Ray Tracing Goodies

Goodies

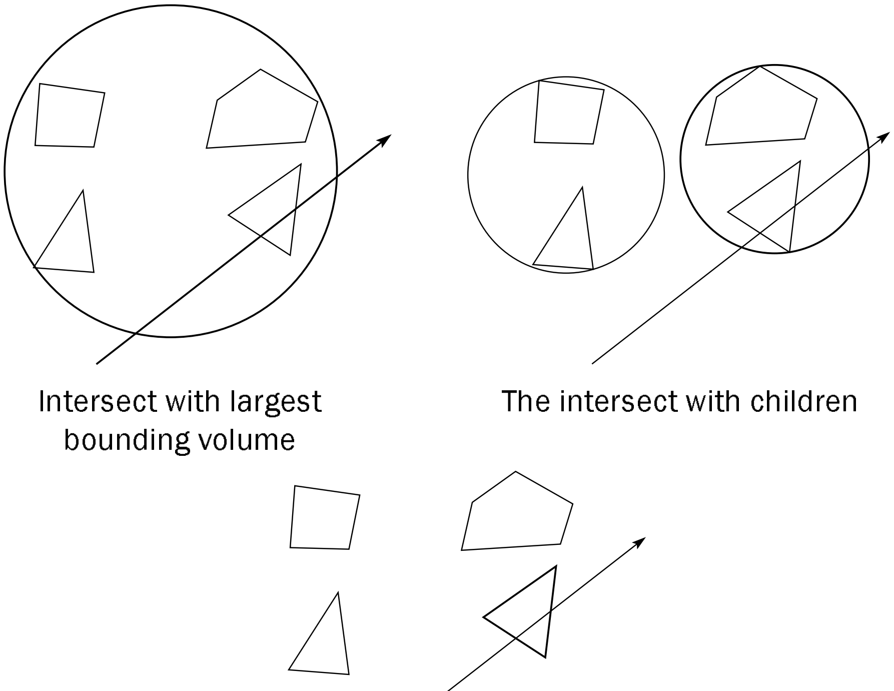
- There are some advanced ray tracing feature that self-respecting ray tracers shouldn’t be caught without:
  - Acceleration techniques
  - Antialiasing
  - CSG
  - Distribution ray tracing

- There are some features that dramatically increase the power of a ray tracer:
  - Particle systems
  - Caustics and participating media
  - …

Acceleration Techniques

- Problem: ray-object intersection is very expensive
- So make intersection tests faster and do fewer tests

Hierarchical Bounding Volumes



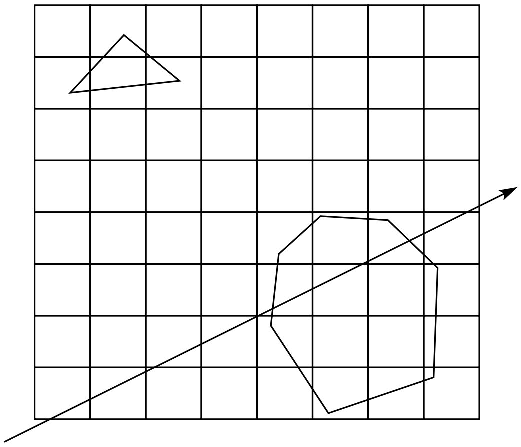
Intersect with largest bounding volume

The intersect with children

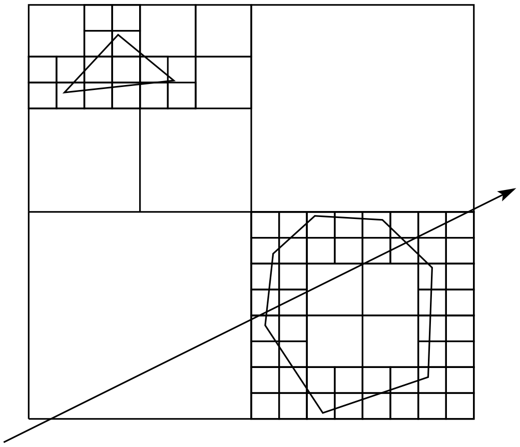
Eventually, intersect with primitives

- Arrange scene into a tree
  - Interior nodes contain primitives with very simple intersection tests (e.g., spheres). Each node’s volume contains all objects in subtree
  - Leaf nodes contain original geometry
- Like BSP trees, the potential benefits are big but the hierarchy is hard to build

