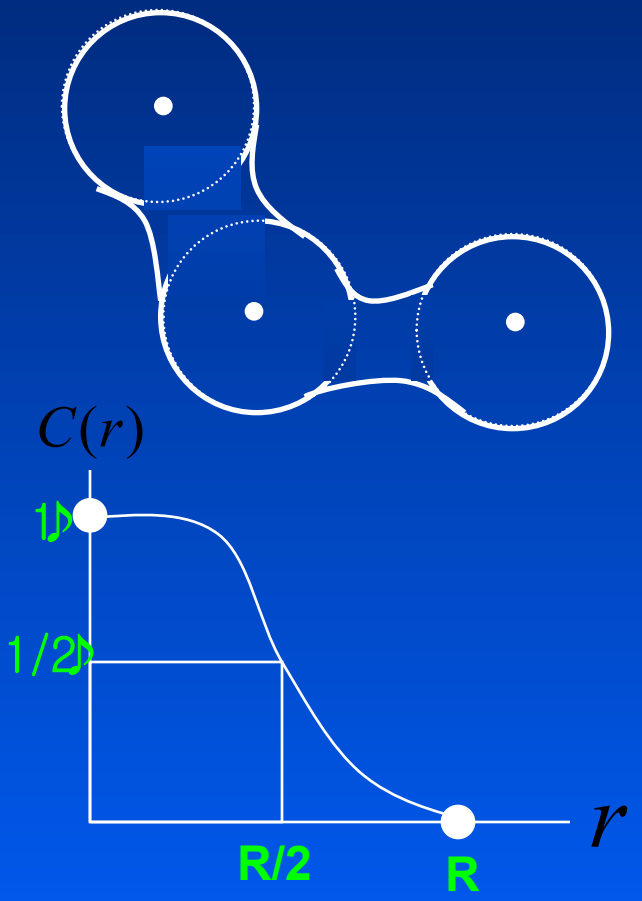
- Single functions
  - E.g. sphere $x^2 + y^2 + z^2 = r^2$
- Composed of implicit components
  - Summation of each contribution define a 3D scalar field
  - Metaballs or blobby models



- Analogous to “electron cloud” model
- Collection of field sources
- Surface defined by isosurfaces
- E.g.

$$C(r) = \begin{cases} -\frac{4}{9}r^6/R^6 + (\frac{17}{9})r^4/R^4 - (\frac{22}{9})r^2/R^2 + 1 \\ \text{for } 0 \le r \le R \\ 0 \\ \text{for } \; R < r \end{cases}$$

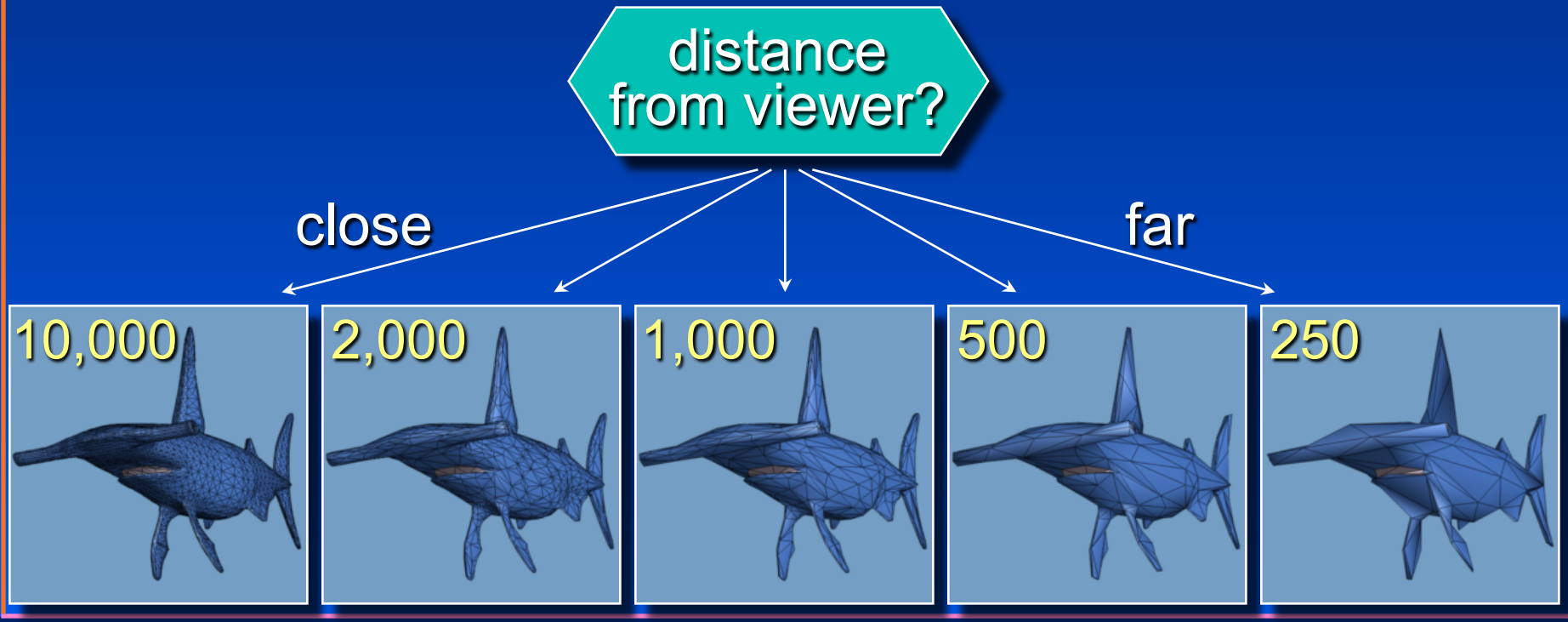
- Amorphous surface generated by moving the component sources
- Problem with unwanted combination of fields (e.g. when used to define the "skin" of an articulated figure
- Render isosurfaces e.g. marching cubes (later)



