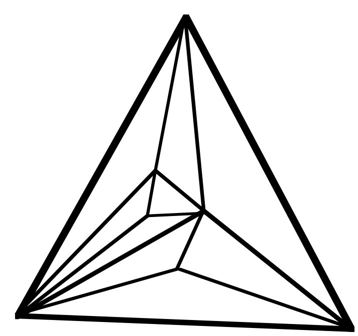
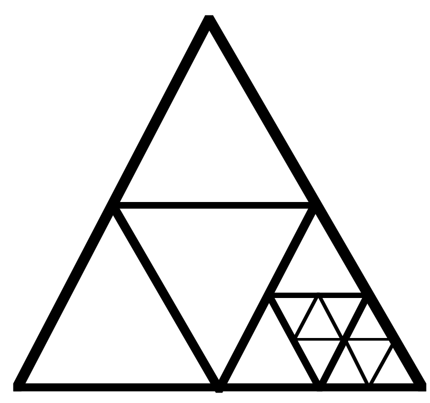
SURFACE DATA

HIERARCHICAL TRIANGULAR DECOMPOSITION

- Approximate surface *S* by planar triangular patches whose vertices are a subset of data points defining *S*
- For each patch, compute an approximation error
  1. maximum error of the data points with projections on the x-y plane overlapping the projection of the patch on the x-y plane
  2. subdivide further if a predefined tolerance is exceeded
- Types
  1. ternary (De Floriani, Falcidieno, Nagy, and Pienovi)
     - use internal point with maximum deviation and decompose the triangle into three triangles by joining it to the vertices
     - surface is continuous at every level
     - triangles are thin and elongated

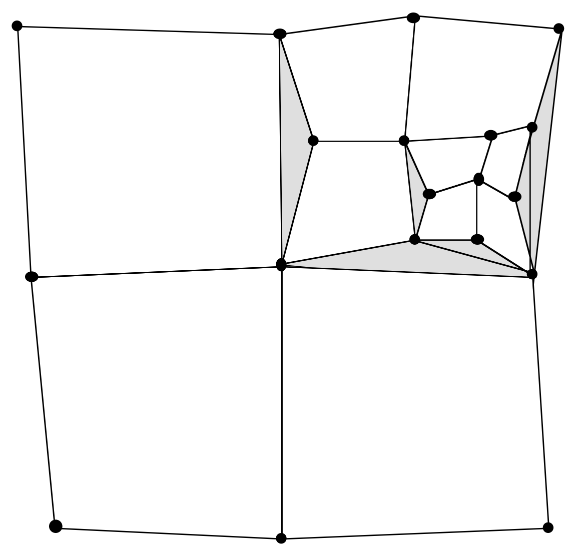


  2. quaternary (Gomez and Guzman)
     - select three points on the sides of the triangle and form four new triangles by connecting them
     - each triangle can be adjacent to many triangles
     - interpolating surface is not continuous unless all triangles are split uniformly—i.e., the result is a complete quadtree
     - good when points are drawn from a uniformly spaced grid



HIERARCHICAL RECTANGULAR DECOMPOSITION

- Similar to triangular decomposition
- Good when data points are the vertices of a rectangular grid
- Drawback is absence of continuity between adjacent patches of unequal width (termed the *alignment problem*)

