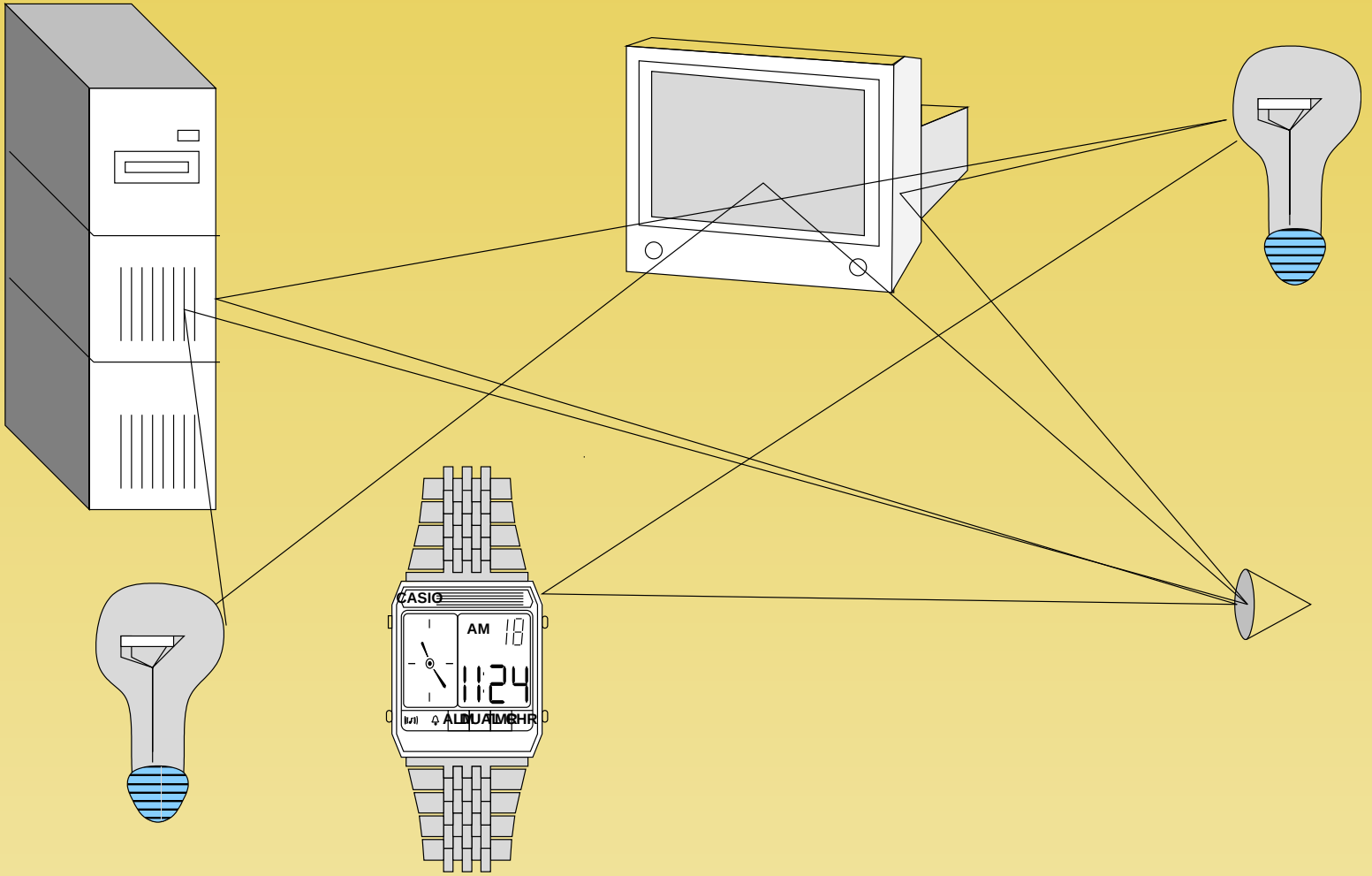