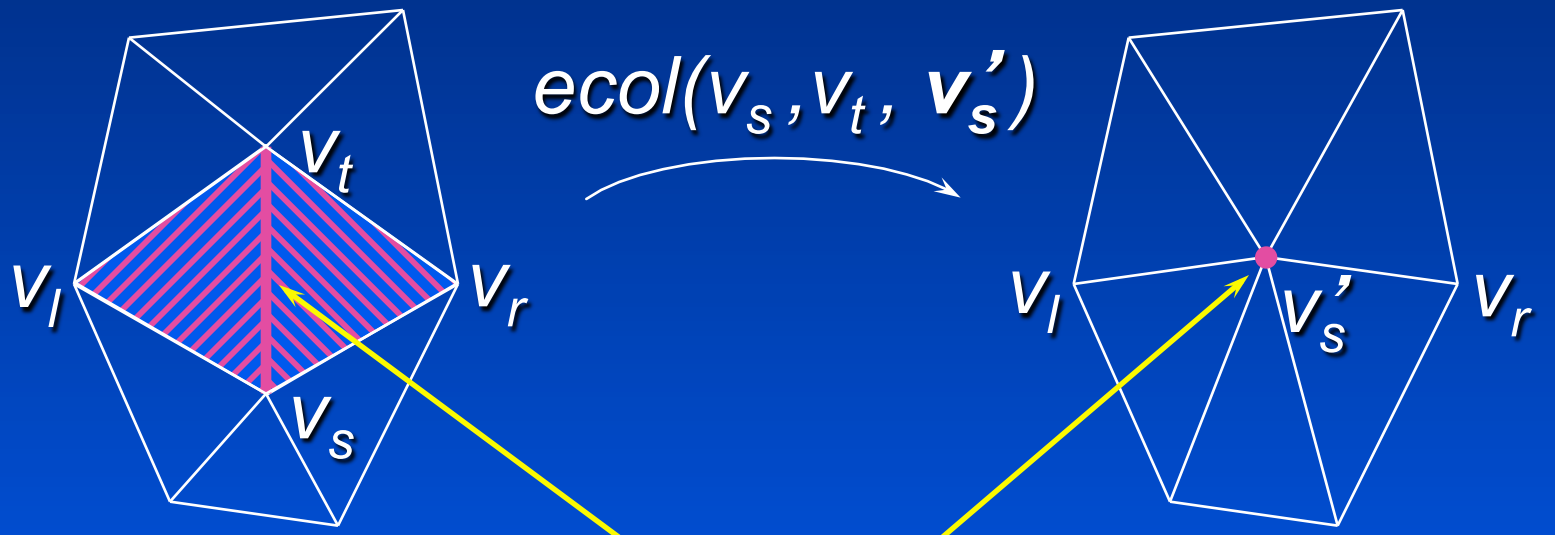
Polygon Mesh Optimization

- One level of detail is not sufficient
  - Either too coarse or inefficient by rendering large number of polygons onto small number of pixels
- Hierarchy of mesh simplification
  - Use appropriate level depending on projected size of polygon in image space (similar to MIP maps)
  - Discontinuous switching levels leads to “popping”
  - Geomorph: smooth morphing between geometry at adjacent levels