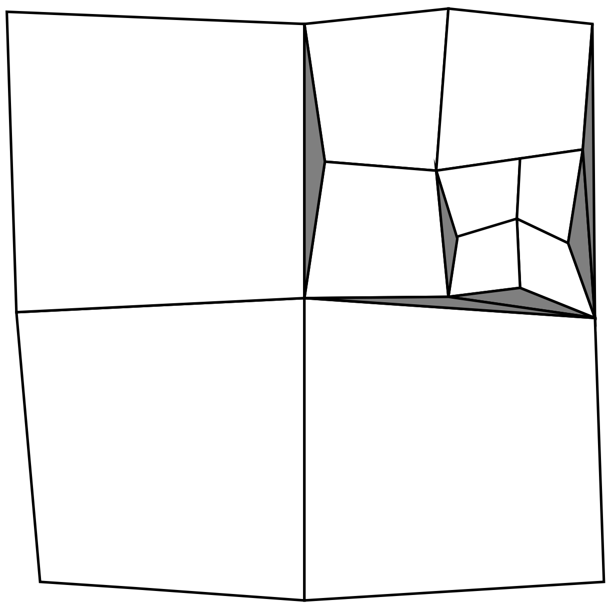


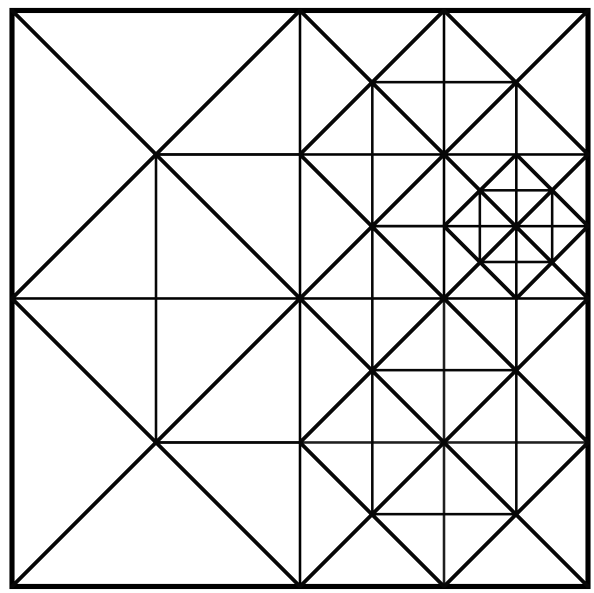
- Overcoming the presence of cracks
  1. use the interpolated point instead of the true point (Barrera and Hinjosa)
  2. triangulate the squares (Von Herzen and Barr)
     - can split into 2, 4, or 8 triangles depending on how many lines are drawn through the midpoint
     - if split into 2 triangles, then cracks still remain
     - no cracks if split into 4 or 8 triangles

RESTRICTED QUADTREE (VON HERZEN/BARR)

- All 4-adjacent blocks are either of equal size or of ratio 2:1

  Note: also used in finite element analysis to adptively refine an element as well as to achieve element compatibility (termed *h-refinement* by Kela, Perucchio, and Voelcker)