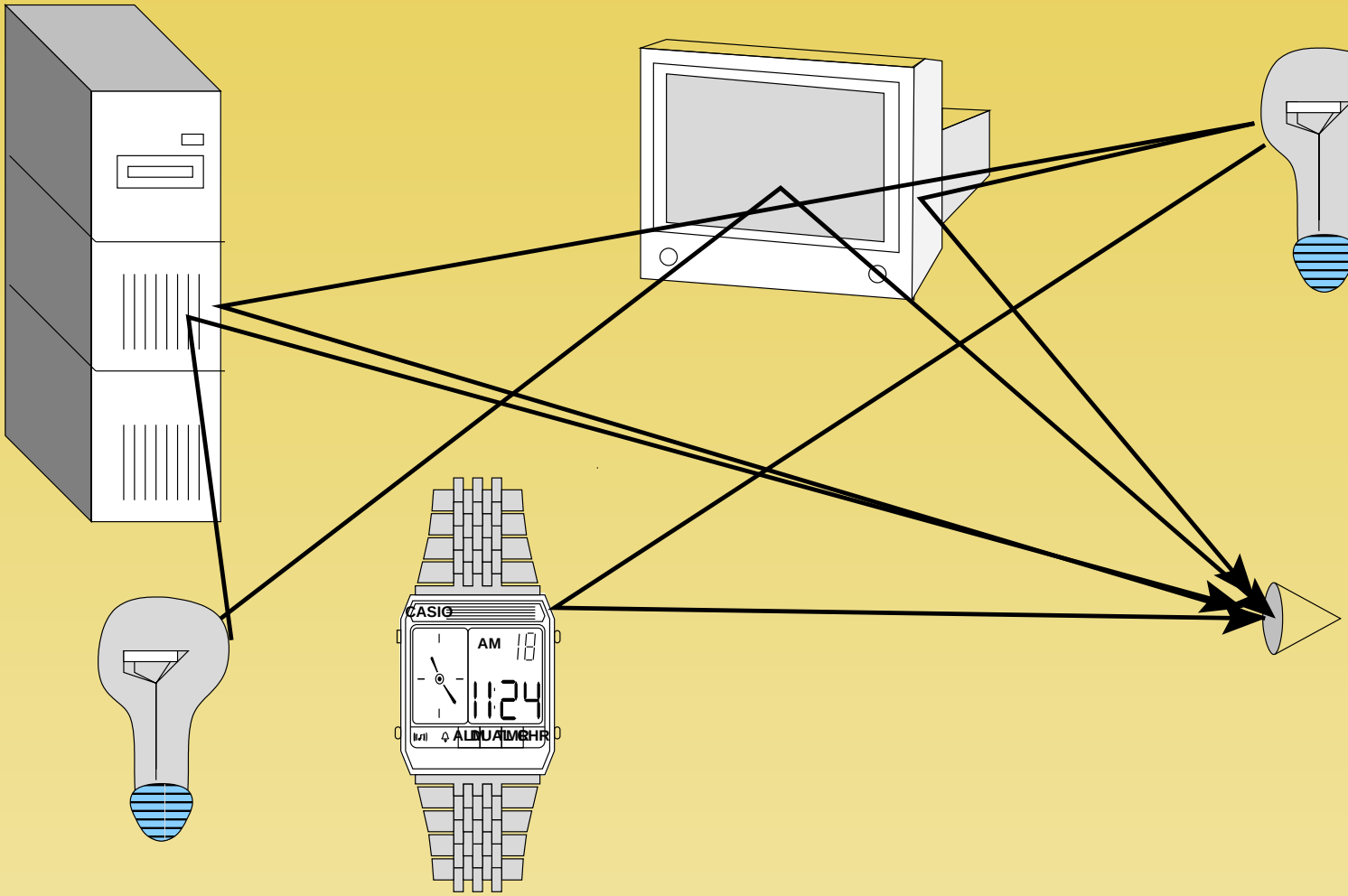
How does light get to our eyes?



