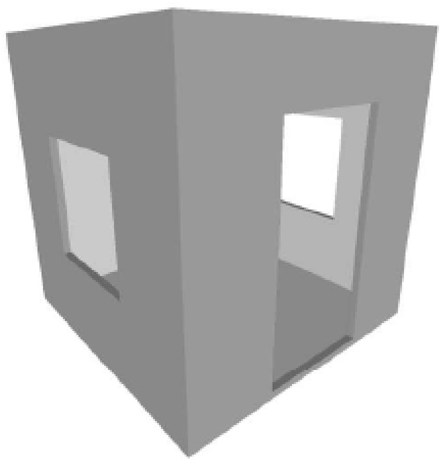
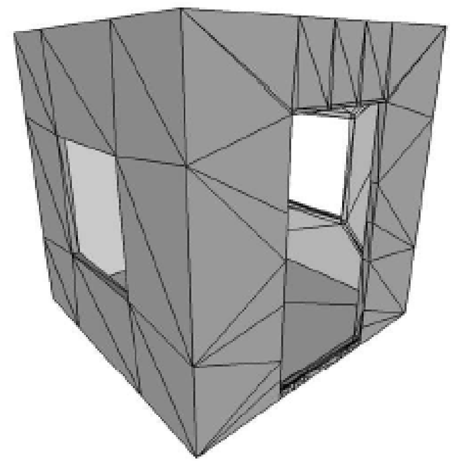
Dual Marching Cubes

Scott Schaefer and Joe Warren

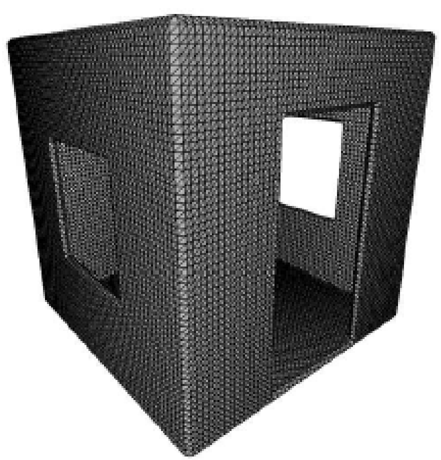
Perform marching cubes
over a sparse dual grid

Goals: Capture thin features & Use fewer triangles.

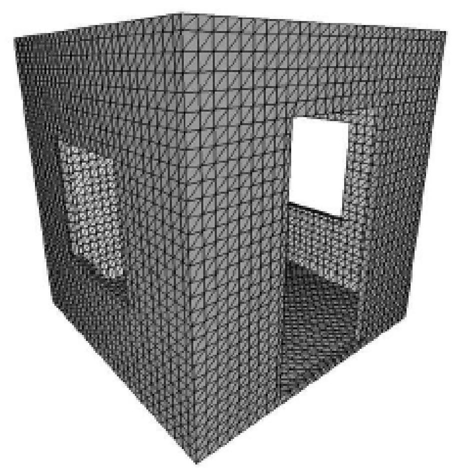




Dual Marching Cubes (440 tris)

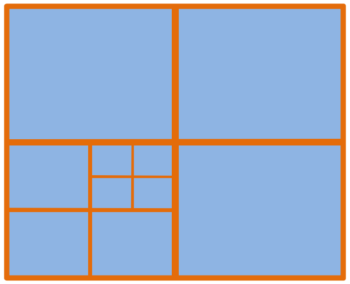


Marching Cubes (67k tris)

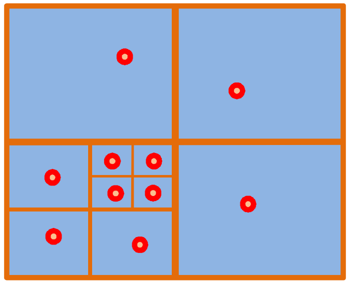


Dual Contouring (17k tris)

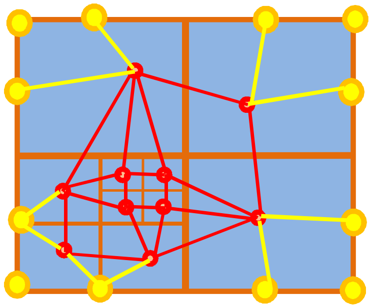
Process Overview:



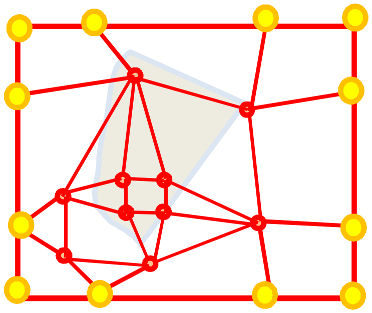
1. Octree defines resolution



2. Grid vertex placed per octree cell at features of signed distance function

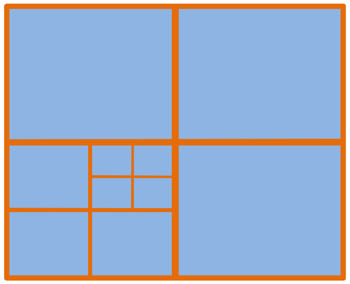


3. Dual grid edges and faces are found

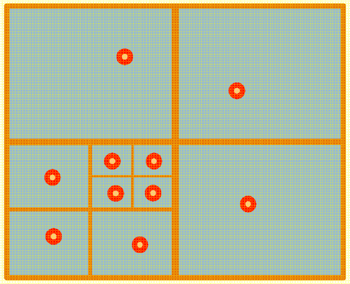


4. Perform marching cubes over dual grid

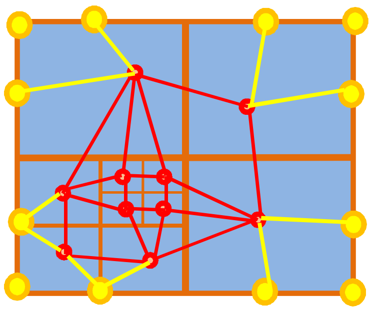
Process Overview:



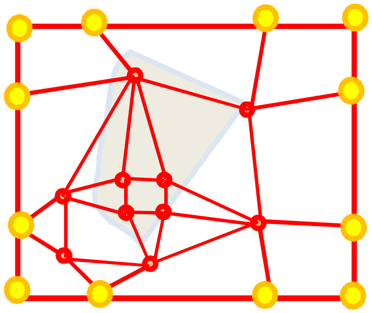
1. Octree defines resolution



2. Grid vertex placed per octree cell at features of signed distance function

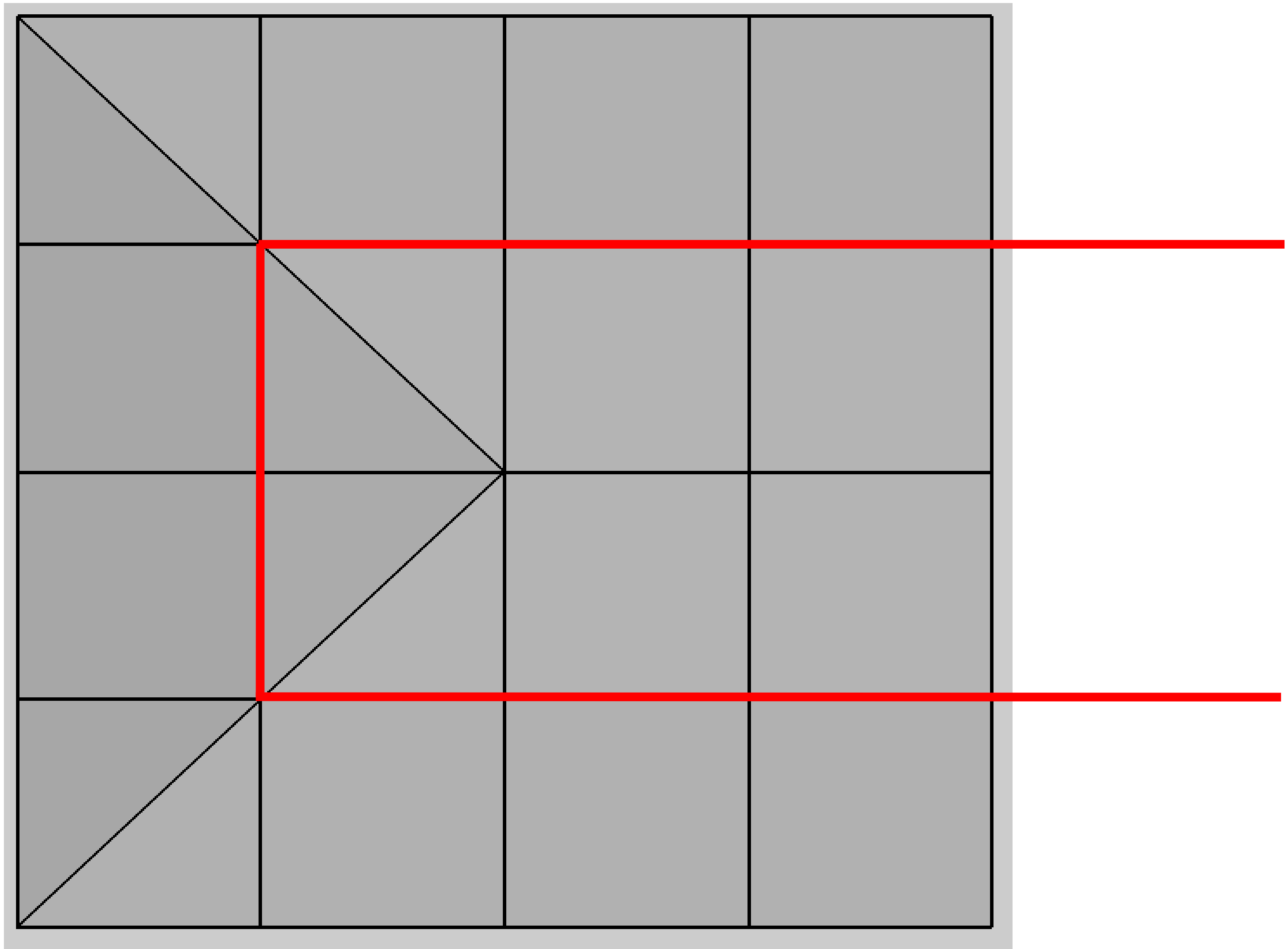


3. Dual grid edges and faces are found



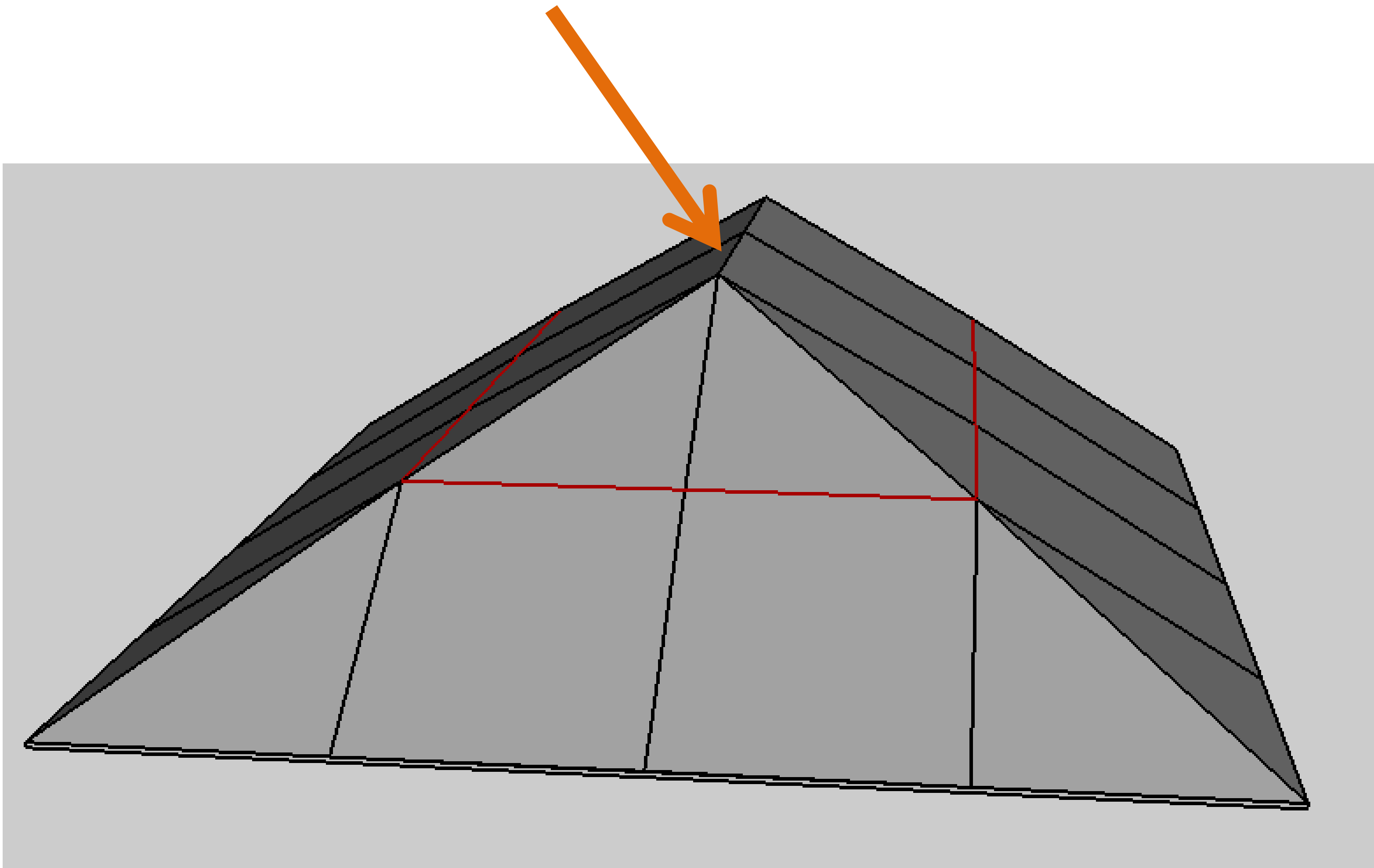
4. Perform marching cubes over dual grid

What are features of a signed distance function?