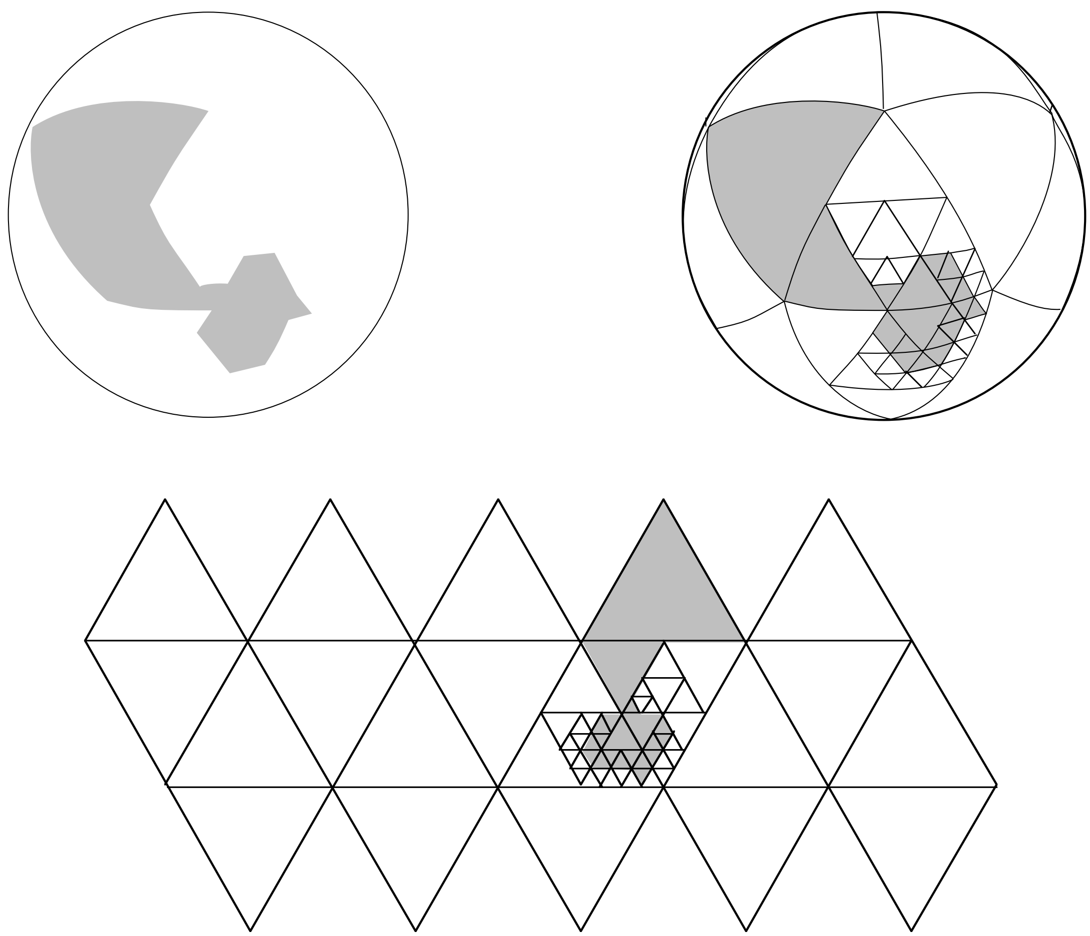




- 8-triangle decomposition rule
  1. decompose each block into 8 triangles (i.e., 2 triangles per edge)
  2. unless the edge is shared by a larger block
  3. in which case only 1 triangle is formed
- 4-triangle decomposition rule
  1. decompose each block into 4 triangles (i.e., 1 triangle per edge)
  2. unless the edge is shared by a smaller block
  3. in which case 2 triangles are formed along the edge
- Prefer 8-triangle rule as it is better for display applications (shading)

PROPERTY SPHERES (FEKETE)

- Approximation of spherical data
- Uses icosahedron which is a Platonic solid
  1. 20 faces—each is a regular triangle
  2. largest possible regular polyhedron

