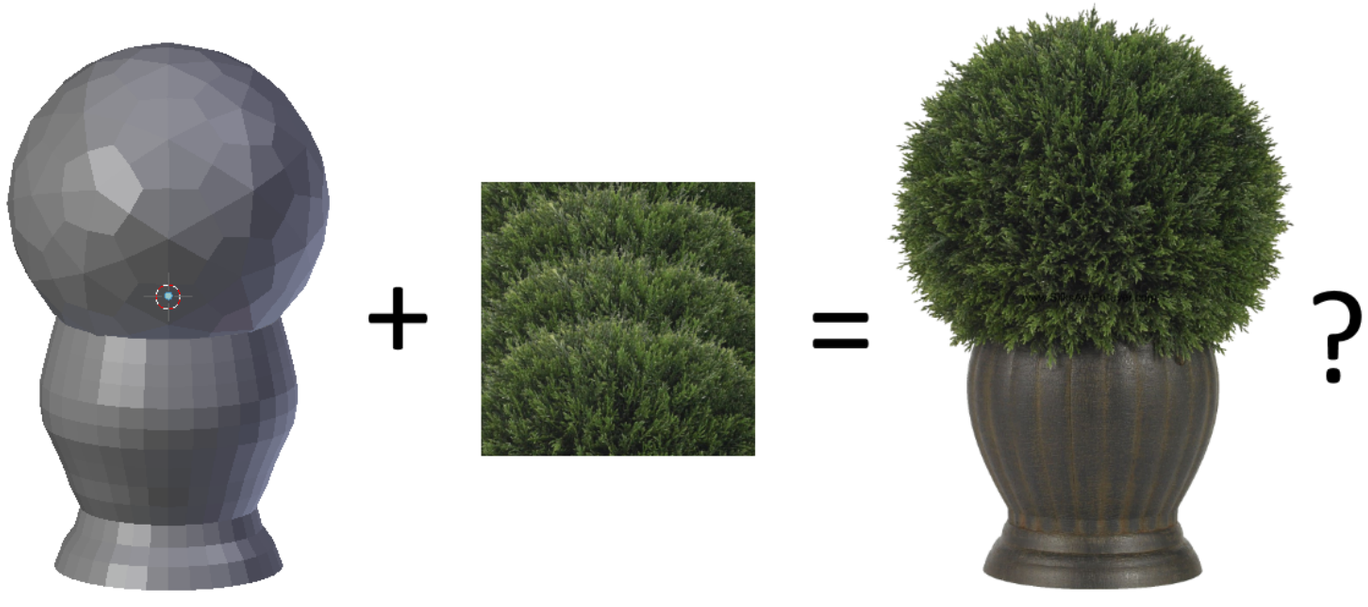
CS 184 Project Milestone

Peter Manohar, **B**rian Lei , **J**ames Fong

Our Project

- Raymarching is a rendering technique that lets us render very complicated implicit geometry (such as fractals) efficiently.
- Raymarching uses a distance field. A distance field gives the distance to the nearest surface for every point in world space.
- Our idea: find a way to paste a “distance field texture” onto surfaces, so that we can locally add detail at little additional cost in the rendering pipeline.