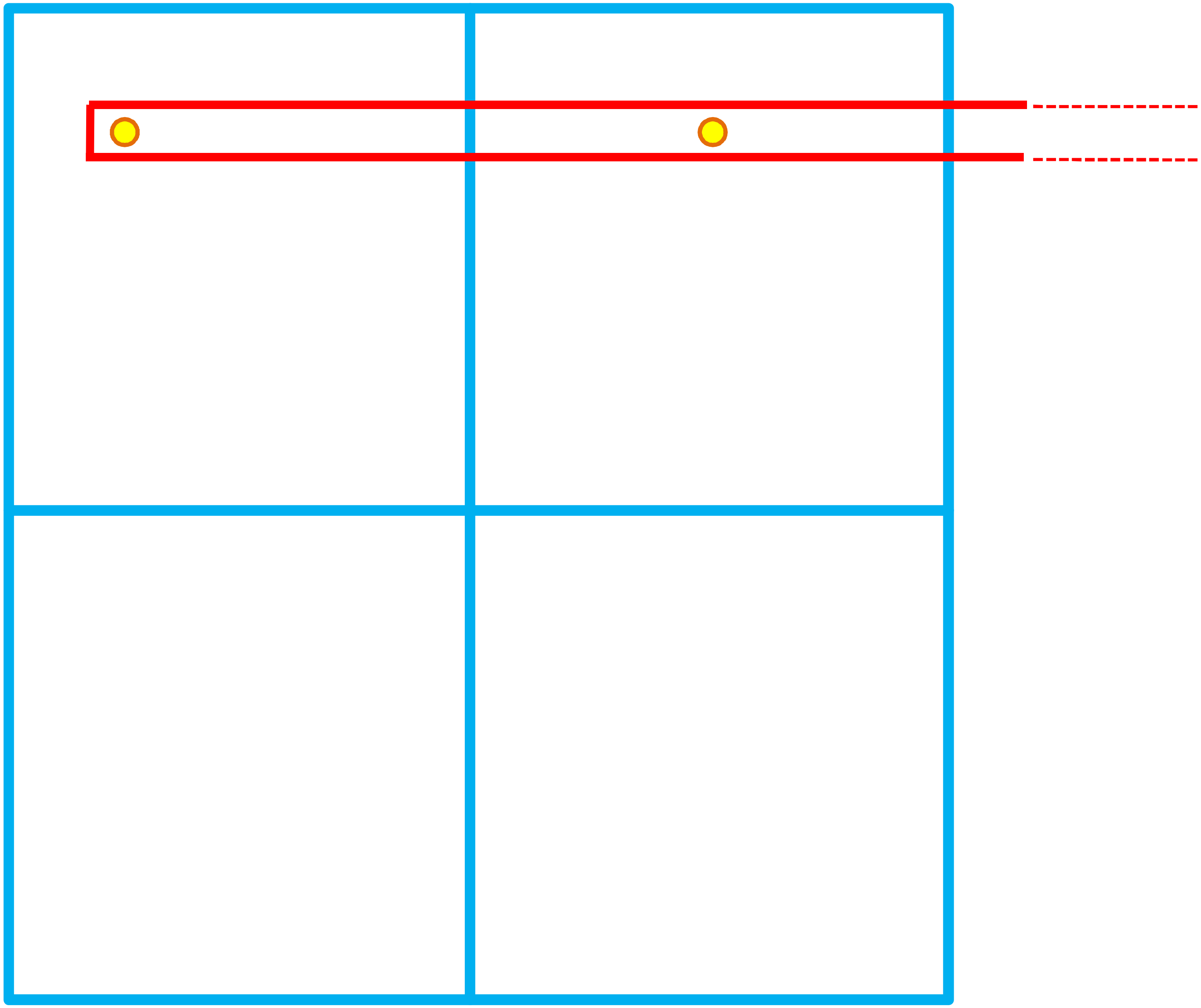
Example: Red line = isosurface

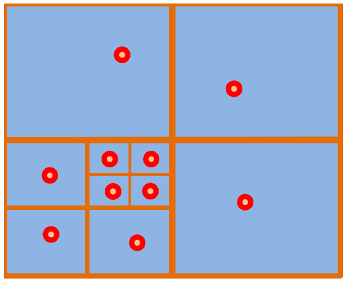
Features go where planes collide



Example: Grey shape = distance function

Thin shape will 'grab' vertices

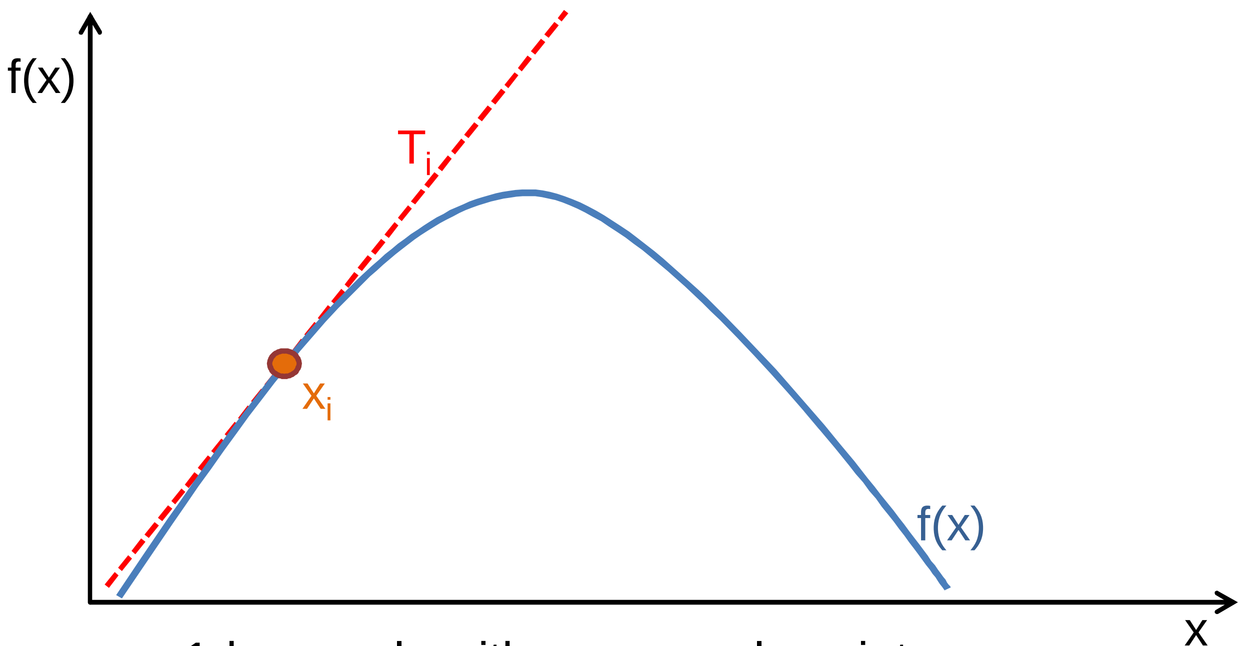




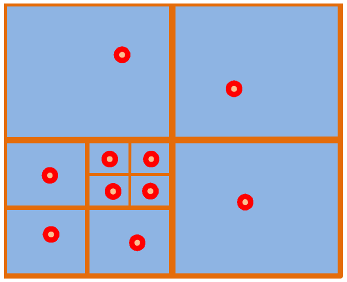
Process to place vertices

1. Approximate surface at sample points with planes

$$T_i(x, y, z) = \nabla f(x_i, y_i, z_i) \cdot ((x, y, z) - (x_i, y_i, z_i))$$



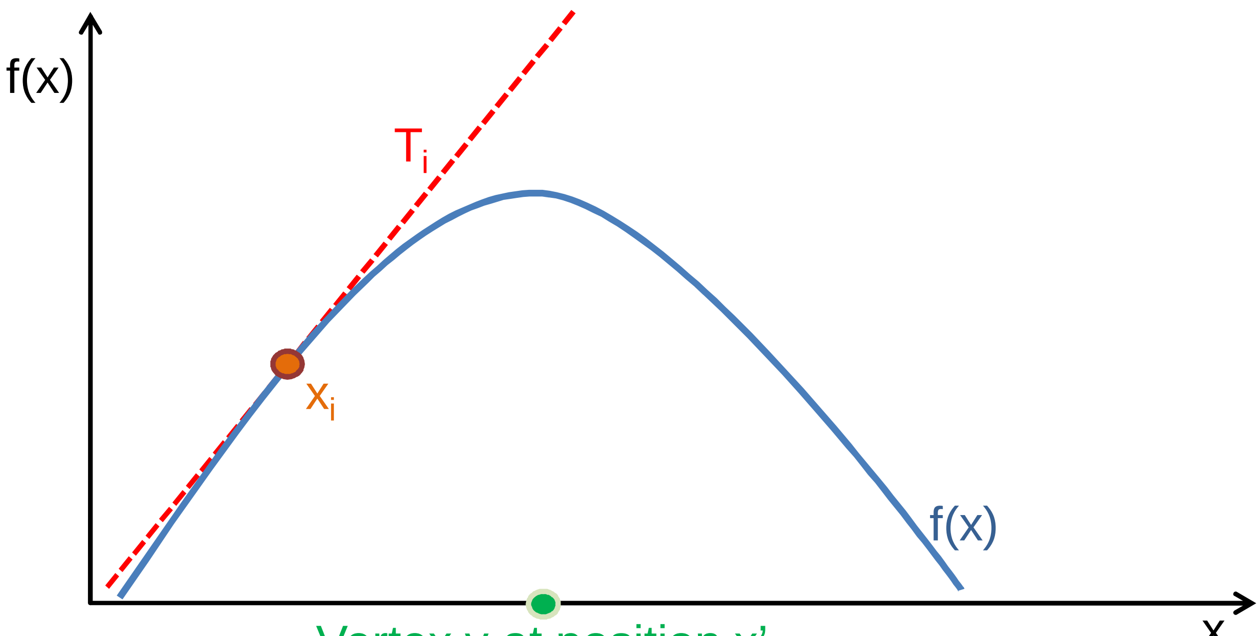
1d example with one sample point

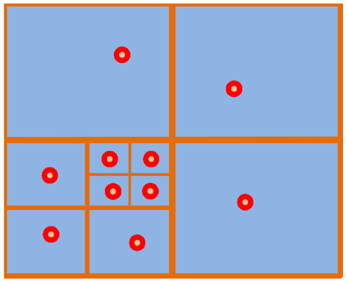


Process to place vertices

2. Find vertex position to minimize the Error Quadric