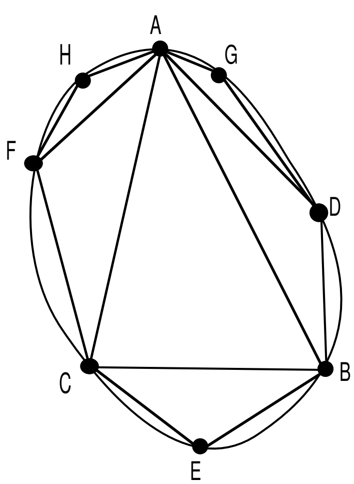


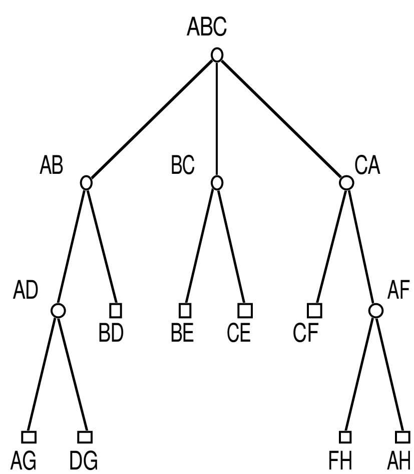
ALTERNATIVE SPHERICAL APPROXIMATIONS

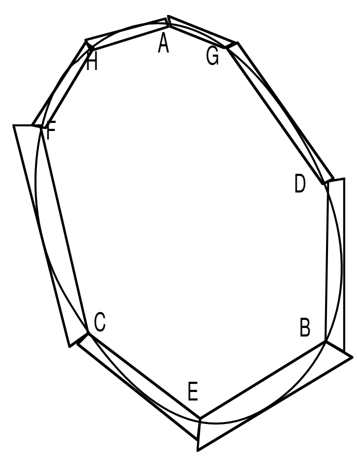
- Could use other Platonic solids
  1. all have faces that are regular polygons
     - tetrahedron: 4 equilateral triangular faces
     - hexahedron: 6 square faces
     - octahedron: 8 equilateral triangular faces
     - dodecahedron: 12 pentagonal faces
  2. octahedron is nice for modeling the globe
     - it can be aligned so that the poles are at opposite vertices
     - the prime meridian and the equator intersect at another vertex
     - one subdivision line of each face is parallel to the equator
- Decompose on the basis of latitude and longitude values
  1. not so good if want a partition into units of equal area as great problems around the poles
  2. project sphere onto plane using Lambert’s cylindrical projection which is locally area preserving
- Instead of approximating sphere with the solids, project the faces of the solids on the sphere (Scott)
  1. all edges become sub-arcs of a great circle
  2. use regular decomposition on triangular, square, or pentagonal spherical surface patches

2D PRISM TREE (PONCE/FAUGERAS)

- Hierarchical representation of an approximation used to model surfaces of 3d polyhedral objects with no holes
- Based on polyhedral approximation
  1. initially model by a triangle with vertices on the surface so surface is decomposed into three segments
  2. for each segment of the surface pick the point *M* at a maximum distance from the triangle
  3. if the distance from *M* is greater than a predefined threshold, then subdivide into two triangles
  4. repeat until all approximations are within threshold
- Can be represented as a tree structure
- Approximate each boundary segment by a quadrilateral
  1. easier to manipulate a boundary (e.g., detecting intersections between boundaries of adjacent objects)
  2. result is 2d prism tree

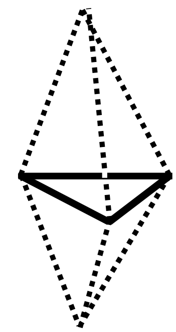




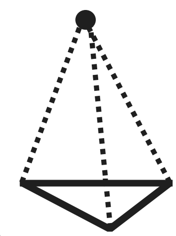


3D PRISM TREE: CONSTRUCTION

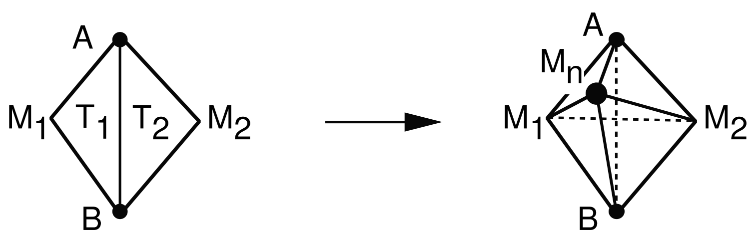
- Use tetrahedra instead of triangles
  1. start with a triangle that splits the surface into 2 segments
  2. for each segment pick a point on the surface at a maximum distance from the triangle
     - results in two tetrahedra which is a triangular bipyramid (= hexahedron)



  3. decompose the triangular faces into tetrahedra
     unless



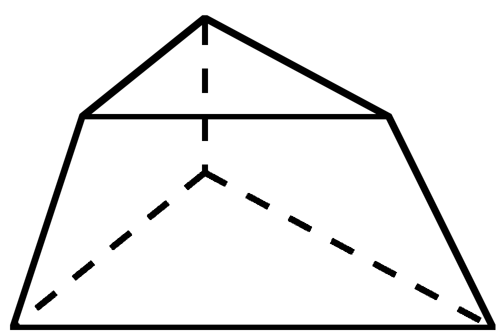
     - the common edge *AB* between 2 adjacent triangles T_1 and T_2 is such that the approximation of the surface formed by them is not within a specified tolerance



     - in this case, remove the common edge between them and replace it and triangles T_1 and T_2 by 4 new triangles with vertices that are the remaining vertices of T_1 and T_2 (i.e., M_1 and M_2) and a new point Mn
     - 4 new triangles $M_1\ A\ M_n$, $M_2\ A\ M_n$, $M_1\ B\ M_n$, and $M_2\ B\ M_n$ where M_n satisfies
       a. being on the surface
       b. it is on the plane that bisects the 2 triangles that are being replaced
       c. of all points satisfying a and b, it is the furthest point from the removed edge