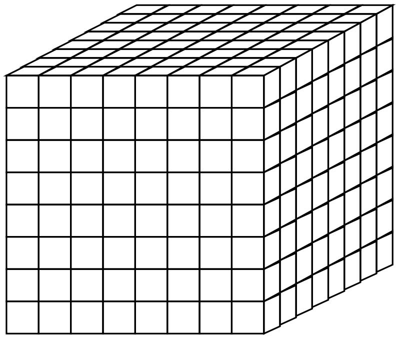
Spatial Subdivision



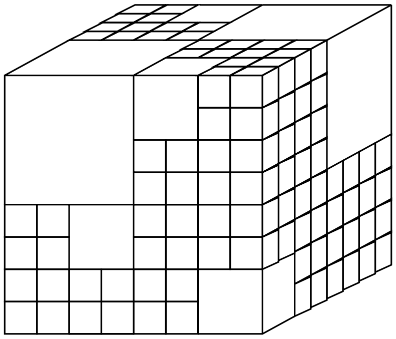
Uniform subdivision
in 2D



Quadtree



Uniform subdivision
in 3D

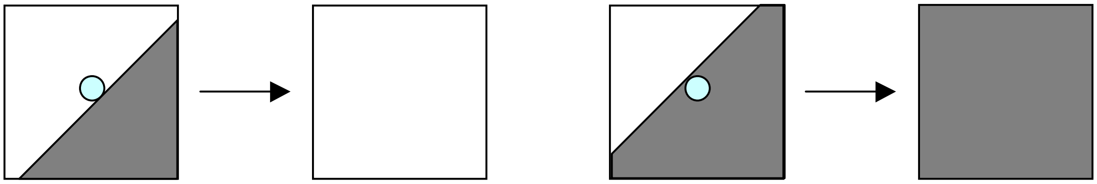


Octree

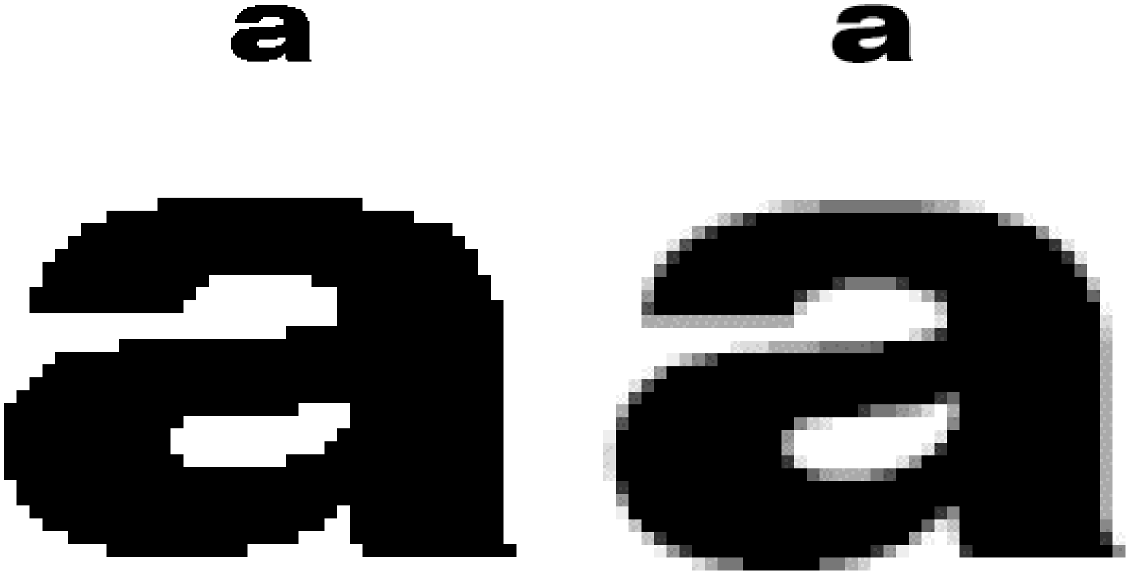
- Divide up space and record what objects are in each cell
- Trace ray through voxel array

Antialiasing

- So far, we have traced one ray through each pixel in the final image. Is this an adequate description of the contents of the pixel?

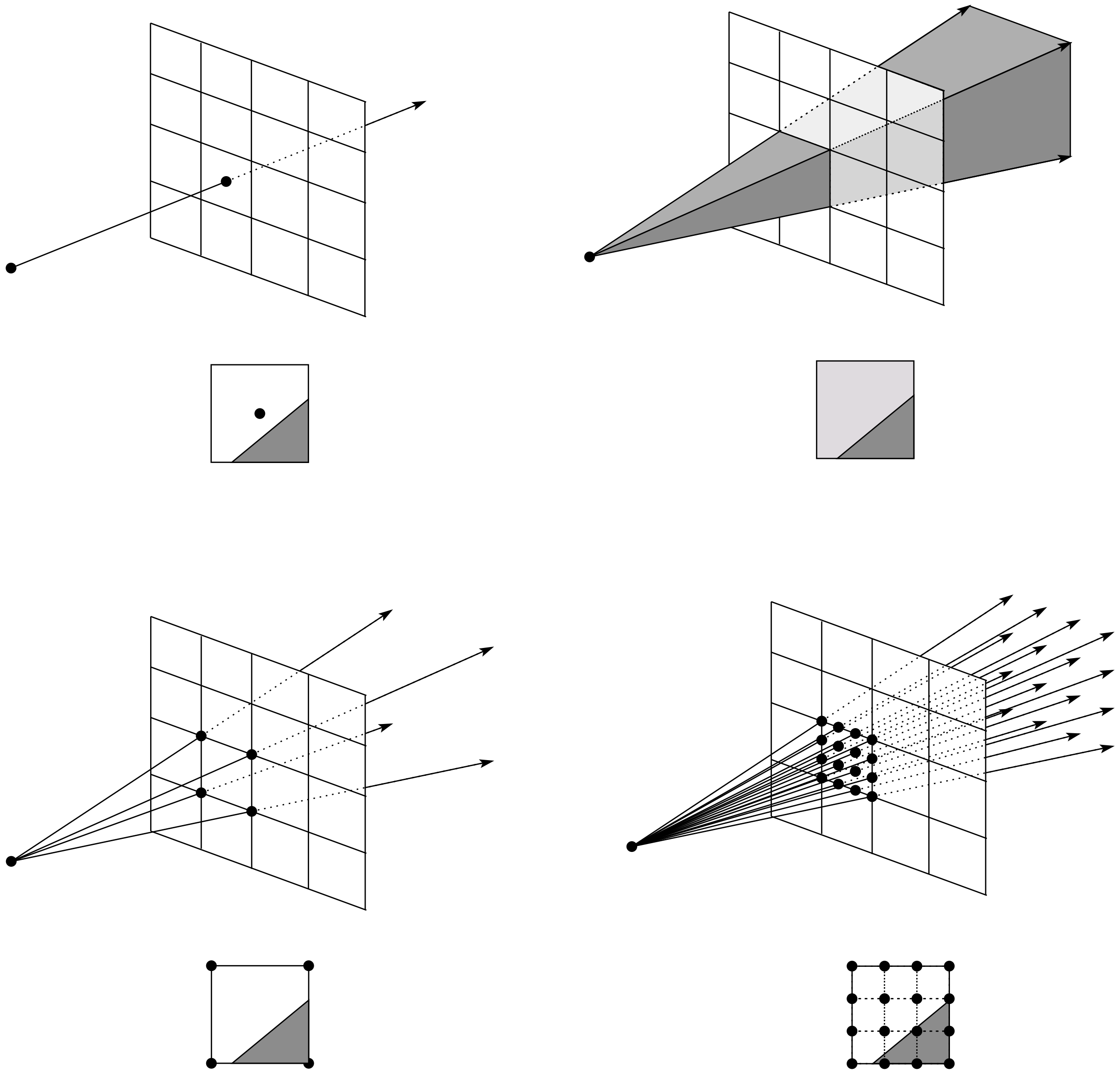


- This quantization through inadequate sampling is a form of aliasing. Aliasing is visible as “jaggies” in the ray-traced image.
- We really need to colour the pixel based on the average colour of the square it defines.