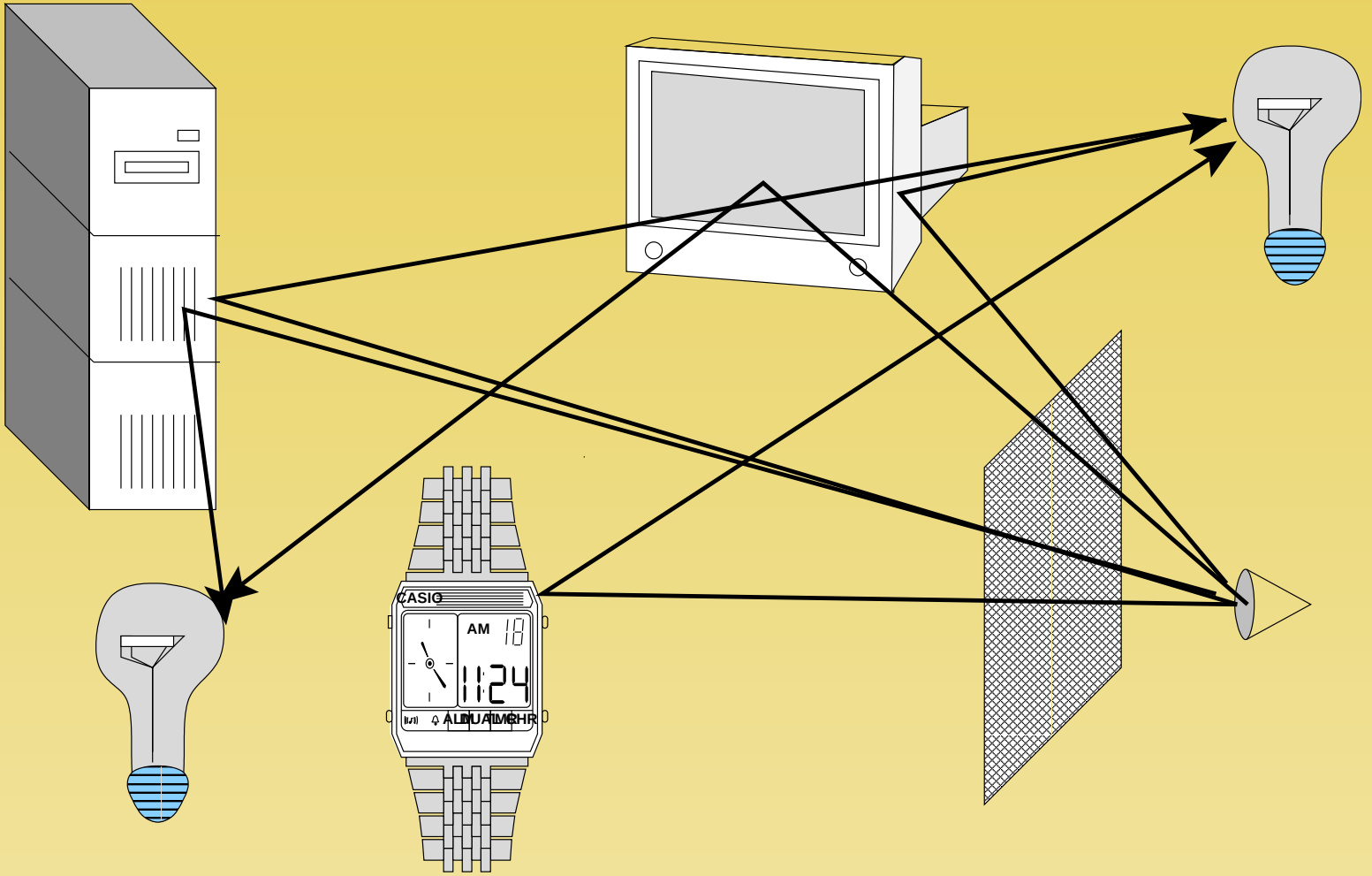
Light Comes Forward From its Source(s).