From “Progressive Meshes” Hughes Hoppe, SIGGRAPH 96



Edge Collapse



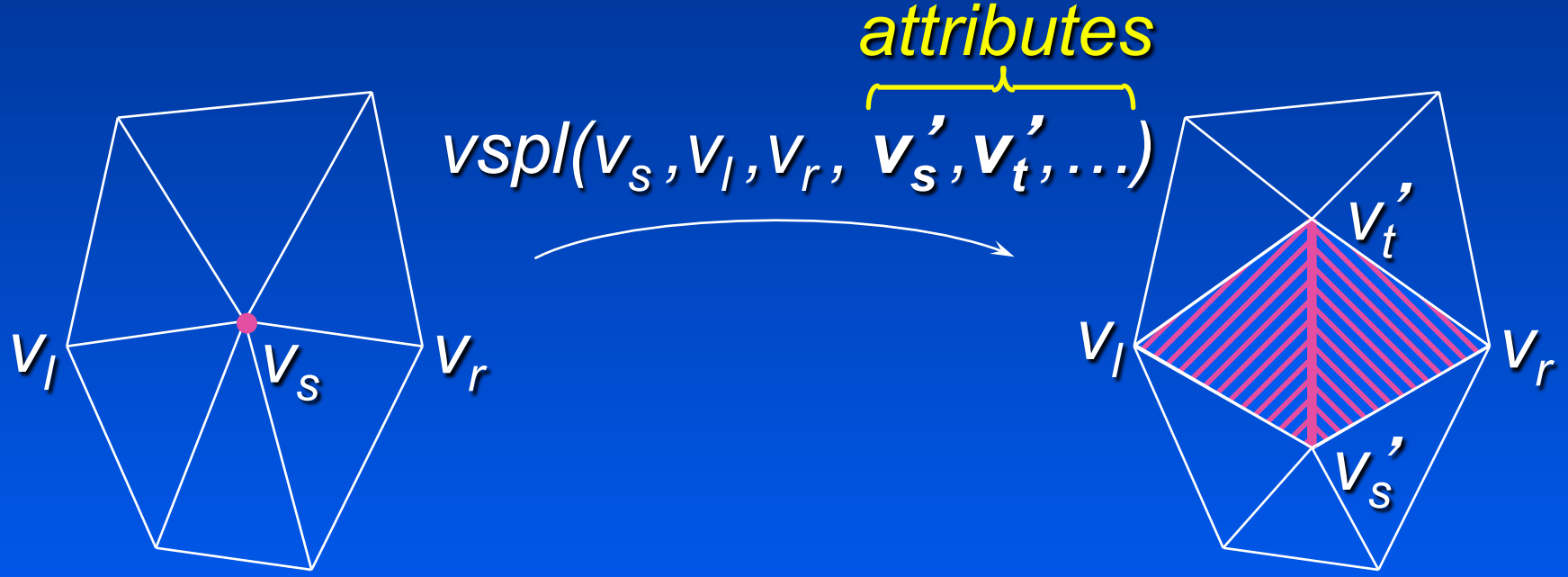
(optimization)

- From original mesh, perform series of edge collapses

$\hat{M}=M^n \longrightarrow \cdots \longrightarrow M^{175} \cdots \longrightarrow M^1 \longrightarrow M^0$