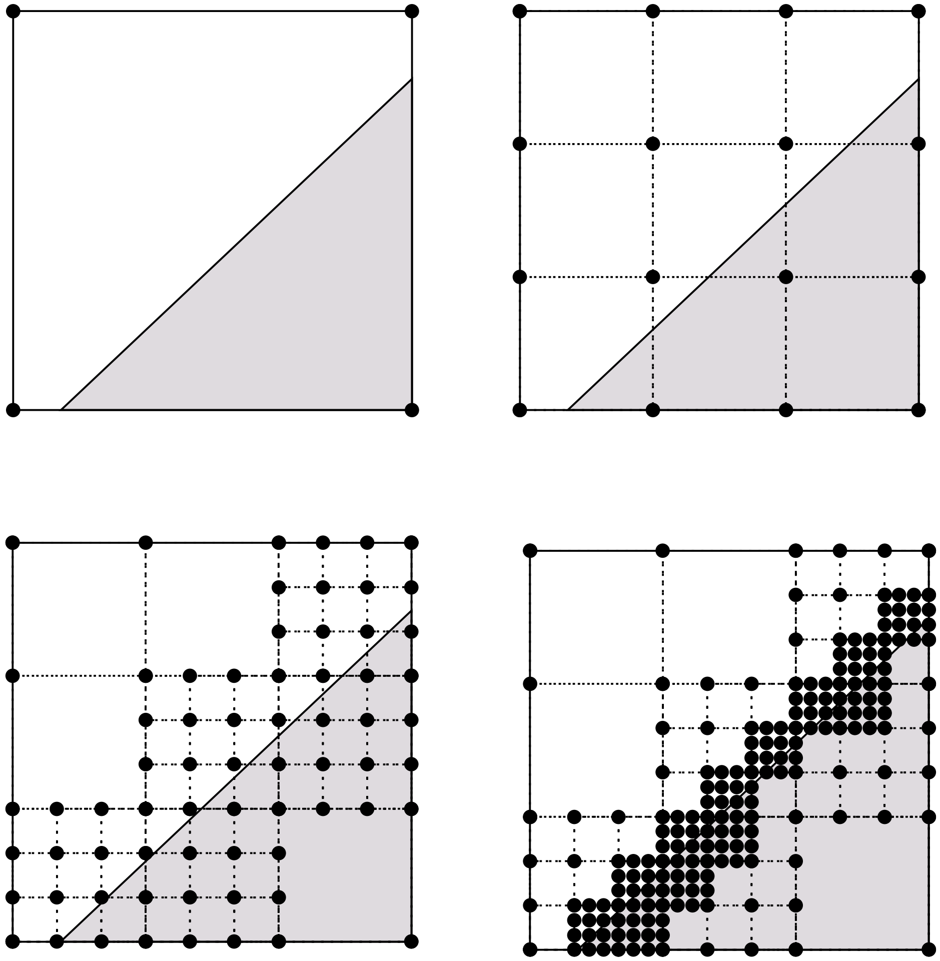
Supersampling

- We can approximate the average colour of a pixel's area by firing multiple rays and averaging the result.



Adaptive Sampling

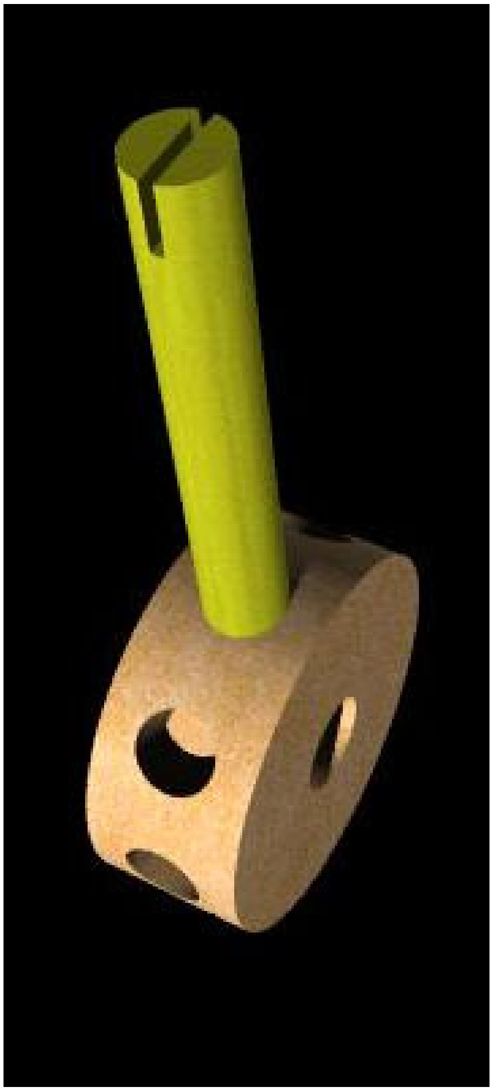
- Uniform supersampling can be wasteful if large parts of the pixel don't change much.
- So we can subdivide regions of the pixel's area only when the image changes in that area:

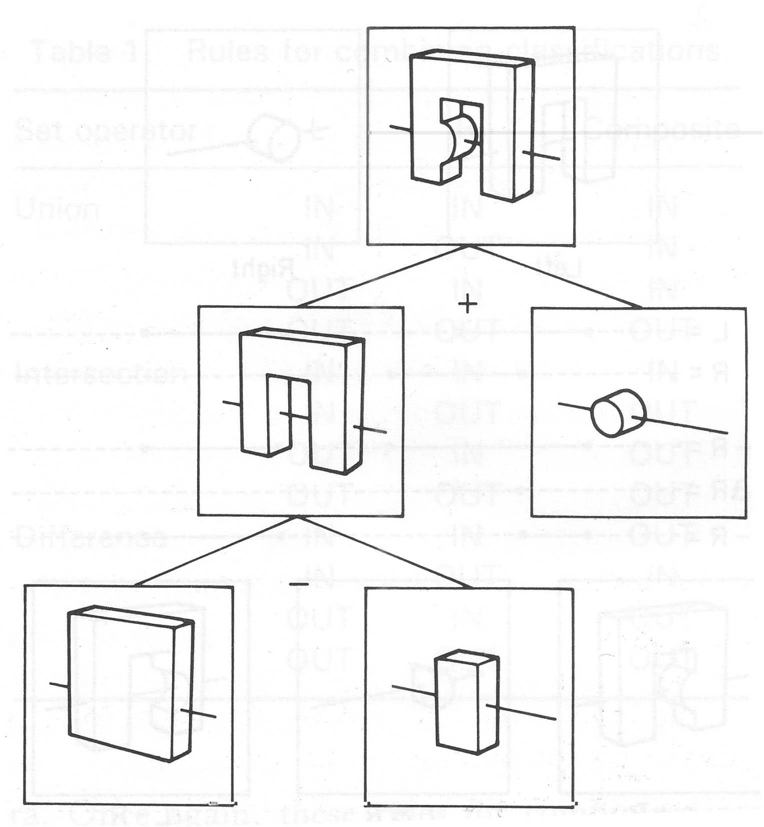


- How do we decide when to subdivide?

CSG

- CSG (constructive solid geometry) is an incredibly powerful way to create complex scenes from simple primitives.

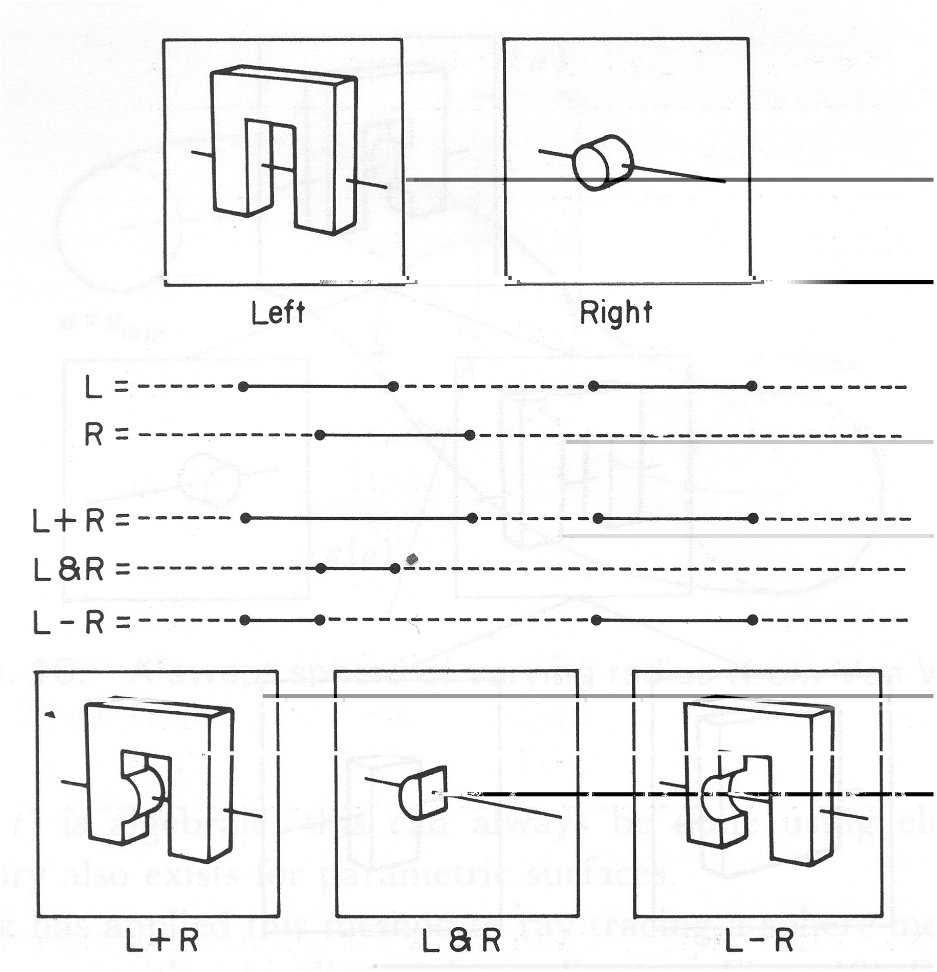




- CSG is a modeling technique; basically, we only need to modify ray-object intersection.