Forward Tracing

- Can this be done? Yes
- How would we do it?
  - ⋆ Well, we would have to trace all rays from all lights.
  - ⋆ If they end up hitting our eye (camera) then we add in it's contribution.
- Is this efficient? No
- Is it accurate? Yes
- What could we do instead?

Backward Tracing.



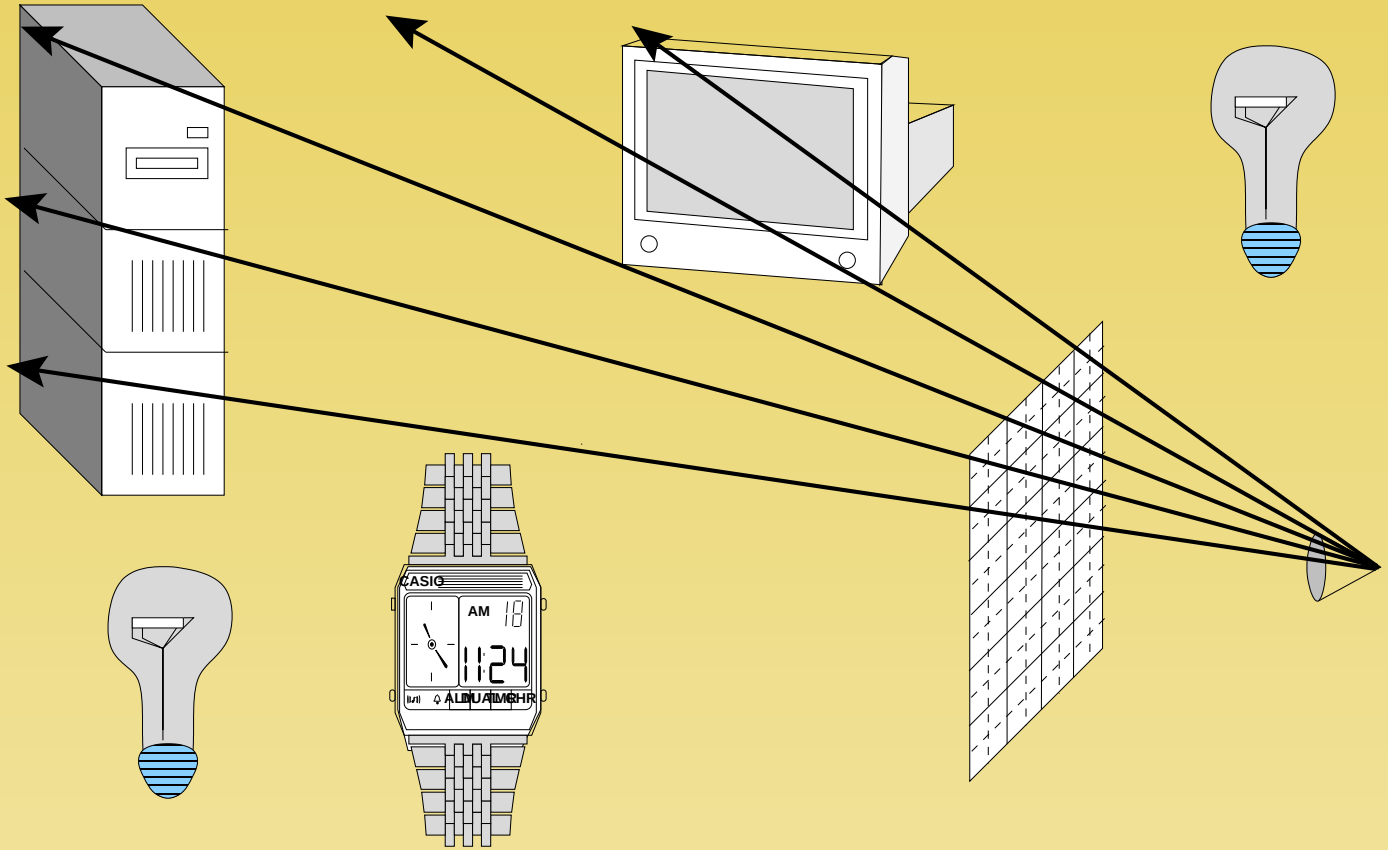
Basic Ray Casting method

$\forall$ pixels in screen

1. Shoot ray $\vec{p}$ from the eye through the pixel.
2. Find closest ray-object intersection.
3. Get color at intersection.