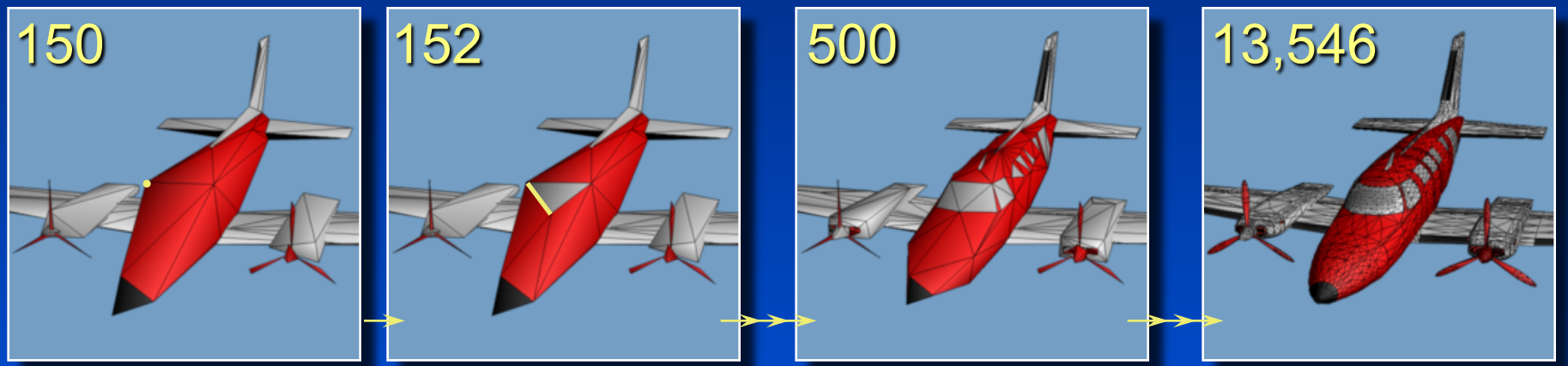
$ecol_{n-1}$ $ecol_i$ $ecol_0$

- Invertible by vertex split operation



Reconstruction process

- From base mesh, can generate original mesh: vertex split

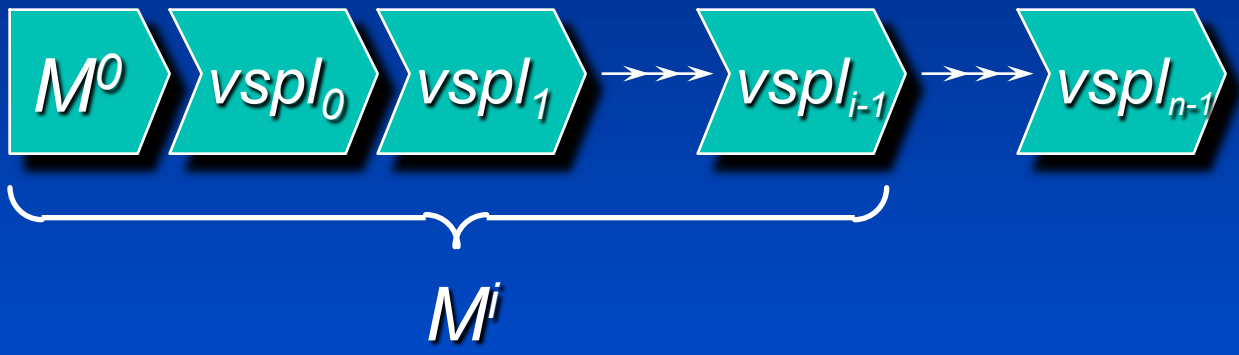


$M^0 \longrightarrow M^1 \longrightarrow \cdots \longrightarrow M^{175} \longrightarrow \cdots \longrightarrow M^n = \hat{M}$

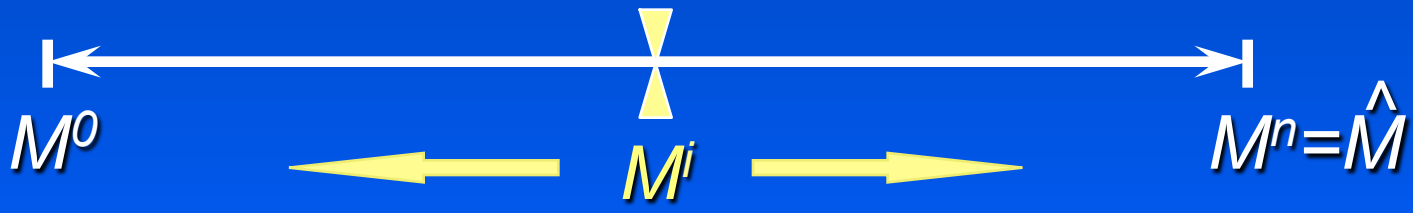
$vspl_0 \quad \ldots \quad vspl_i \quad \ldots \quad vspl_{n-1}$