CSG Implementation

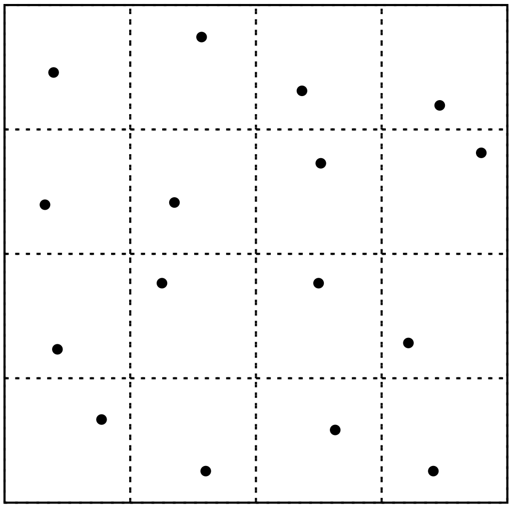
- CSG intersections can be analyzed using “Roth diagrams”.
  - Maintain description of all intersections of ray with primitive
  - Functions to combine Roth diagrams under CSG operations



- An elegant and extremely slow system

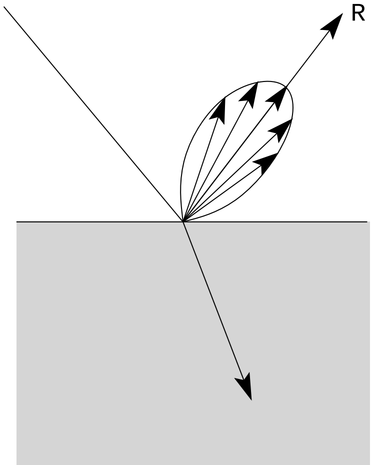
Distribution Ray Tracing

- Usually known as “distributed ray tracing”, but it has nothing to do with distributed computing
- General idea: instead of firing one ray, fire multiple rays in a jittered grid

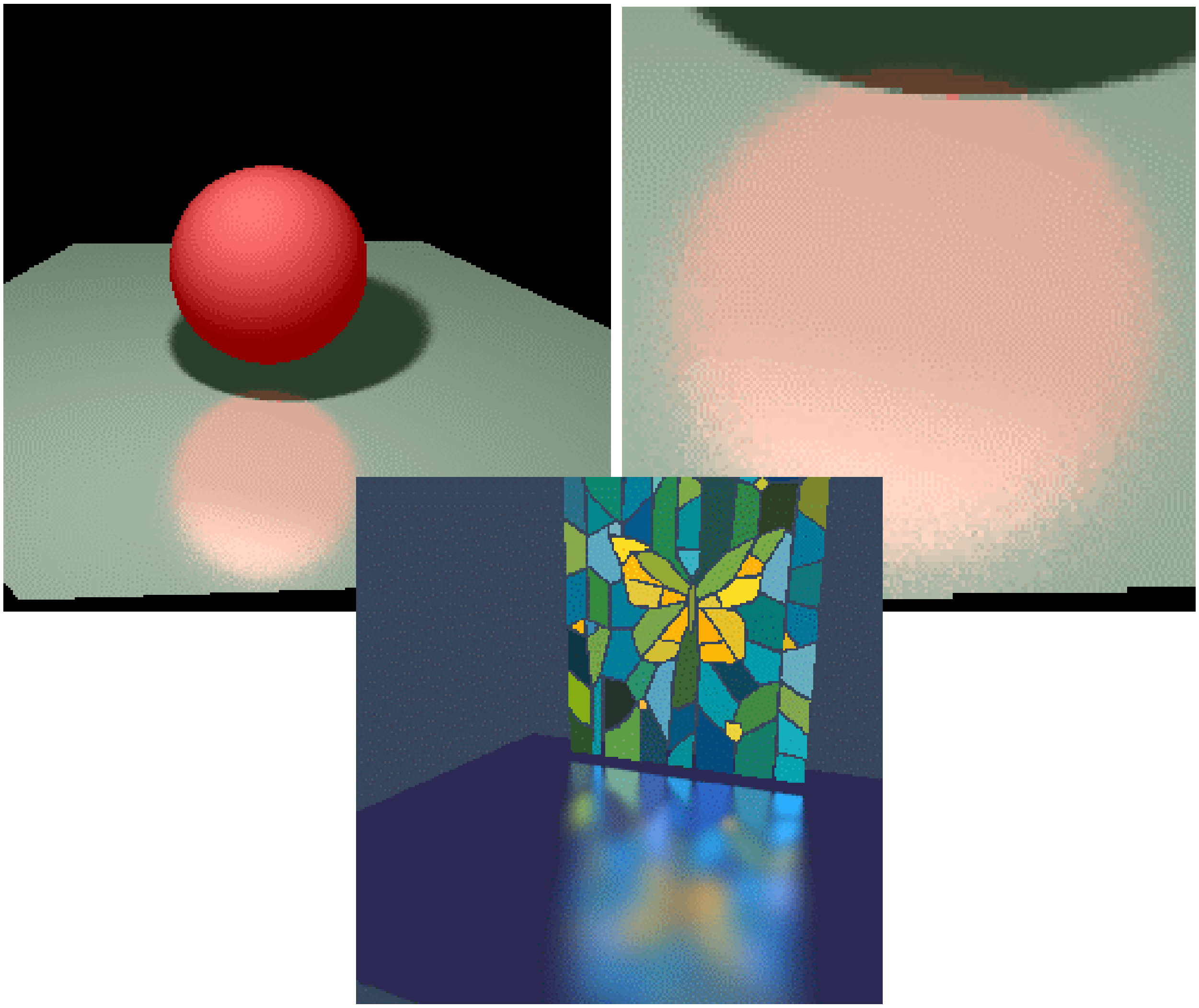


- Distributing over different dimensions gives different effects
- Example: what if we distribute rays over pixel area?