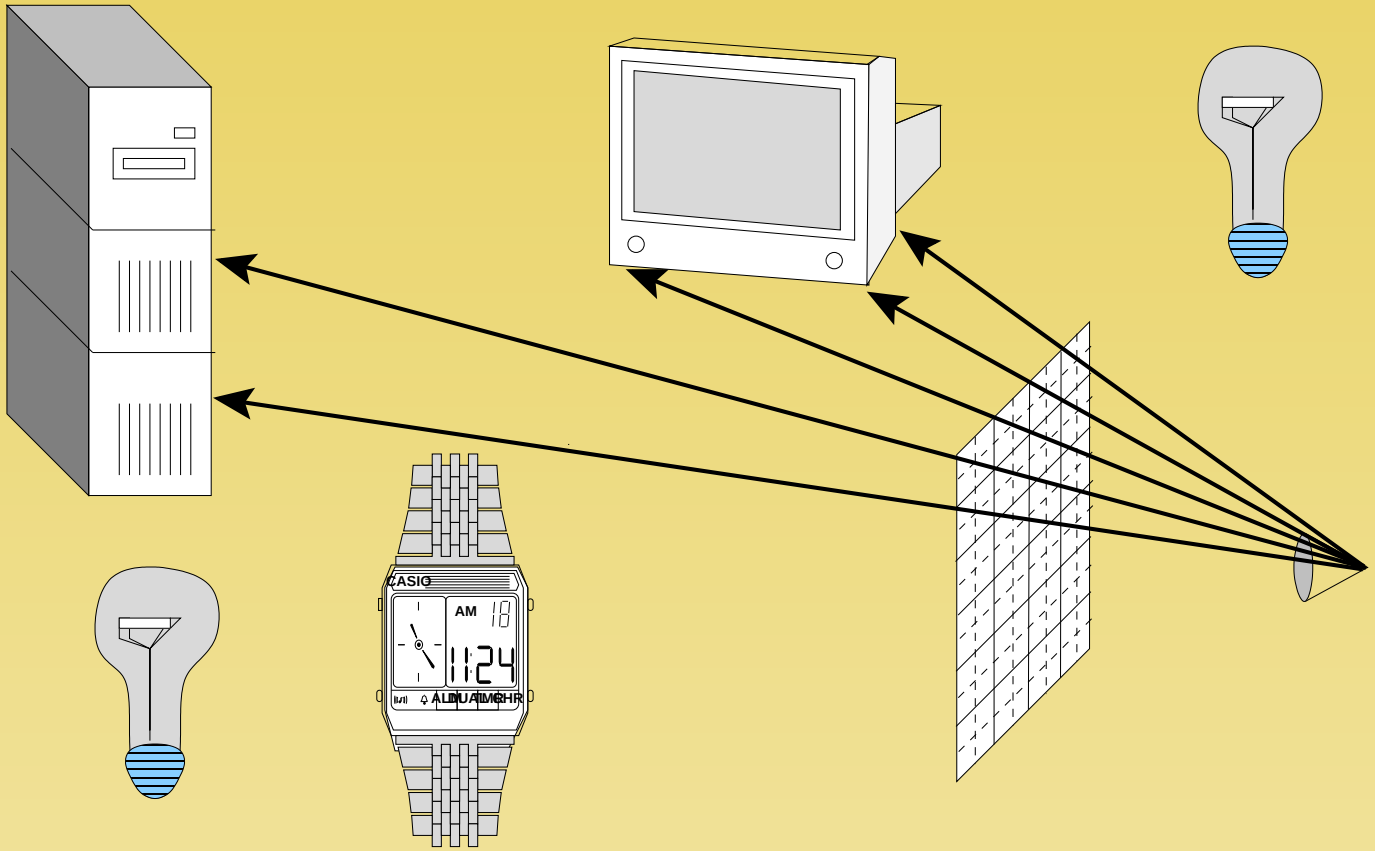
Basic Ray Casting method

Shoot ray $\vec{p}$ from eye through pixel.