progressive mesh (PM) representation

- Result in continuous LOD without discontinuities

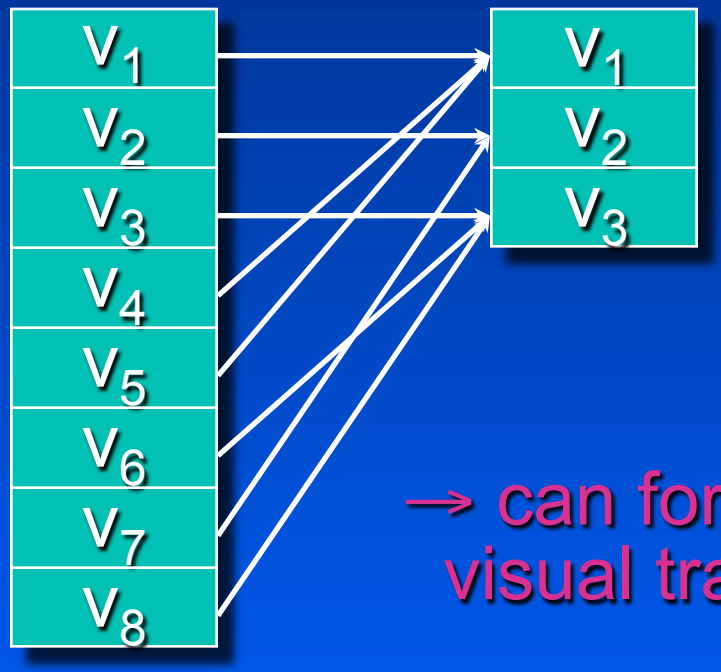


3,478 faces?



Geomorph

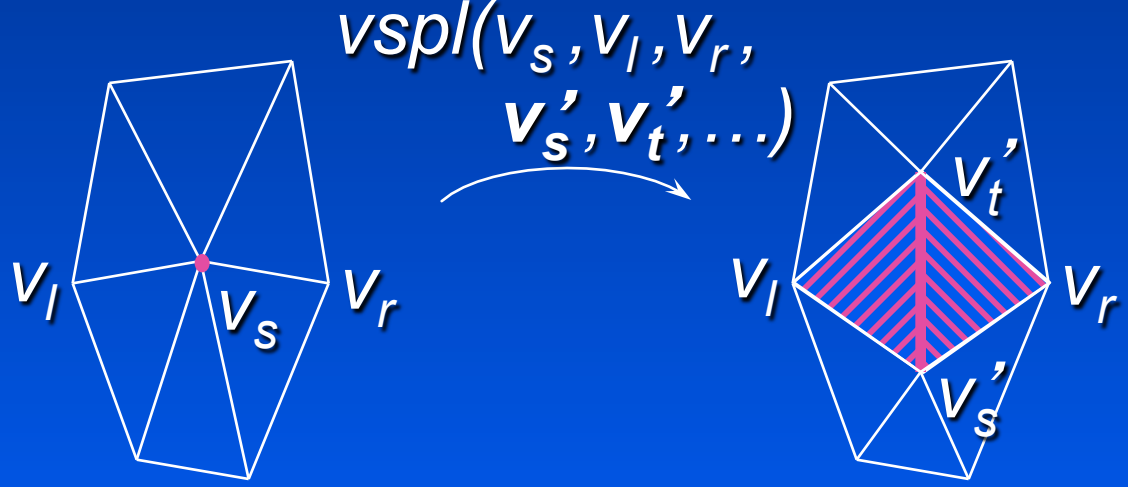
- Correspondence between vertices allow smooth morph between LOD



→ can form a smooth visual transition: *geomorph*

Mesh Compression

- Rather than storing each mesh explicitly, store just the deltas to generate successive (higher) LOD