$$E(w,x,y,z) = \sum_i \frac{(w - T_i(x,y,z))^2}{1 + |\nabla f(x_i,y_i,z_i)|^2}$$

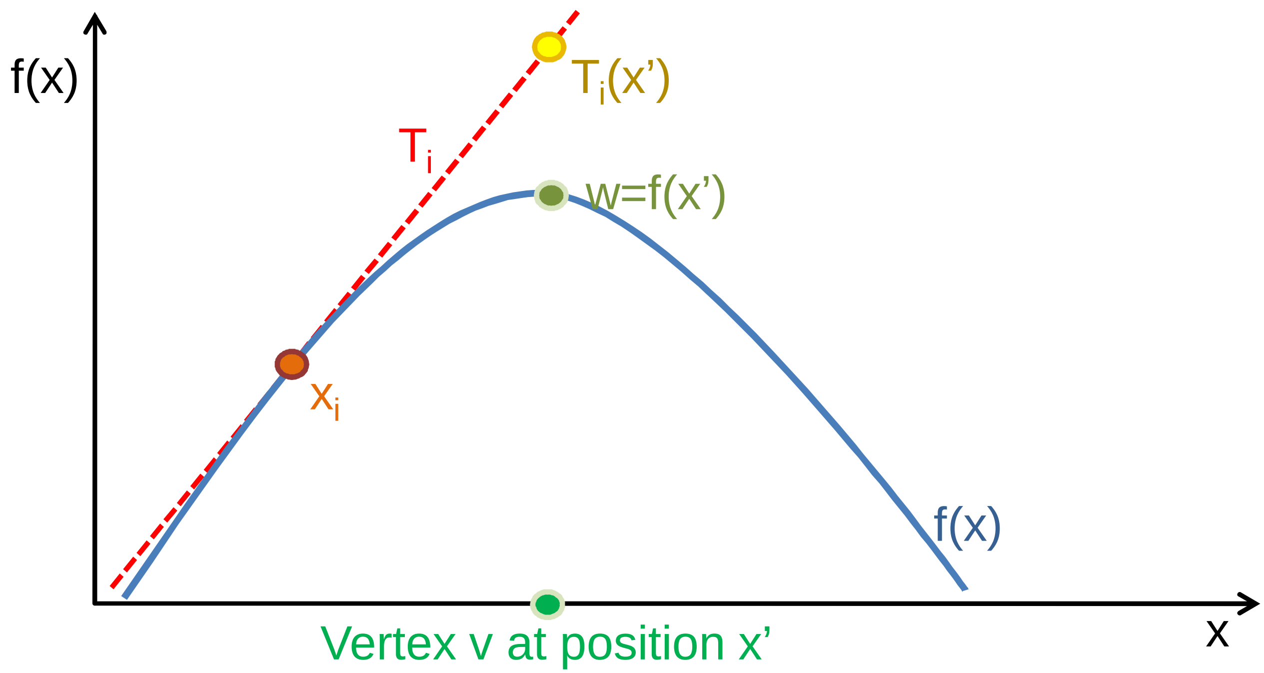


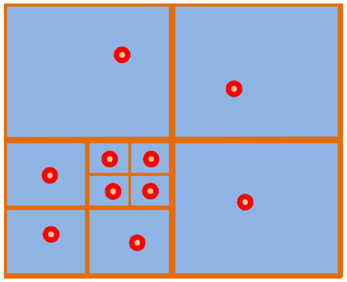


Actual value of f(x,y,z)

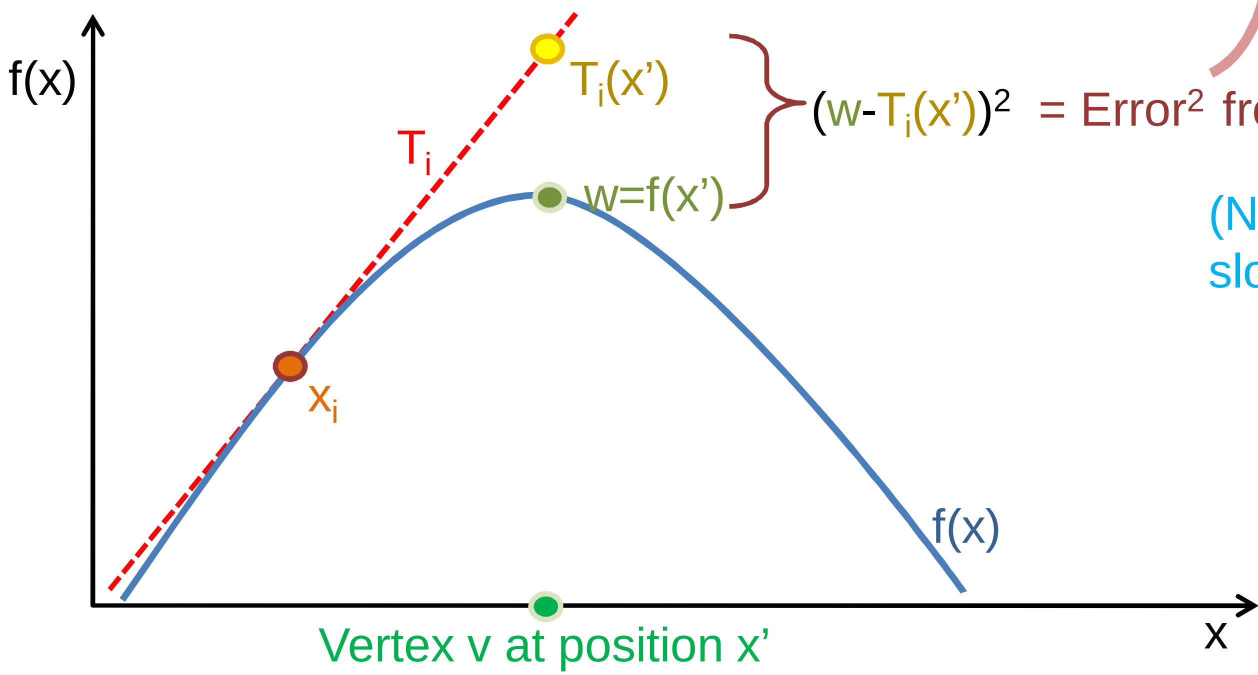
Value predicted from i^{th} sample

$$E(w, x, y, z) = \sum_i \frac{(w - T_i(x, y, z))^2}{1 + |\nabla f(x_i, y_i, z_i)|^2}$$



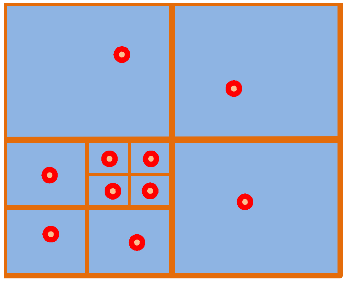


$$E(w,x,y,z) = \sum_i \frac{(w - T_i(x,y,z))^2}{1 + |\nabla f(x_i,y_i,z_i)|^2}$$

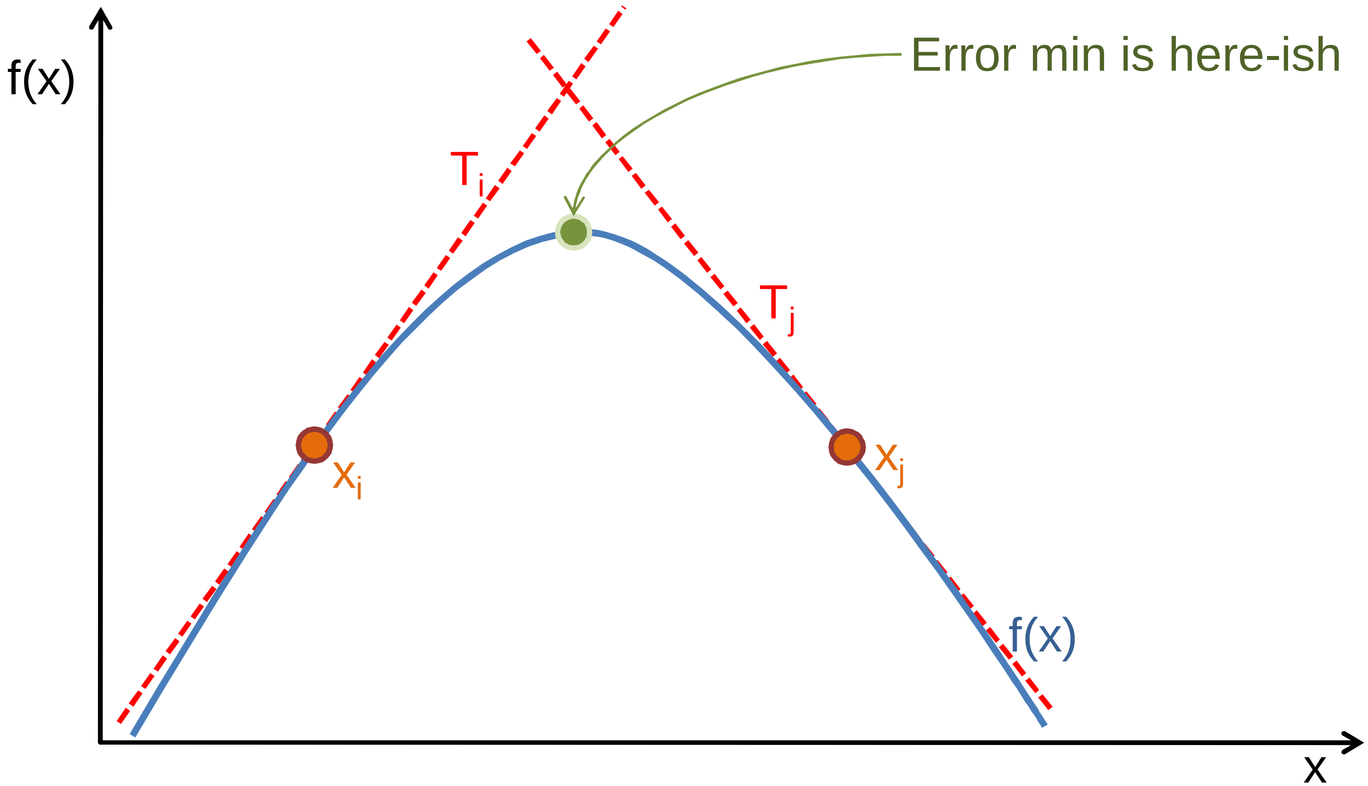


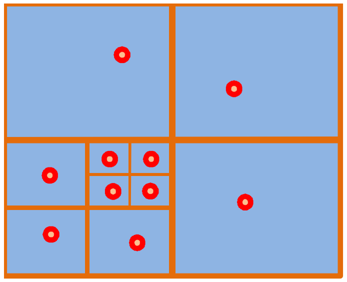
$(w-T_i(x'))^2$ = Error^2 from this sample

(Normalize by slope of T_i)