3D PRISM TREE: APPROXIMATION

- Can be represented as a tree structure
  1. 2 sons at initial level
  2. remaining levels have 3 sons unless an edge was removed in which case there are 2 sons
- Approximate each boundary segment by a 5-sided polyhedron in the shape of a truncated tetrahedron (misnamed a prism)
  1. one side is triangular base *T* of tetrahedron
  2. one side is a plane parallel to *T* passing through the point on the surface at the maximum distance from *T*
  3. remaining 3 sides are formed by bisecting planes of *T* and the triangles adjacent at the edges of *T*

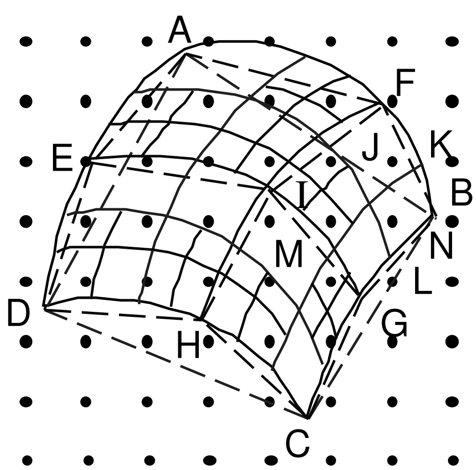


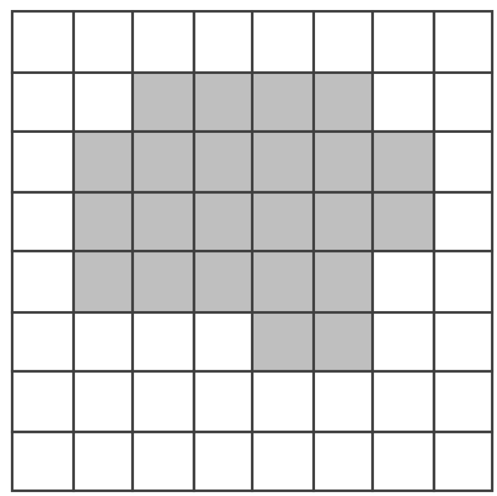
COMPARISON OF PRISM AND STRIP TREES

- Approximation of boundary segments
  1. strip tree uses rectangles while the prism tree uses trapezoids (truncated tetrahedra)
  2. for convex objects the interiors of the trapezoids are disjoint while adjacent ones have common edges (faces)
  3. disjointness and common boundaries rarely arise in a strip tree
- Shape of the approximating trapezoid for a boundary segment *S*
  1. depends on the line (triangle) that approximates the segments adjacent to *S* as well as on the level of the tree at which they are found
  2. shape of rectangles used in the strip tree are independent of the approximations of the neighboring segments and of their level

DISPLAYING CURVED SURFACES (CATMULL)

- Can represent by a collection of parametric bicubic surface patches
- Recursively decompse each patch into subpatches until the subpatches are so small that they only span the center of one pixel (or can be shown to liie outside the display region)
- Test for number of pixel centers spanned by a patch is based on an approximation by a polygon connecting the patch’s corners





quadrilateral ABCD approximates patch ABCD

decompose ABCD into four quadrilateral patches

subdivide further but no distinction between subpatches and their quadrilateral approximations

each subpatch spans at most one pixel center

raster image resulting from the decomposition

MECHANICS OF CURVED SURFACE DISPLAY

- Processing follows quadtree decomposition paradigm in parameter space—does not generate the quadtree explicitly
- Several patches can span the same pixel center as they are in three-dimensional space and thus use z-buffer to keep track of patch that is closest to the viewpoint
- Can use with other patch representations
  1. characteristic polyhedra (Bezier and B-spline patches)
  2. allow approximation by convex hull enclosing its control points
     - more accurate than Catmull's method which is based only on the four corners of the patch