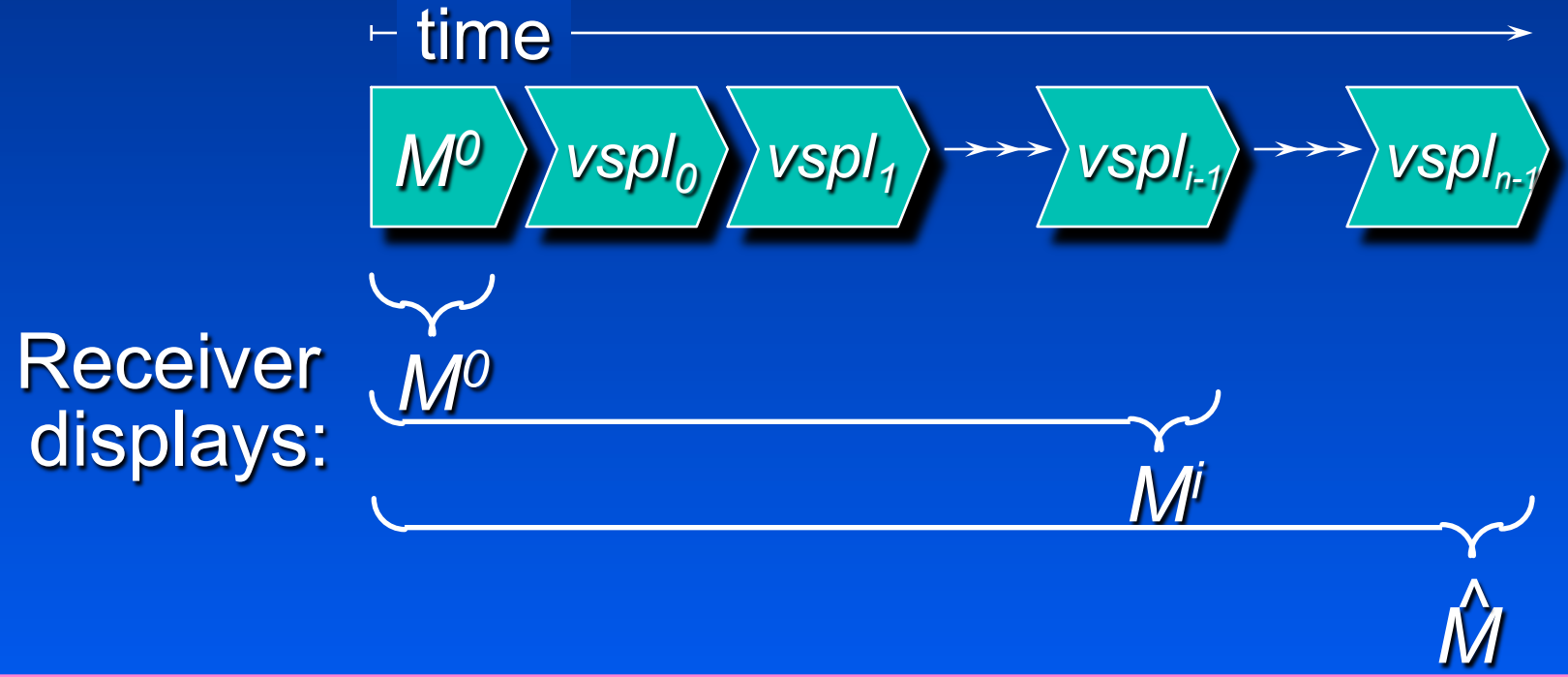


Record deltas:

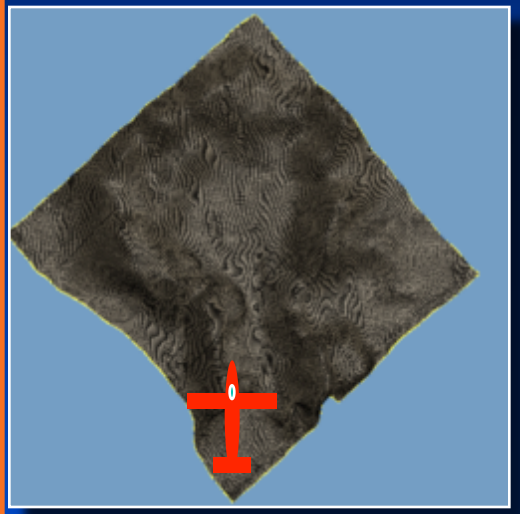
- $v_t' - v_s$
- $v_s' - v_s$
- ...