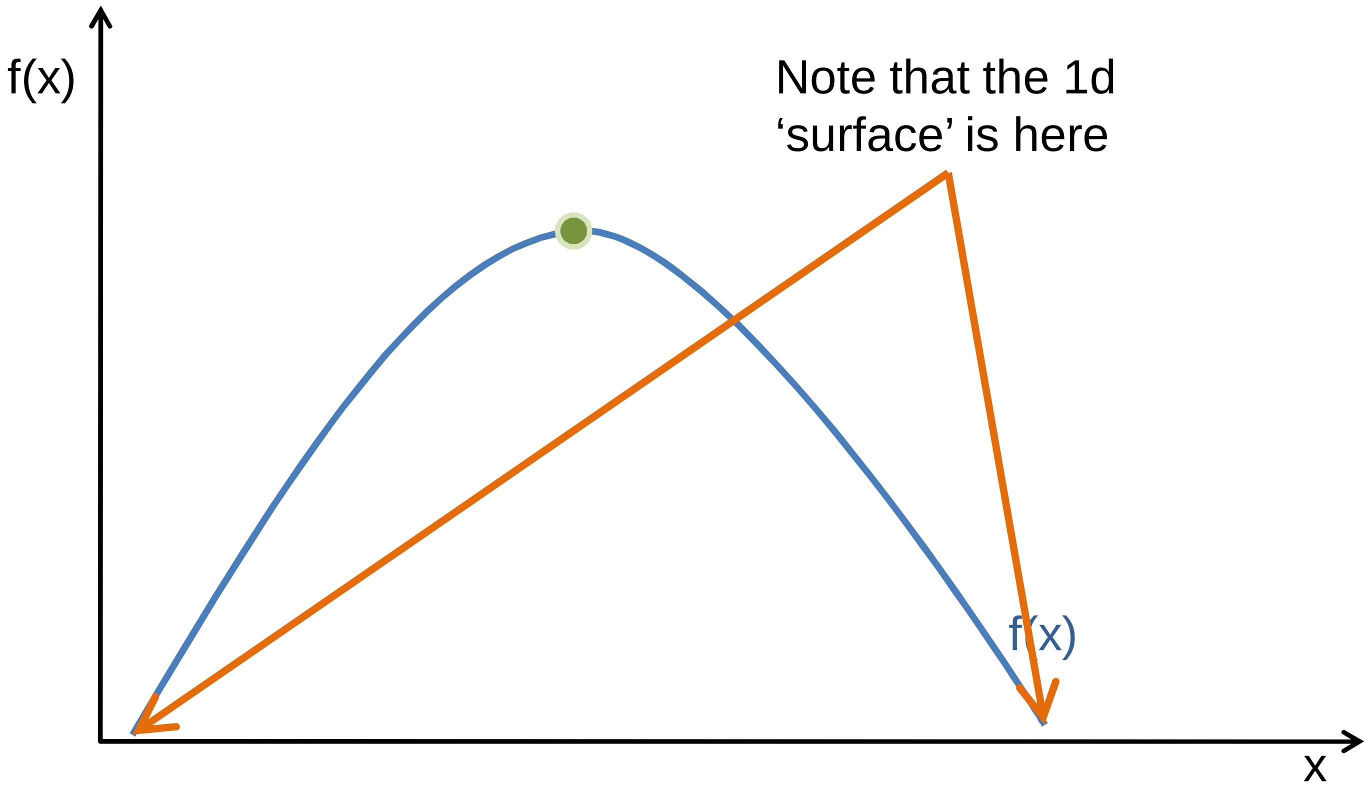


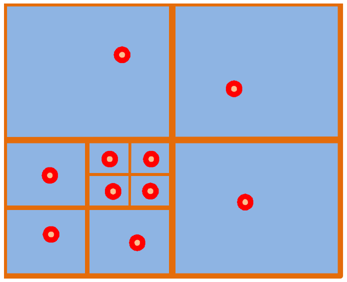
$$E(w,x,y,z) = \sum_i \frac{(w - T_i(x,y,z))^2}{1 + |\nabla f(x_i,y_i,z_i)|^2}$$



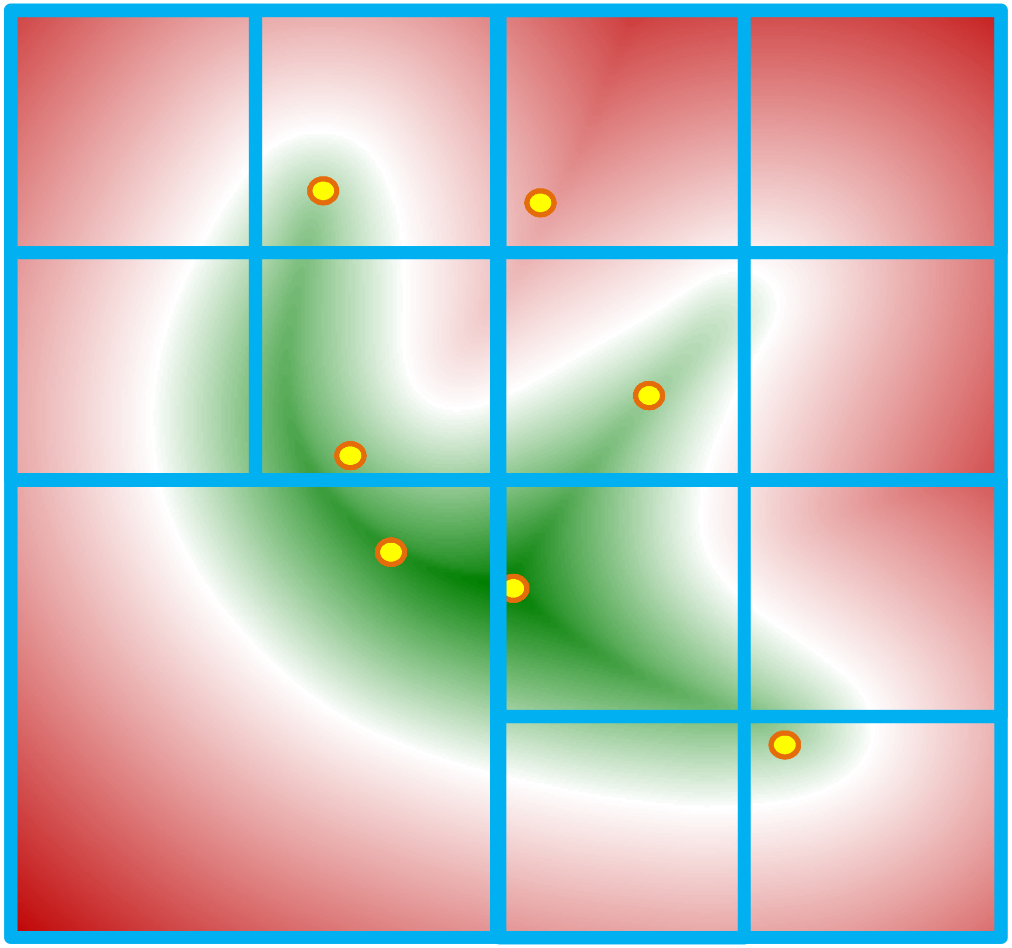


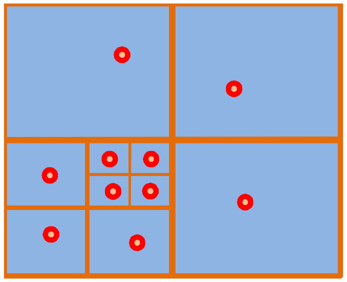
Features of distance field are not features of the surface!

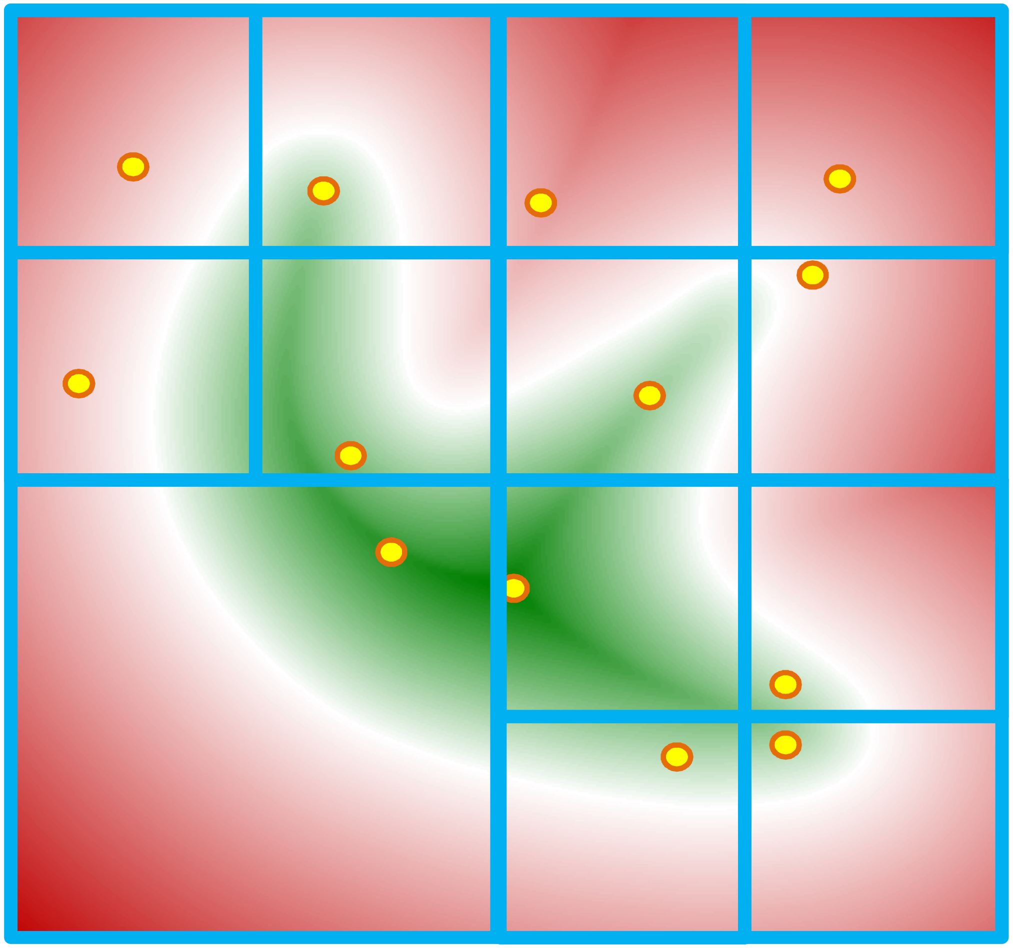




Features are often near medial axis



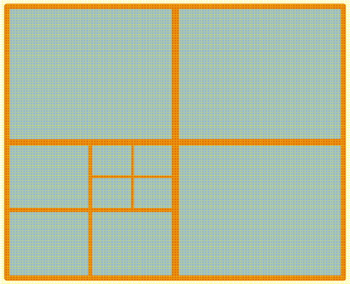




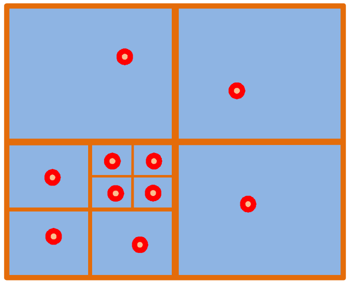
Result of minimization: 1 vertex per cell

(make sure vertex is in its own cell!)

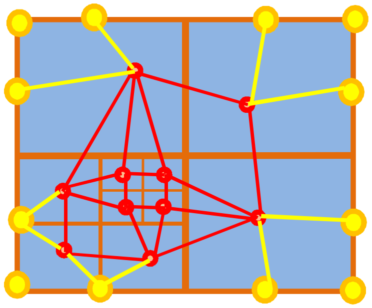
Process Overview:



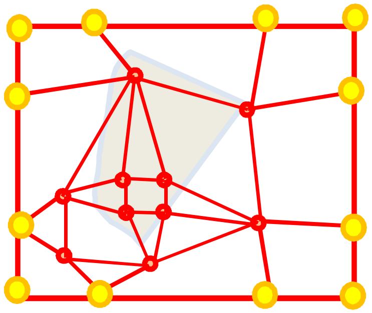
1. Octree defines resolution



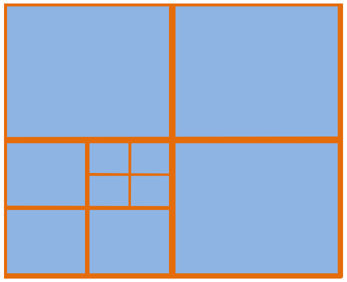
2. Grid vertex placed per octree cell at features of signed distance function



3. Dual grid edges and faces are found



4. Perform marching cubes over dual grid



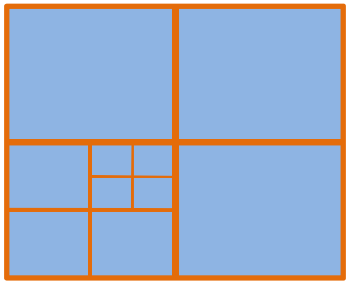
Octree Generation

```
Generate(Node):
    List samples = Sample(Node)
    if (Error(samples) > thresh):
          Generate(Children(Node))
```

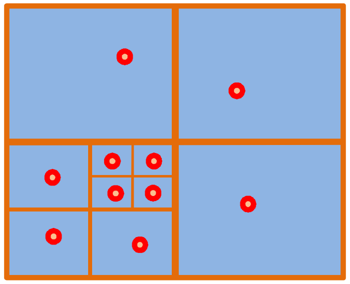
`Sample(Node):` Fine regular sampling? Random sampling?