Distributing Reflections

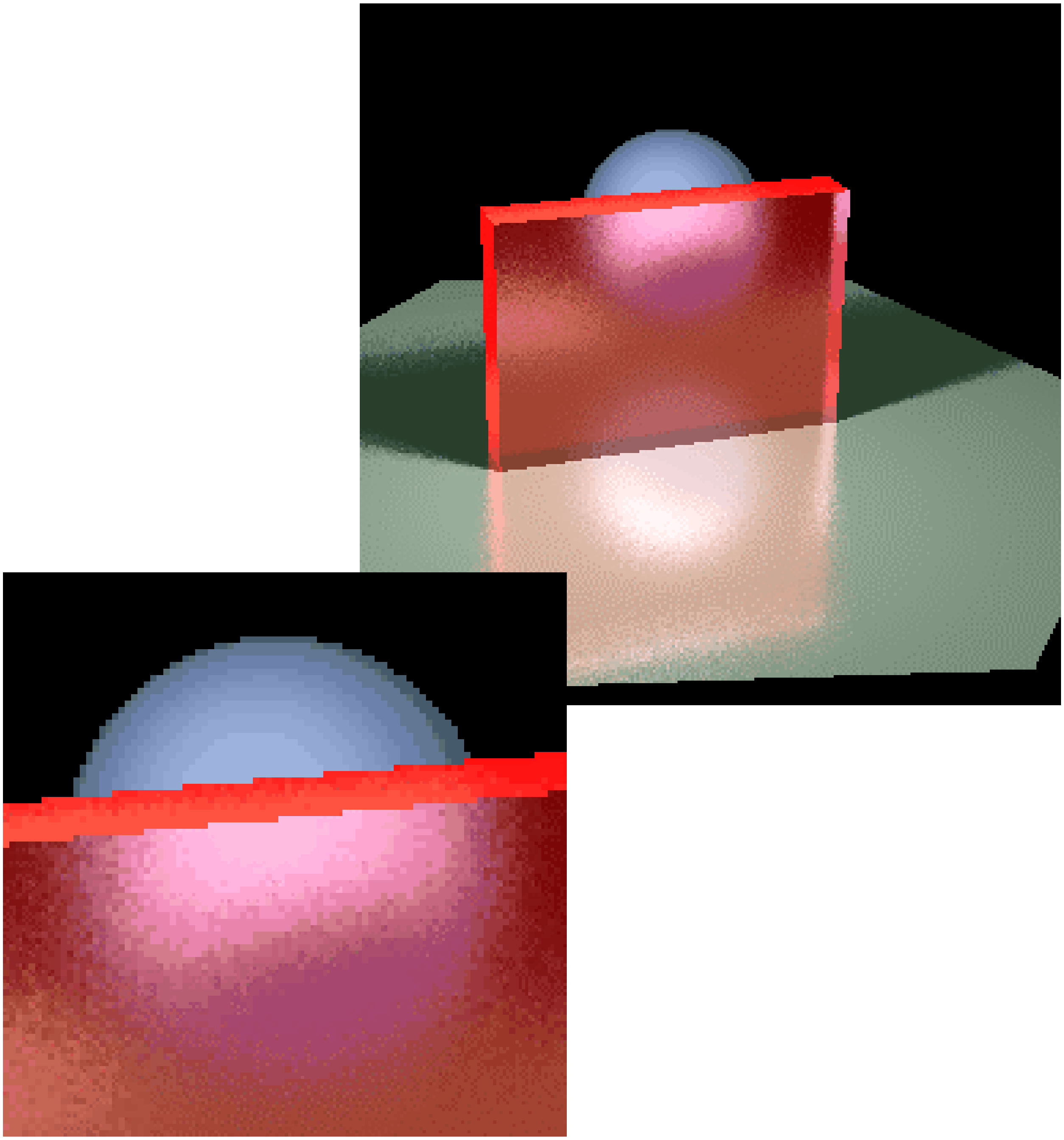


- Distributing rays over reflection direction gives:



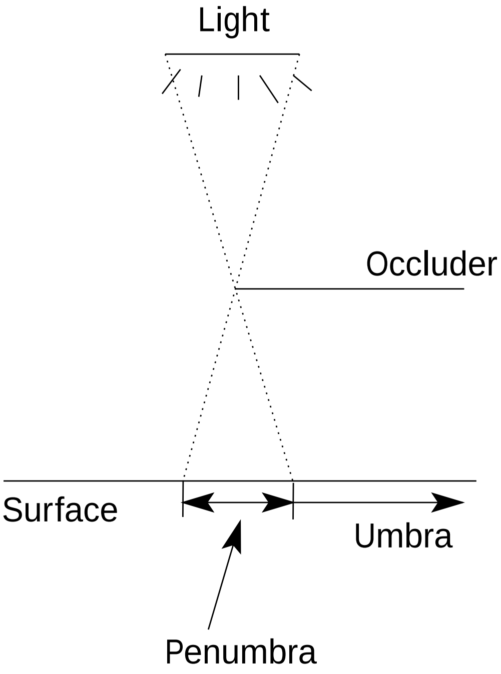
Distributing Refractions

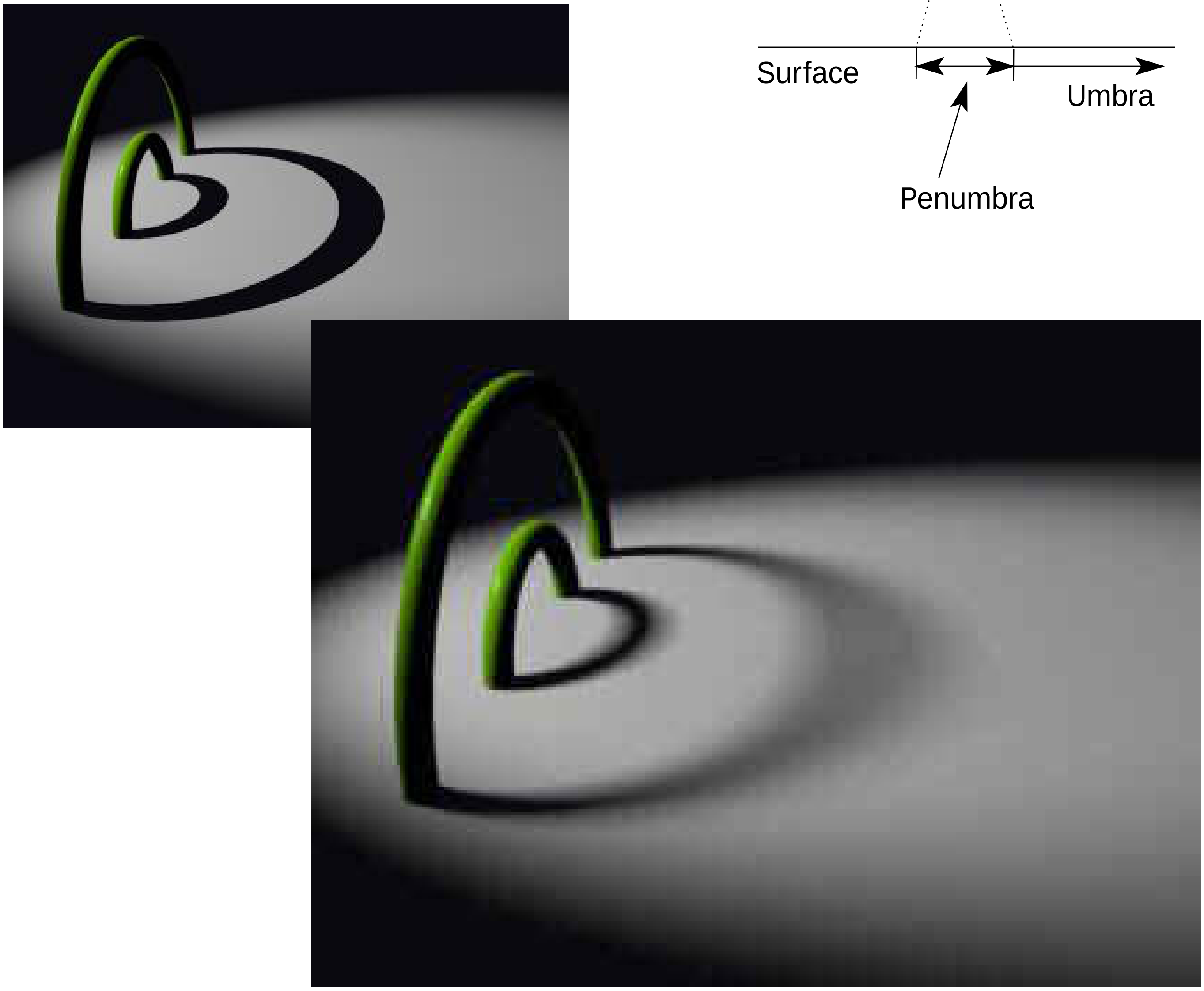
- Distributing rays over transmission direction gives:



Distributing Over Light Area

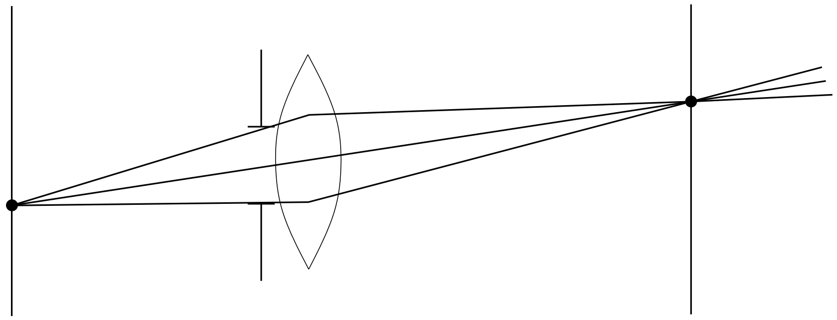
- Distributing over light area gives:



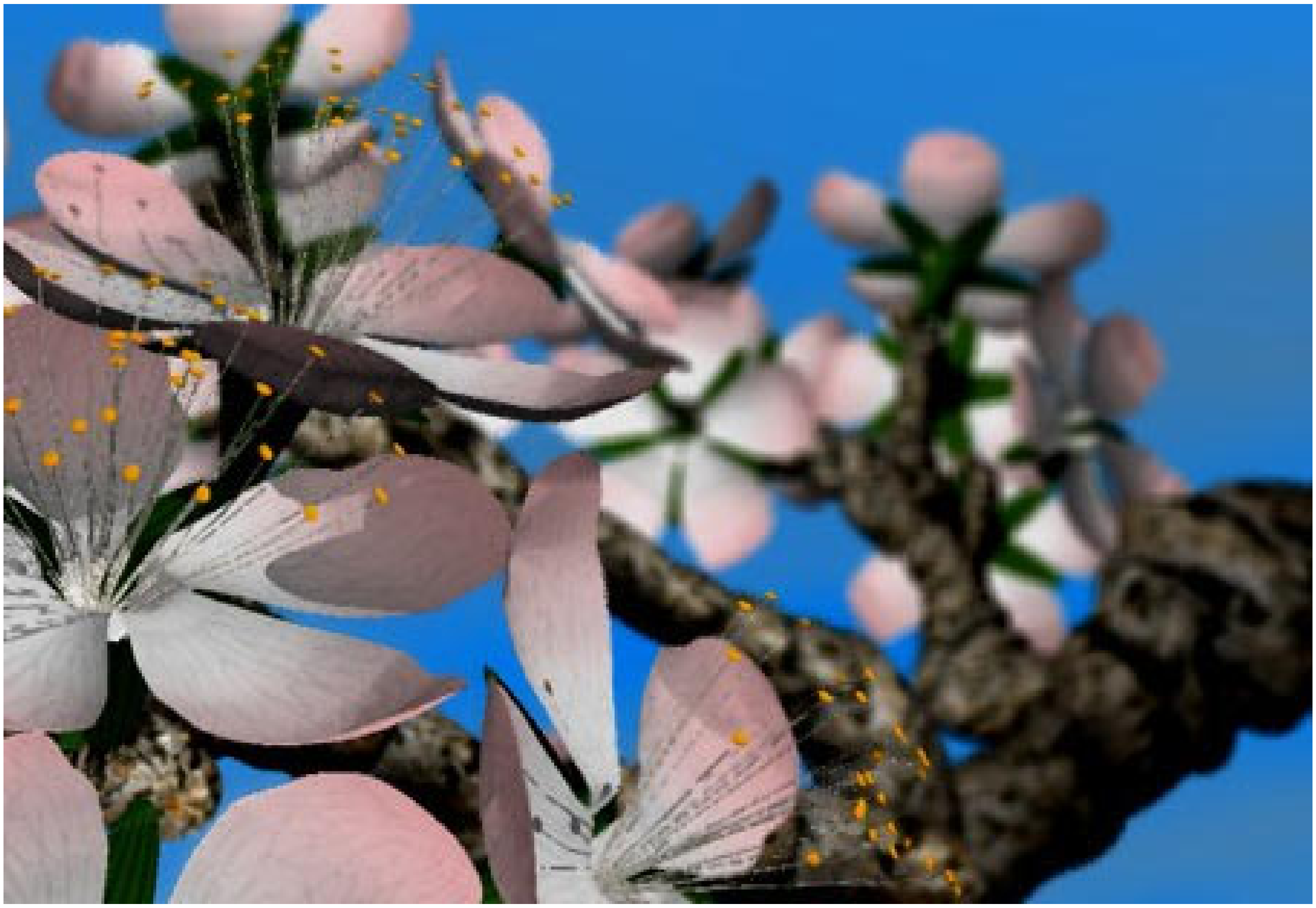


Distributing Over Aperature

- We can fake distribution through a lens by choosing a point on a finite aperature and tracing through the “in-focus point”.

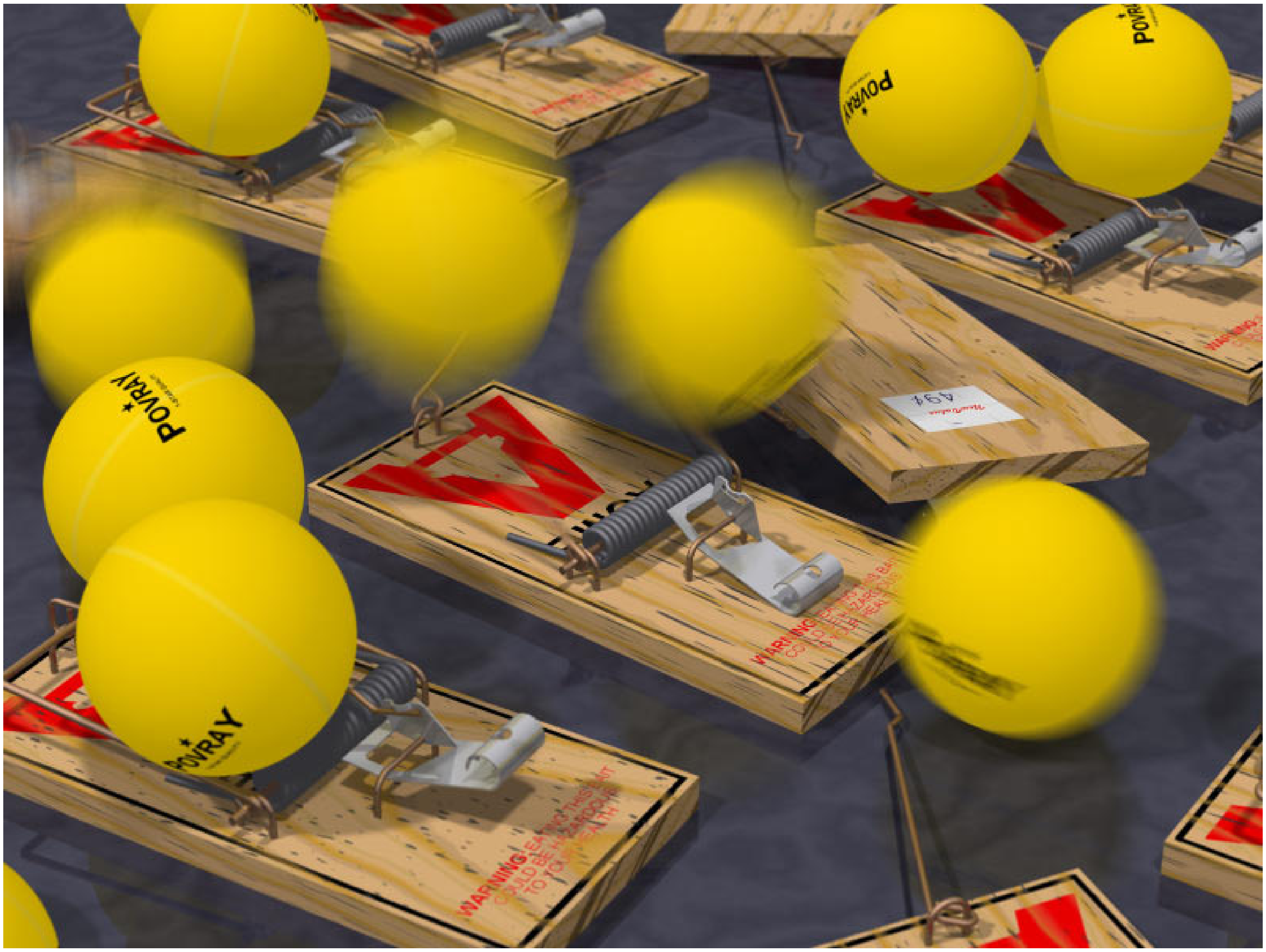


- What does this simulate?



Distributing Over Time

- We can endow models with velocity vectors and distribute rays over time. this gives:



Summary

- Understanding of the idea and implementation strategies for:

- Ray tracing acceleration
  - Hierarchical bounding volumes
  - Spatial subdivision
- Antialiasing
  - Supersampling
  - Adaptive sampling
- CSG
- Distribution ray tracing
  - Antialiasing
  - Glossy reflection
  - Translucency
  - Soft shadows
  - Depth-of-field
  - Motion blur