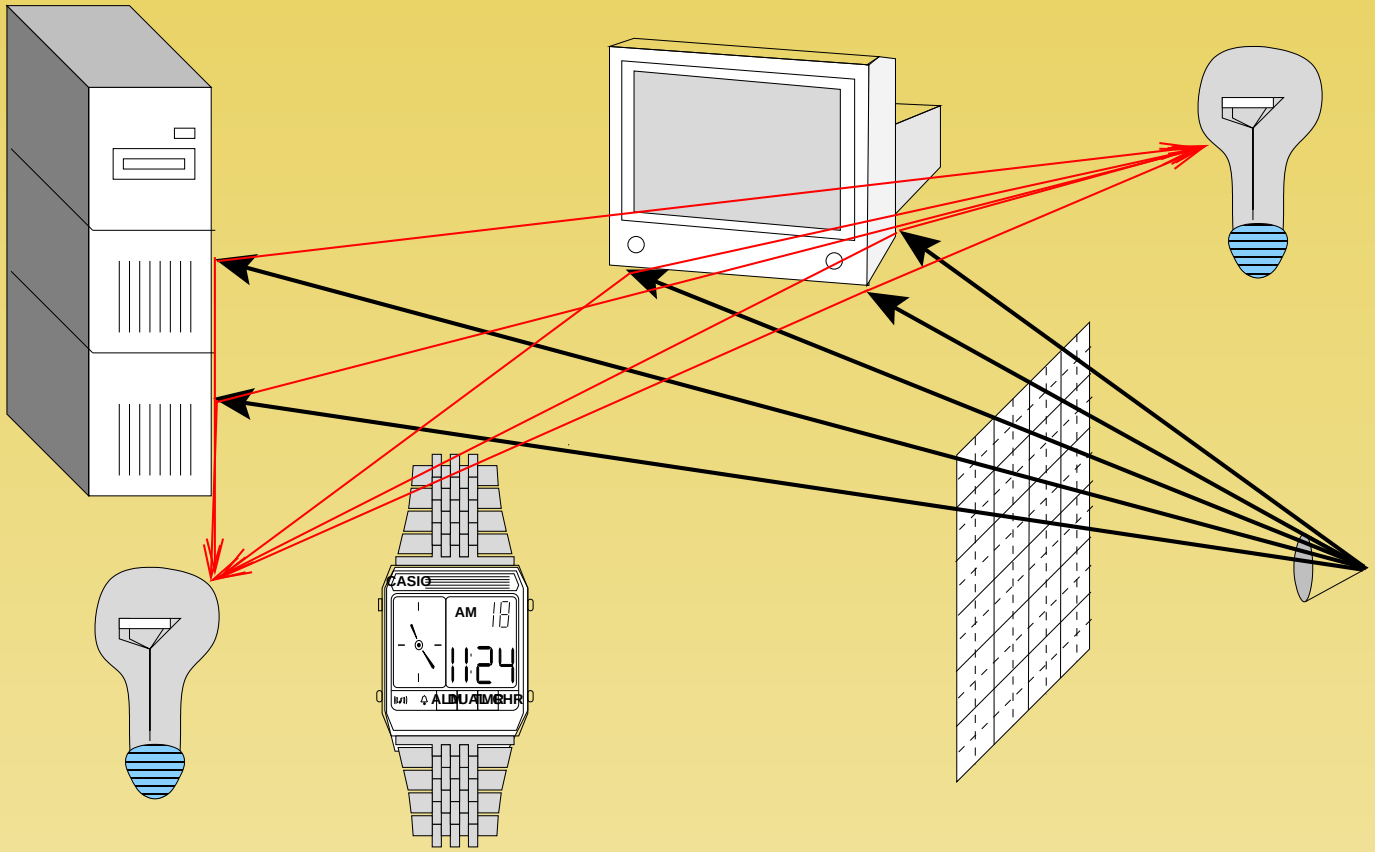
Basic Ray Casting method

Find closest ray-object intersection.