`Error(samples):` Error Quadric

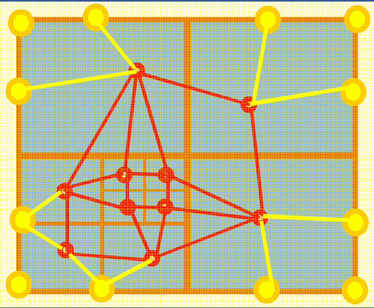
Process Overview:



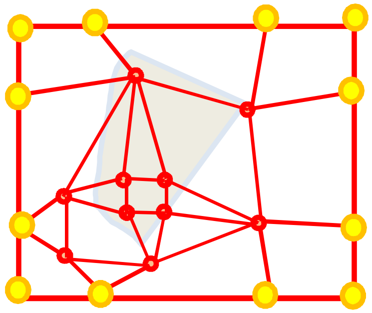
1. Octree defines resolution



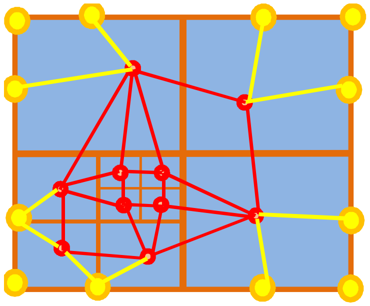
2. Grid vertex placed per octree cell at features of signed distance function



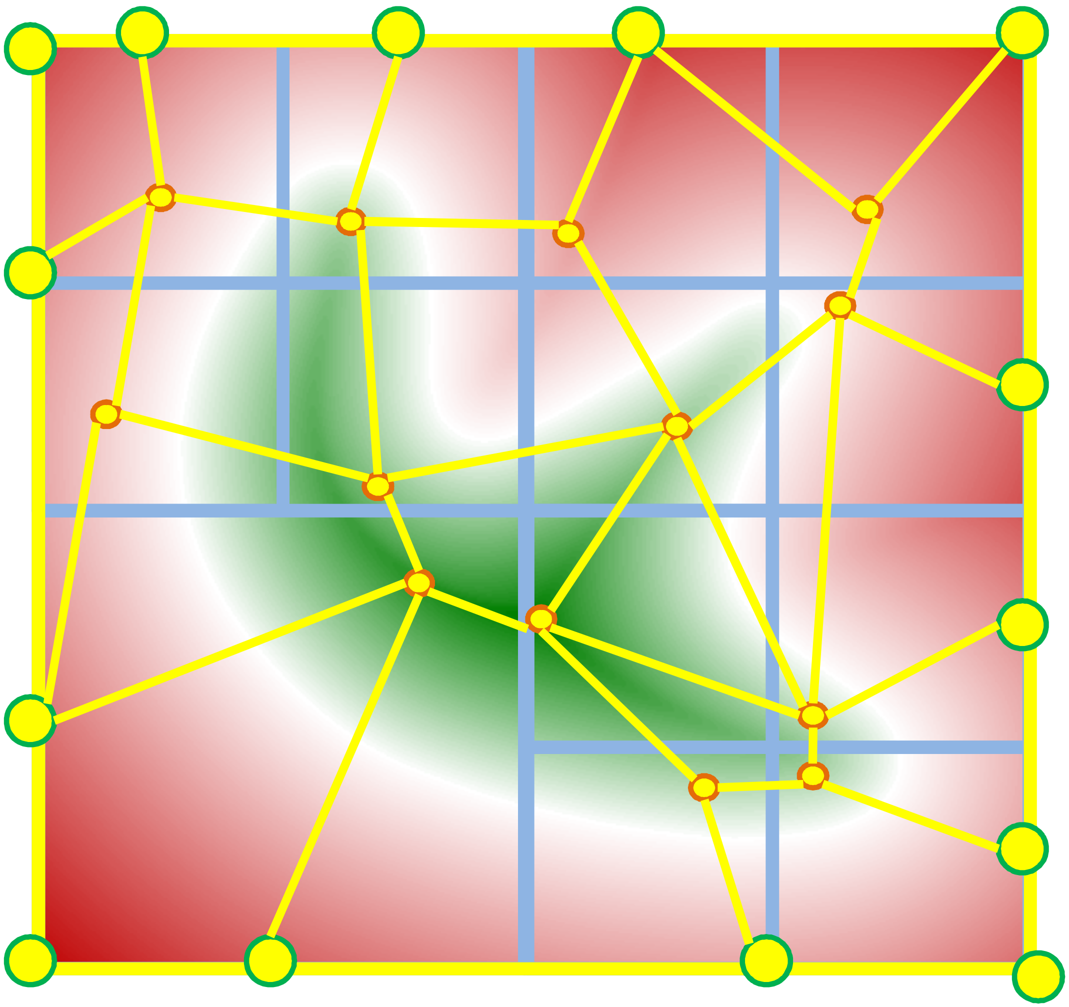
3. Dual grid edges and faces are found

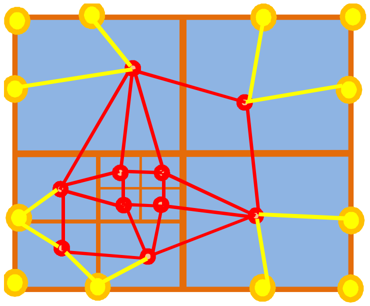


4. Perform marching cubes over dual grid



Dual grid looks like this:

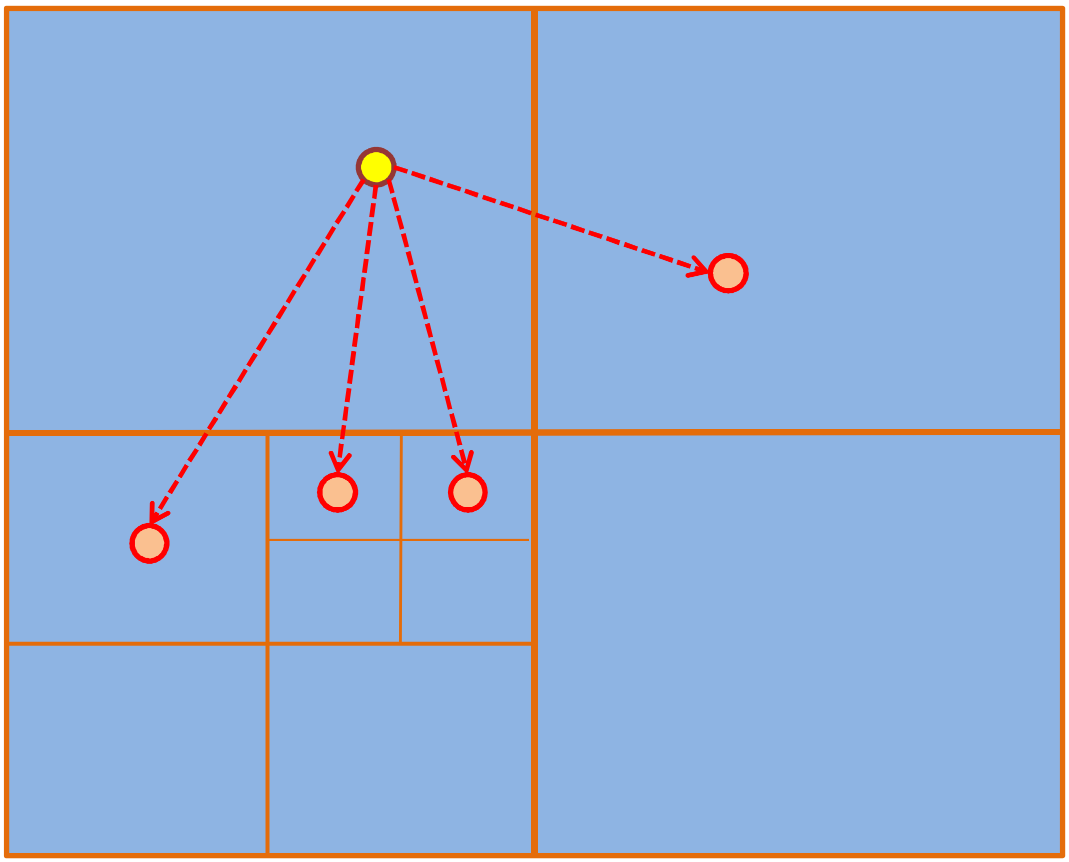


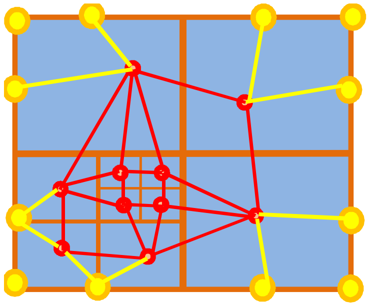


dual grid:

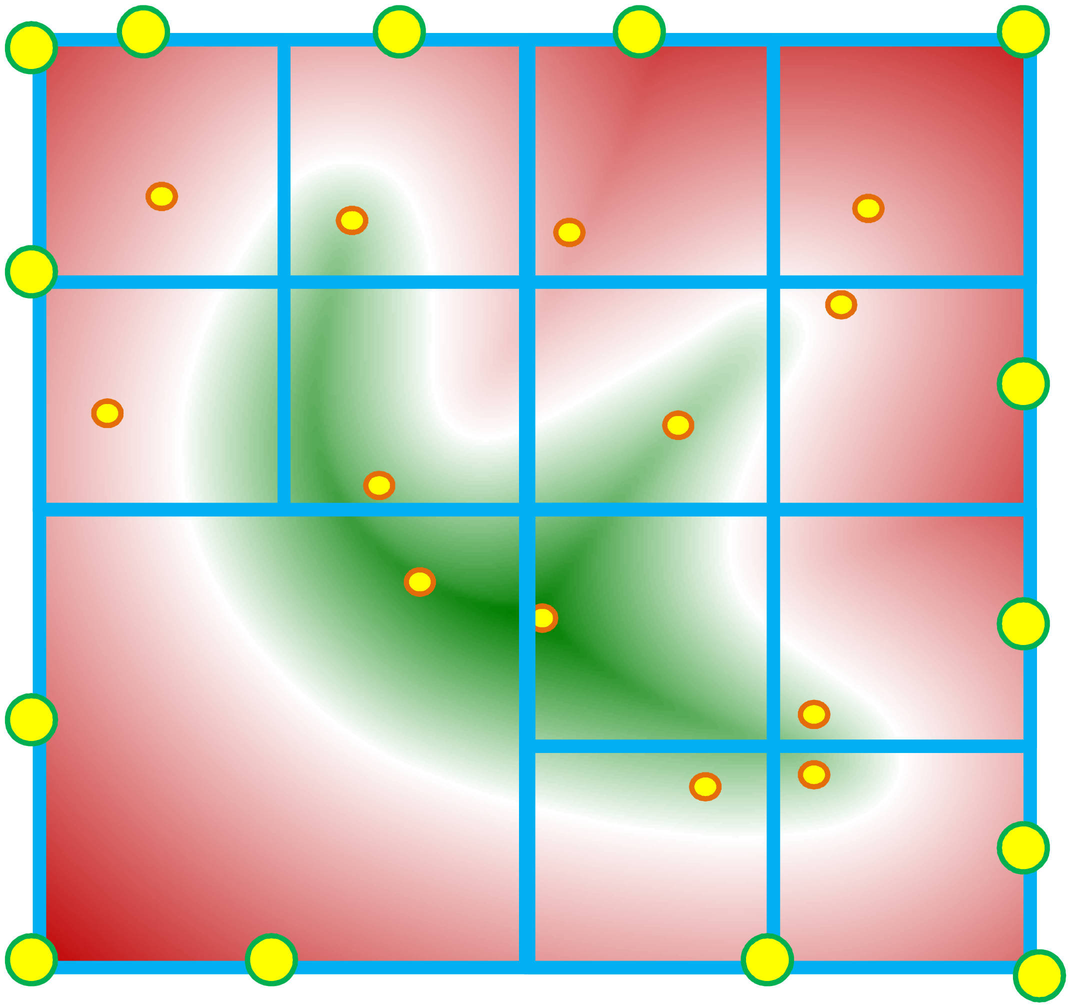
Faces become Vertices

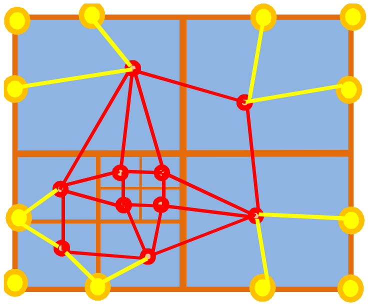
Connect Adjacent Faces' Vertices



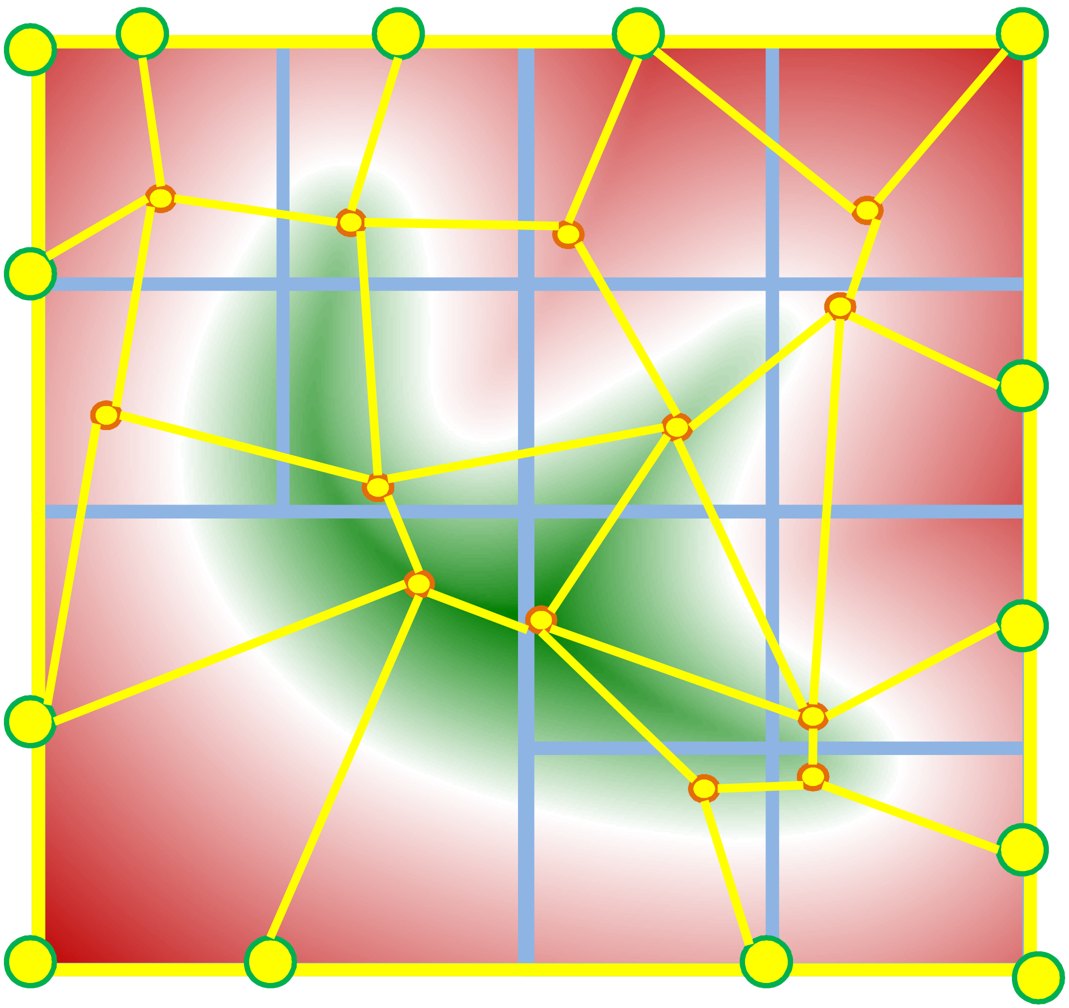


Border vertices are special inserted where needed

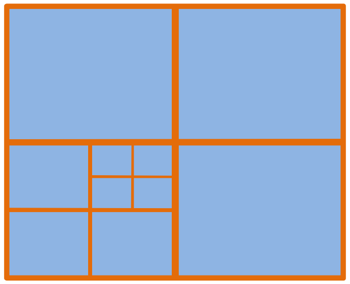




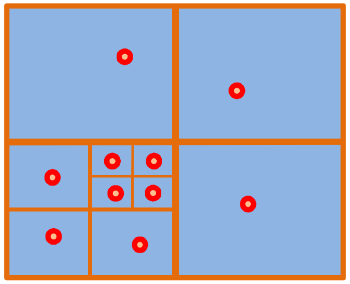
See paper for algorithm



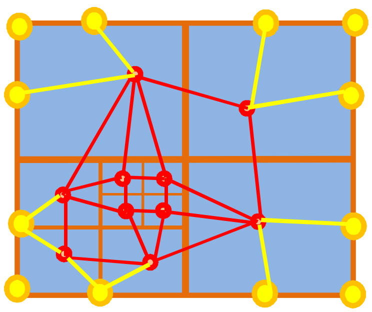
Process Overview:



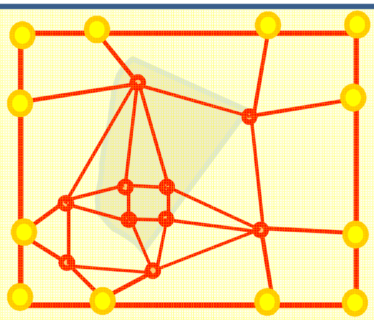
1. Octree defines resolution



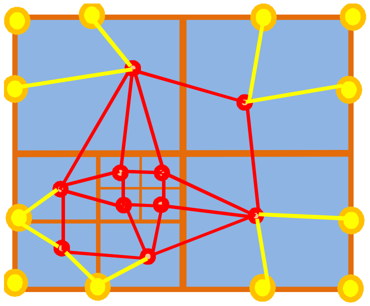
2. Grid vertex placed per octree cell at features of signed distance function