Progressive Transmission

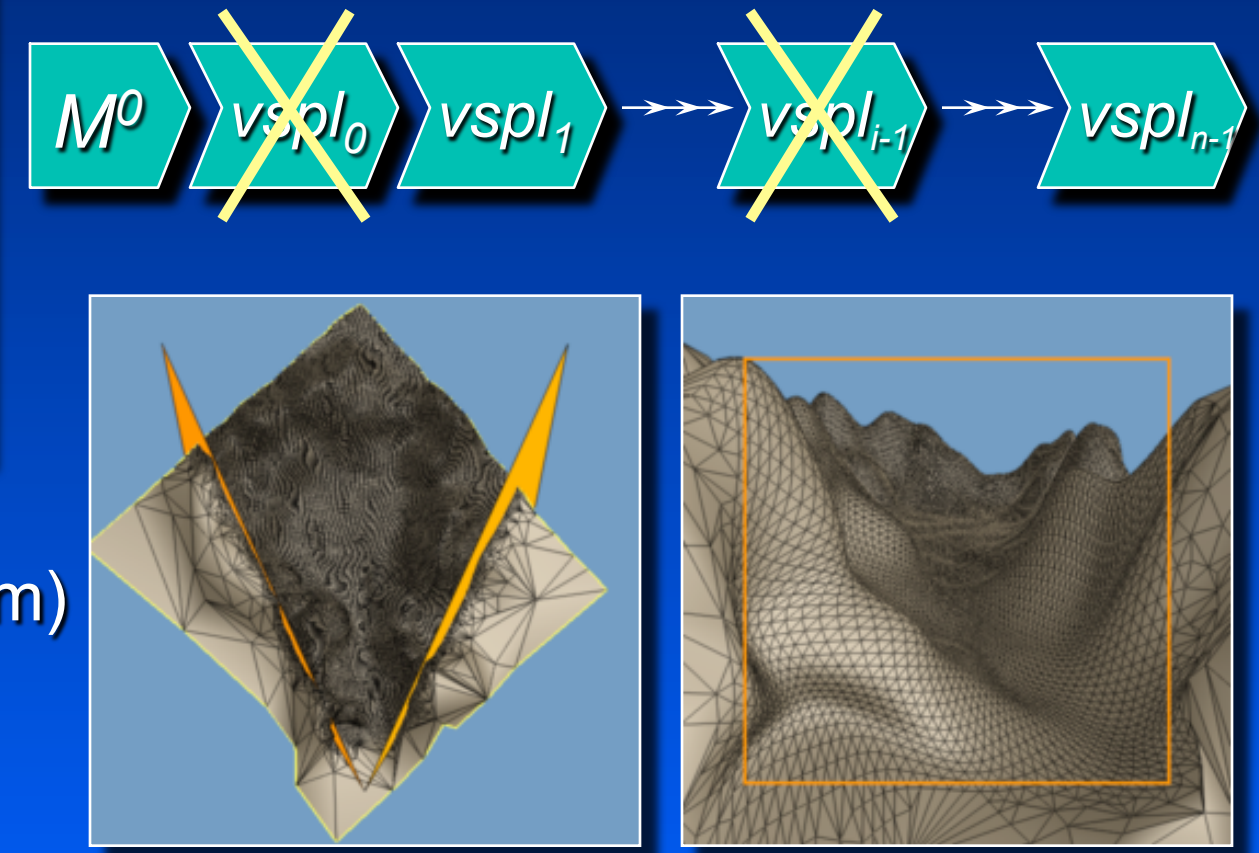
- Similar to JPEG
- Successive LOD transmitted to receiver to be rendered



Selective Refinement



(e.g. view frustum)



Selecting Edge Collapse

- Preserve appearance: minimize energy function consisting of
  - Geometric shape
  - Scalar field (e.g. color)
  - Preserve discontinuities (e.g. crease)

$$E = \sum_{\text{points}} \int (e_{shape} + e_{scalars}) dA + \sum_{\text{points}} \int (e_{disc}) dL$$

- For shape, energy functions given by summing distances between the potential new points and the original surface
- For the scalar field, sum the difference between potential new and original scalar field
- For discontinuities, sum difference between potential new and original discontinuities
- At each step, choose edge that will minimize energy function
  - Local optimization easy, global optimization better