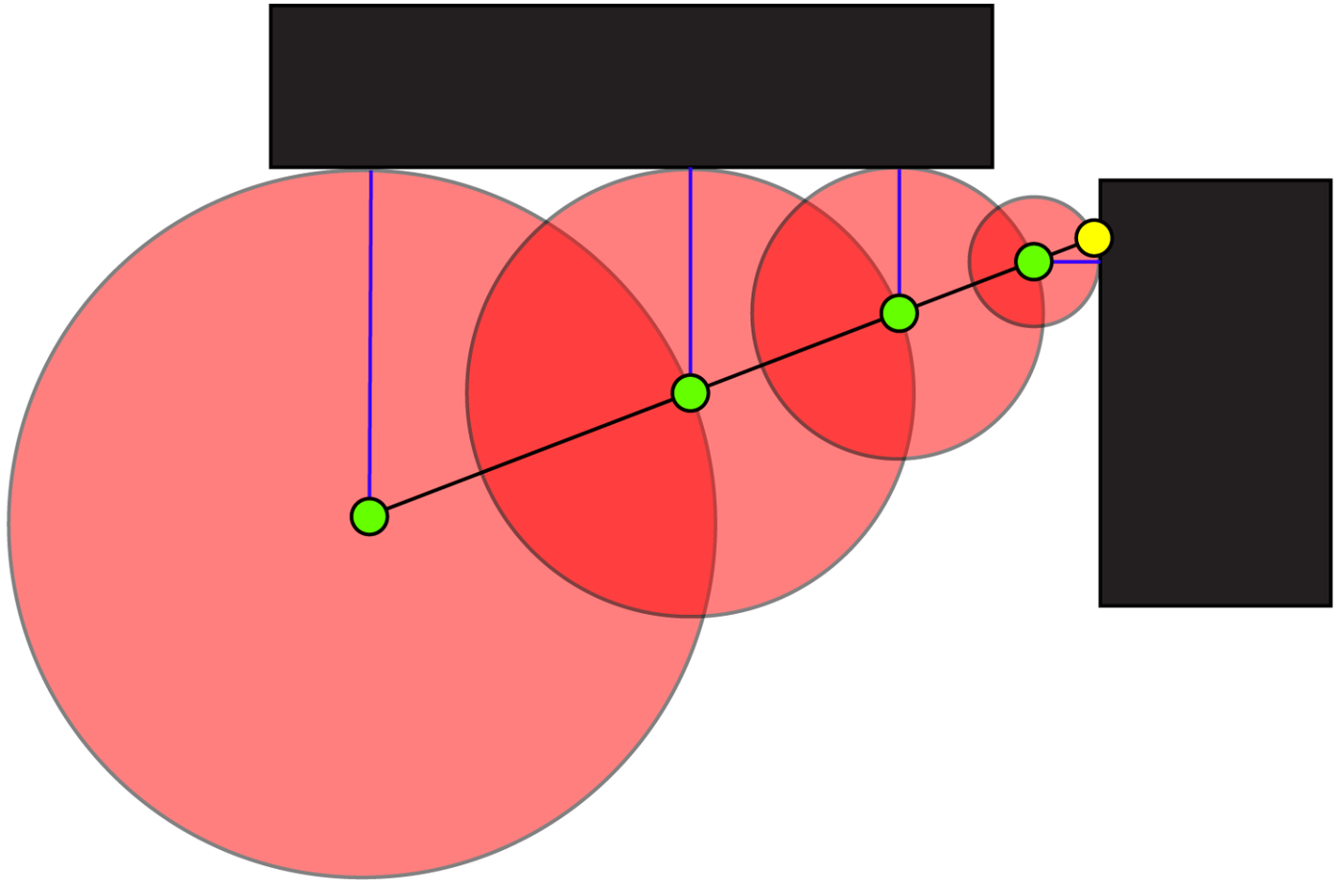


The Raymarcher

- How it works: cast a ray like in raytracing. Instead of computing the t_isect for a ray, advance t by some delta_t until we (approximately) hit a surface.

- Distance field gives us the distance to the closest surface; let's us know how much we can safely advance delta_t.

- When distance field is very small, we have a hit!



Fractal Microgeometry

- The main idea: use distance field as a “texture” that we can render locally with the raymarcher. Use this to add small details to traditional triangle meshes.
- First extend each triangle in the scene to a delta-radius shell.
- Use the standard triangle rasterizer to find the shell the ray hits.
- Use the raymarcher in the vertex shader to find the ray collision point. Raymarch on distance field with local (u,v,h) coordinates. (u,v) are standard texture coordinates, h is the distance to the triangle’s surface (before extending to a shell).