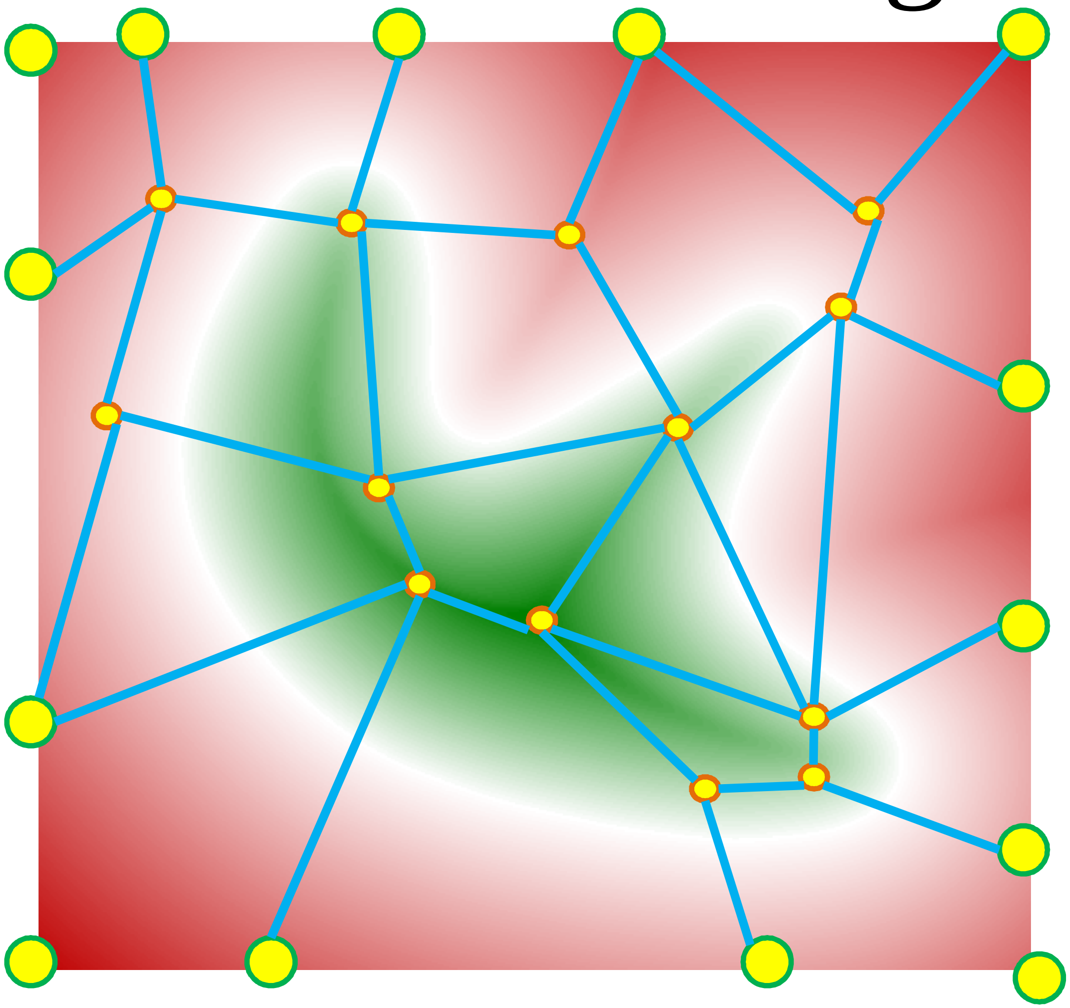
3. Dual grid edges and faces are found

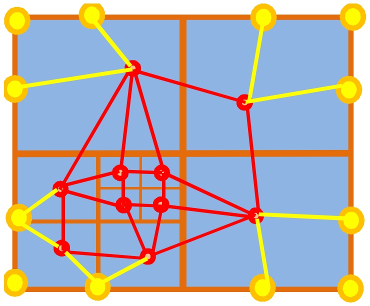


4. Perform marching cubes over dual grid

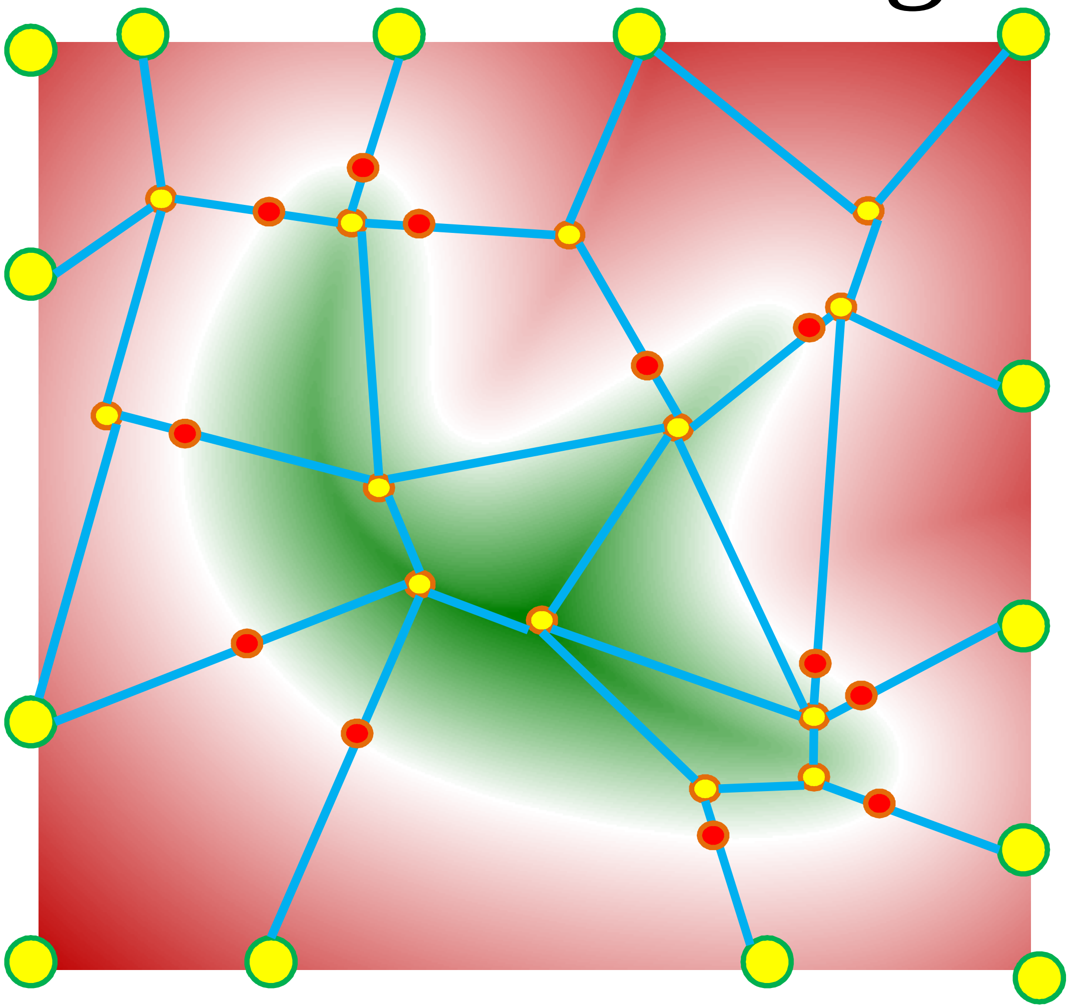


Perform marching cubes

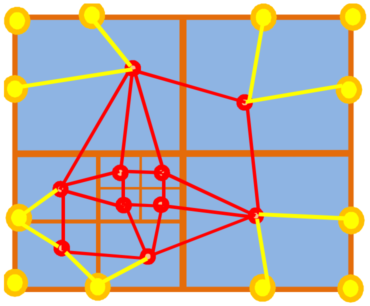




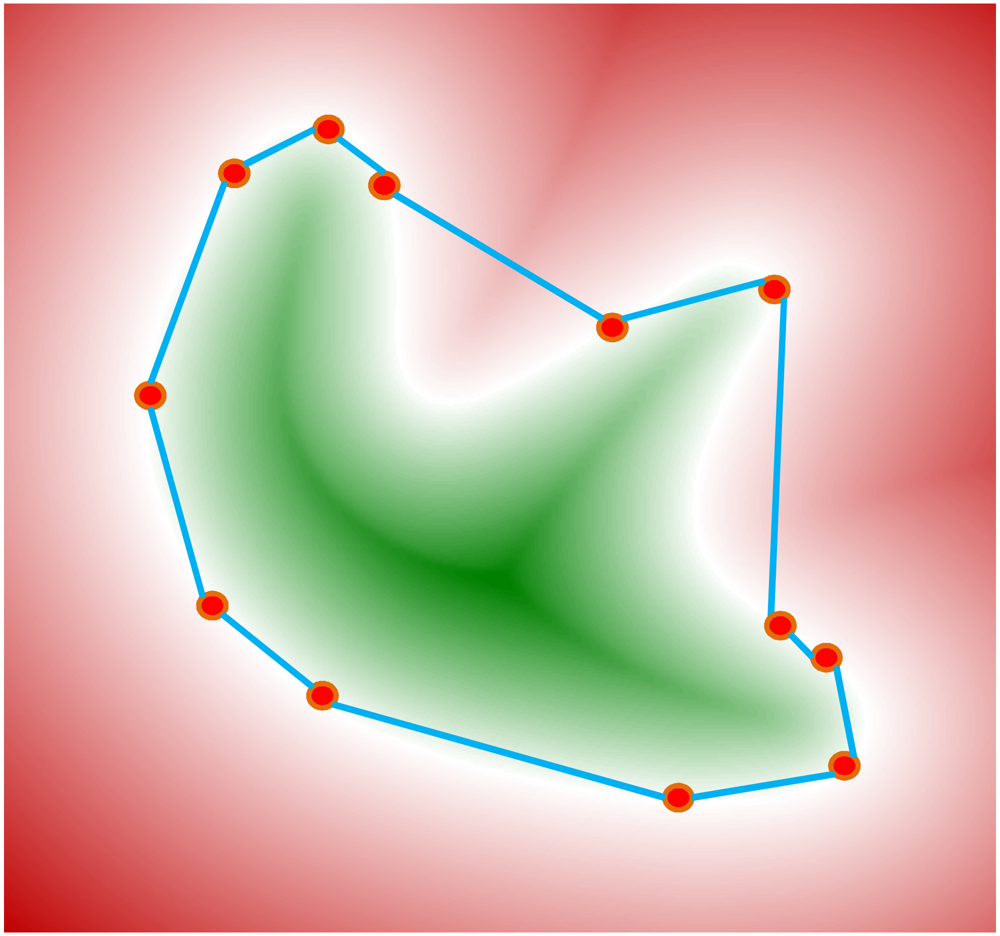
Perform marching cubes



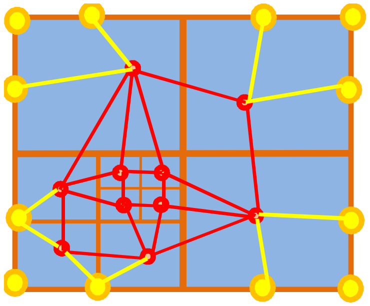
Pretend you have all 8 corners
for every 'cube' in the dual grid