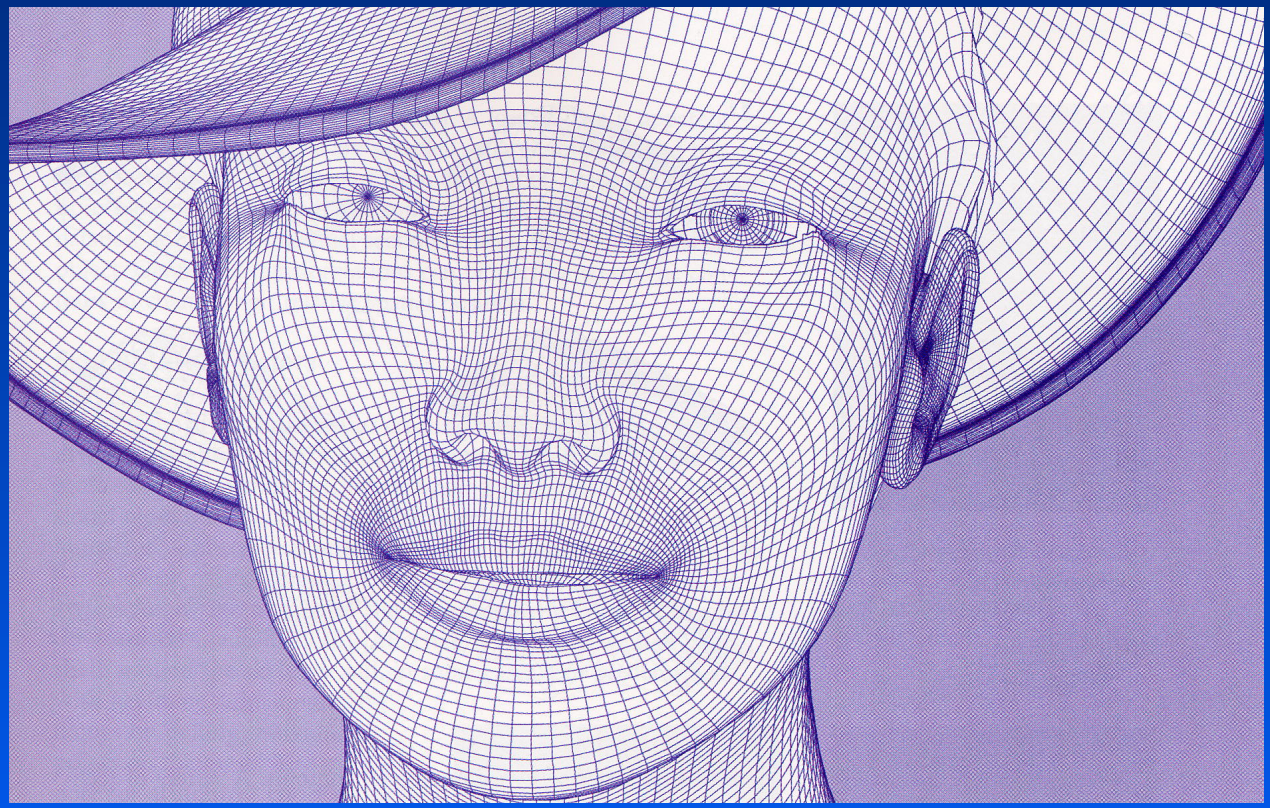
Disadvantages of current modeling approach

- Artificial introduction of additional primitives (e.g. polygons)
- Customization of display algorithms to particular primitives
- Less coherence as complexity of objects increase
- Surface scanning (e.g. Laser, Kinect) produces point clouds
- For optimal detail, size of primitives must approach size of pixels
  - Otherwise, primitives show up in the rendered image
  - Already an issue, e.g. Pixar uses polygons that are smaller than pixels
- Conflict between best representation for modeling vs rendering

Alternatives: Point clouds

- In some sense, the rendered image (consisting of pixels) is a point representation of the surface
- Problem: how to minify (many surface points to one pixel) or magnify (many pixel to one surface point)
  - Apply a convolution filter
  - Variation of “splatting” (later)

Next: Parametric Curves and Surfaces



Virtual Celebrity/Marlene Inc.