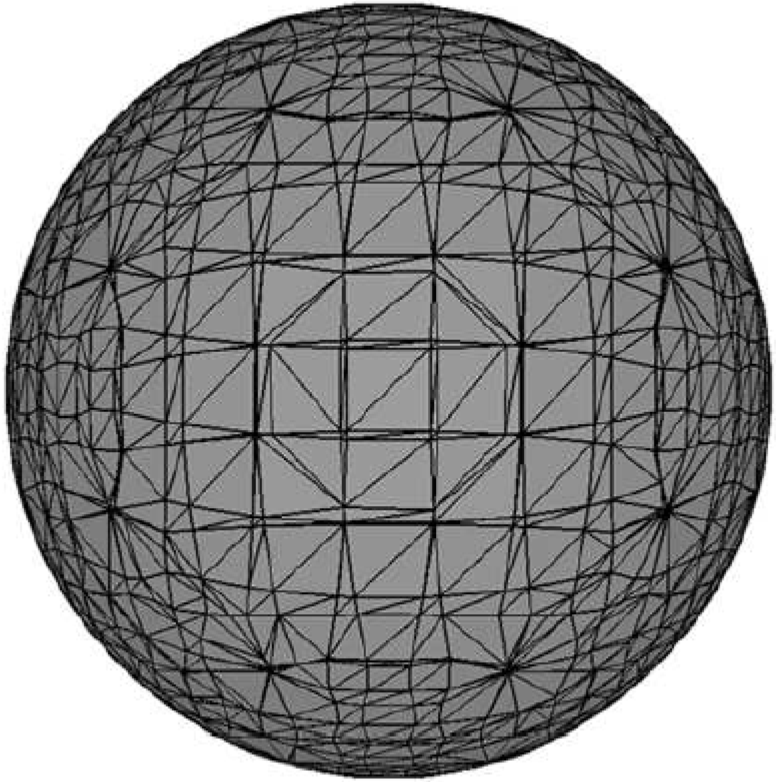
Perform marching cubes

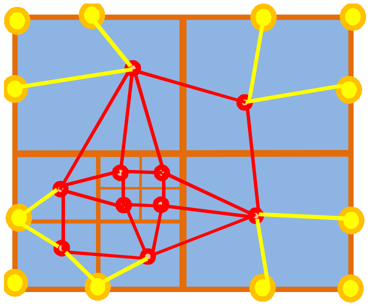


Should have used a finer octree

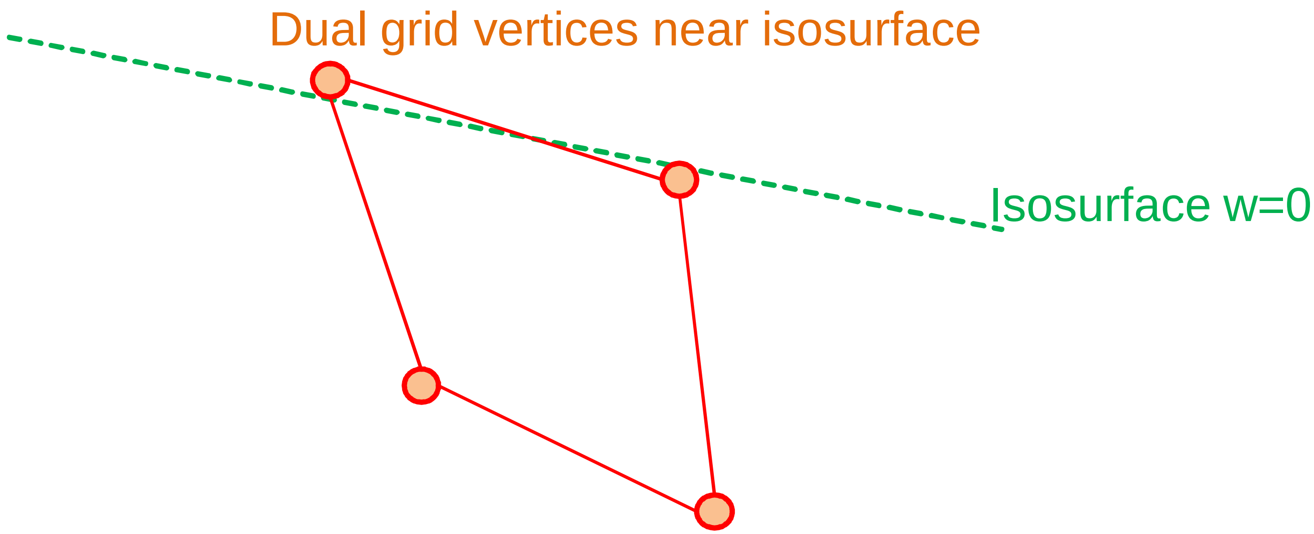


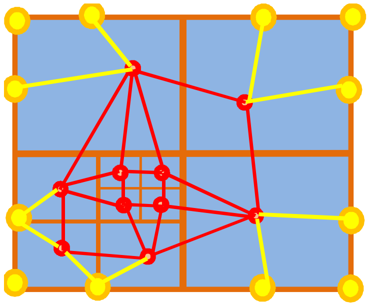
Problem: slivers



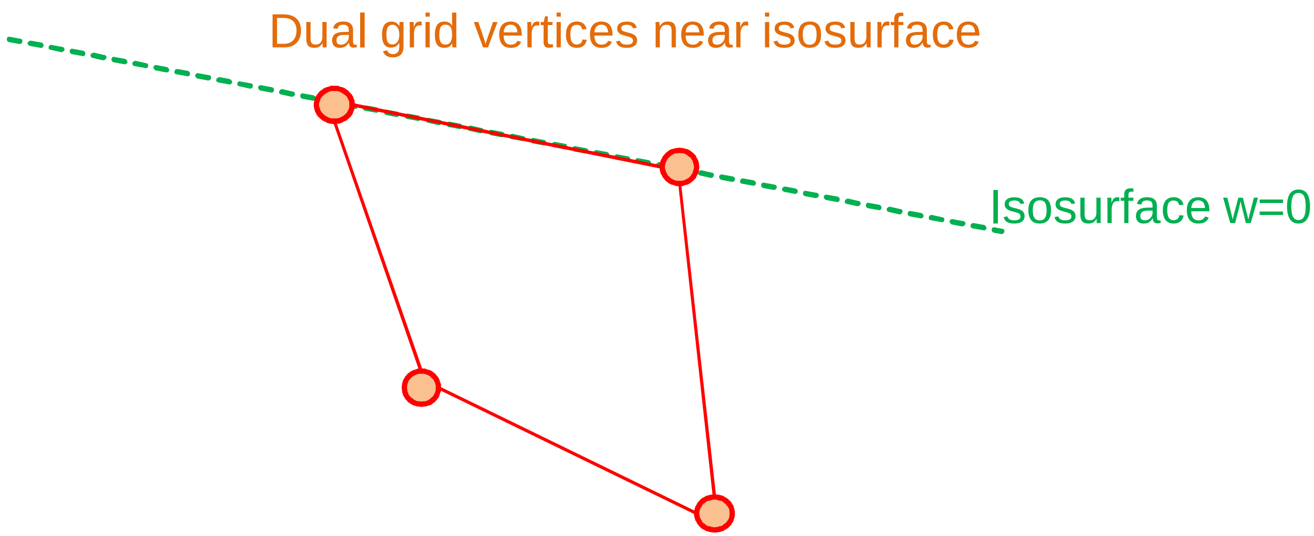


Problem: slivers

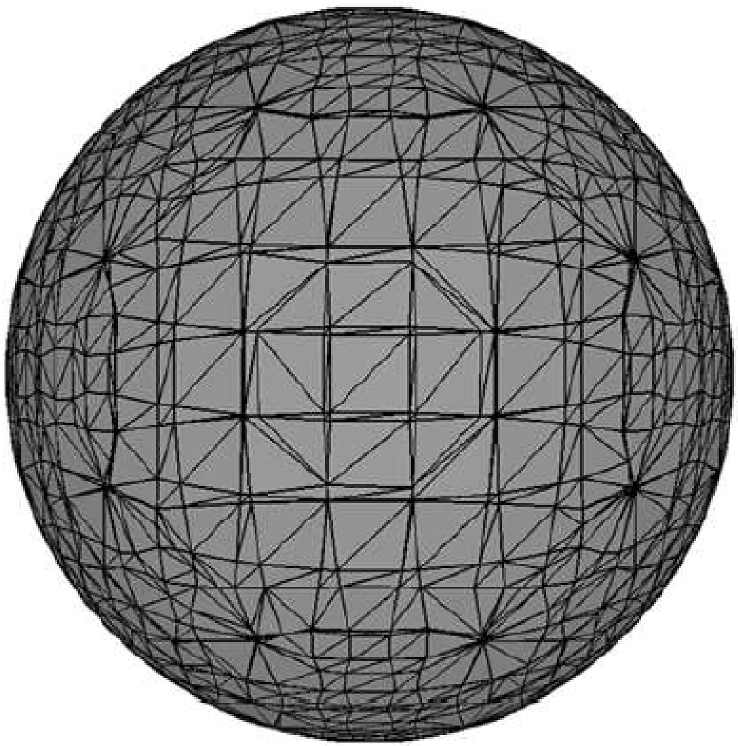




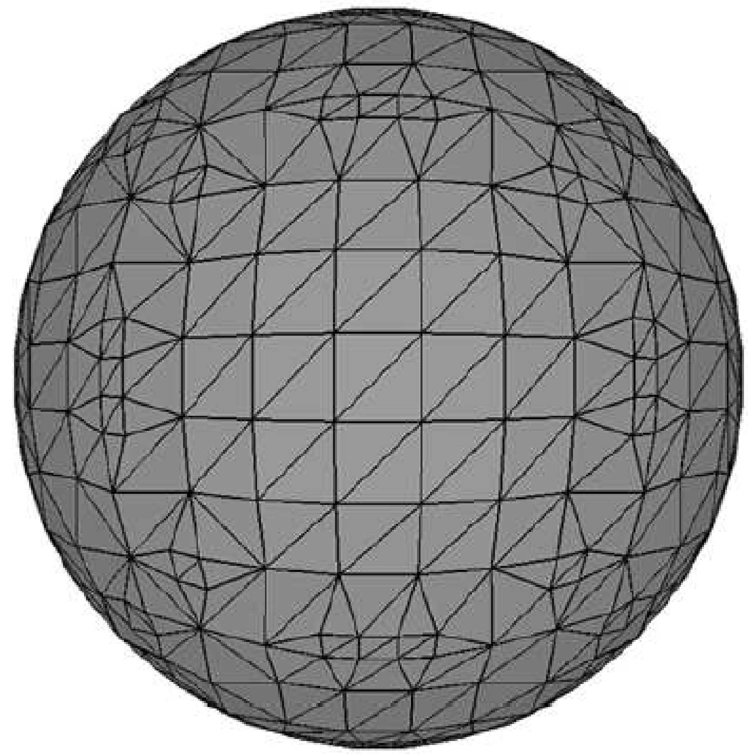
Solution: push vertices to surface



Result of sliver elimination

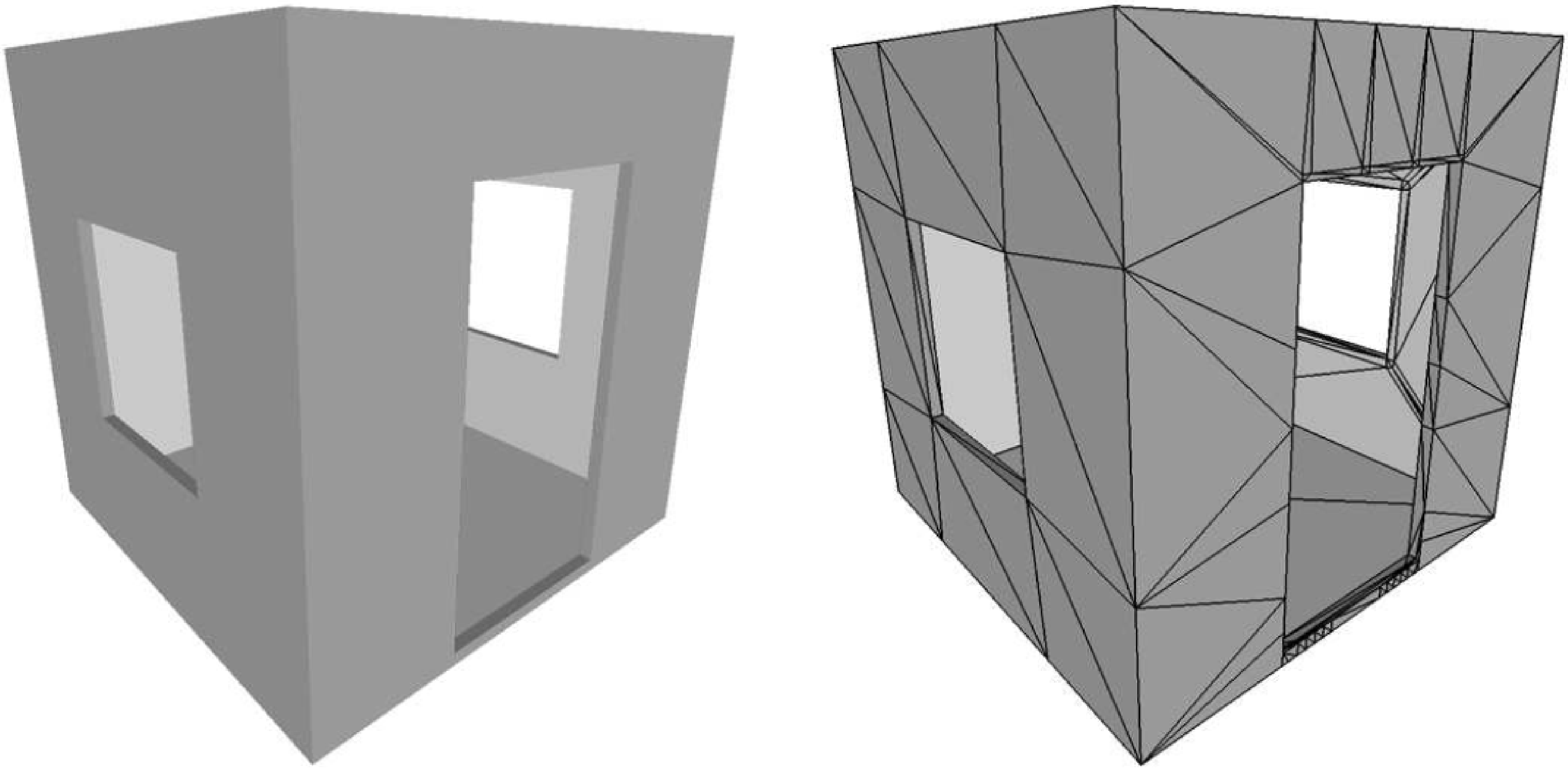


Without sliver elim



With sliver elim

Congratulations, you’re done.

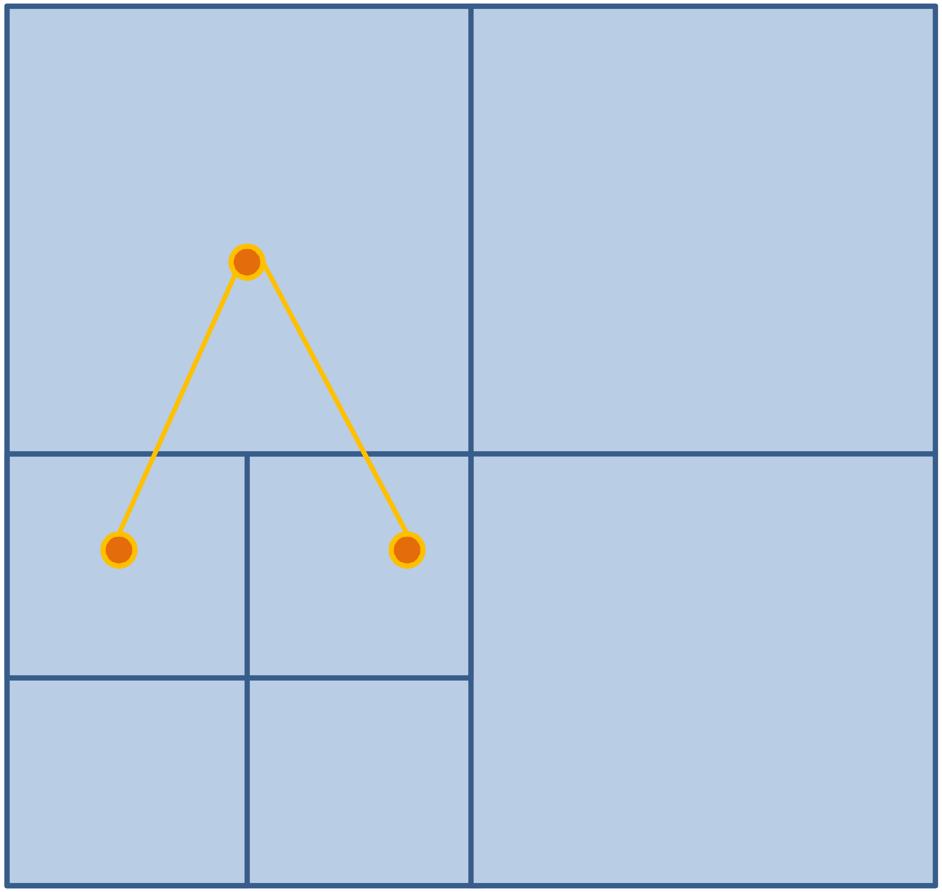


Limitations?

Paper shows thin features work well

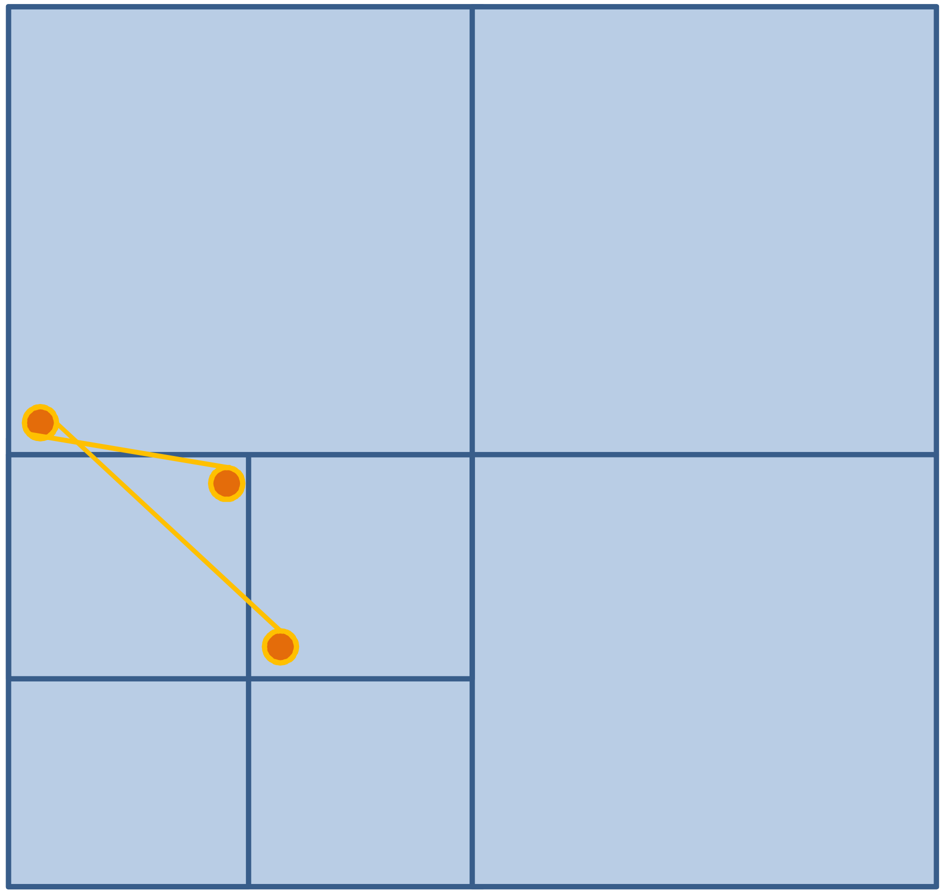
But what about multiple adjacent thin features?

Element inversions?



Can faces of the dual grid become inverted?

Element inversions?



Can faces of the dual grid become inverted?

That's all. Questions?