Basic Ray Casting method

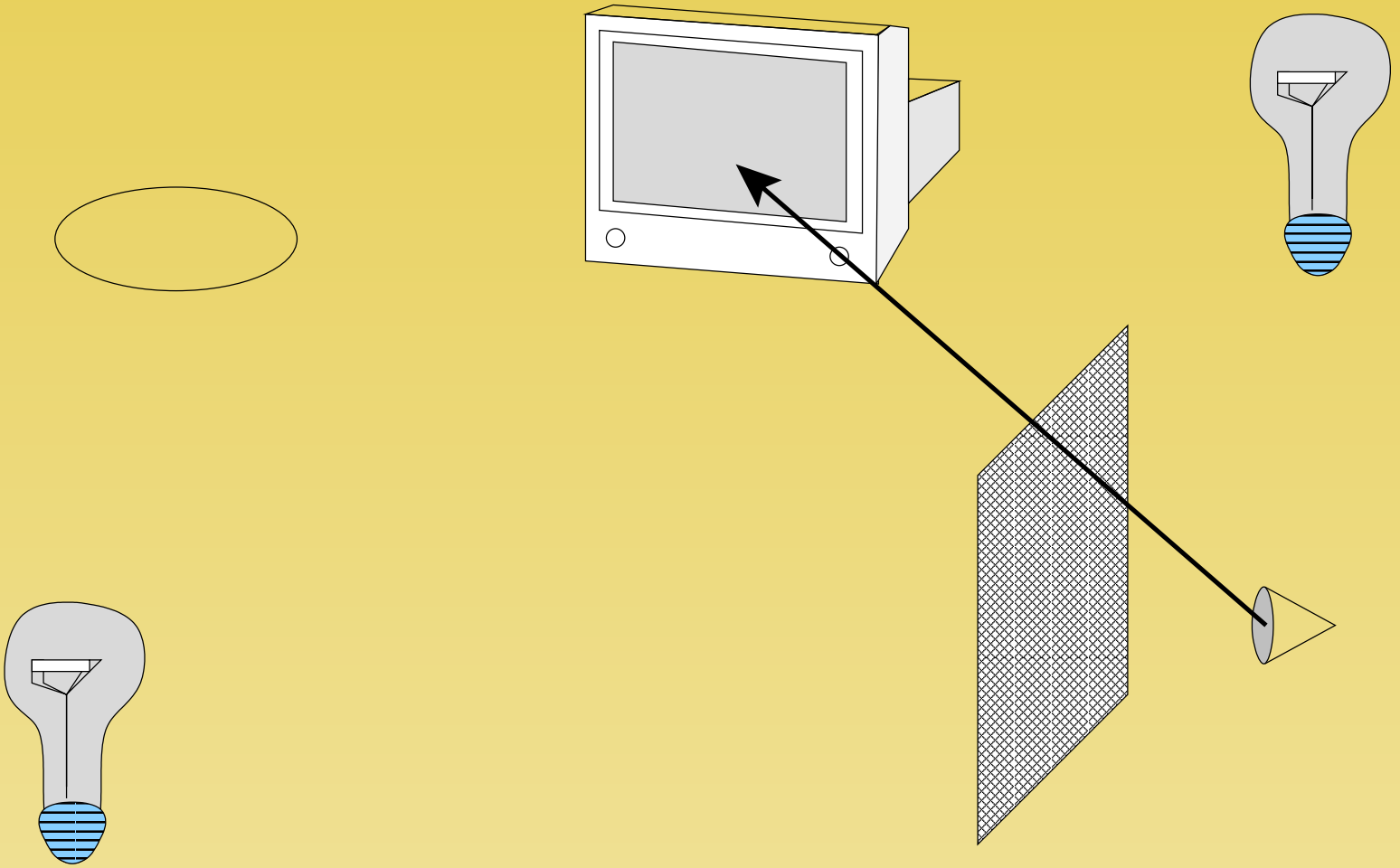
Get color at intersection.



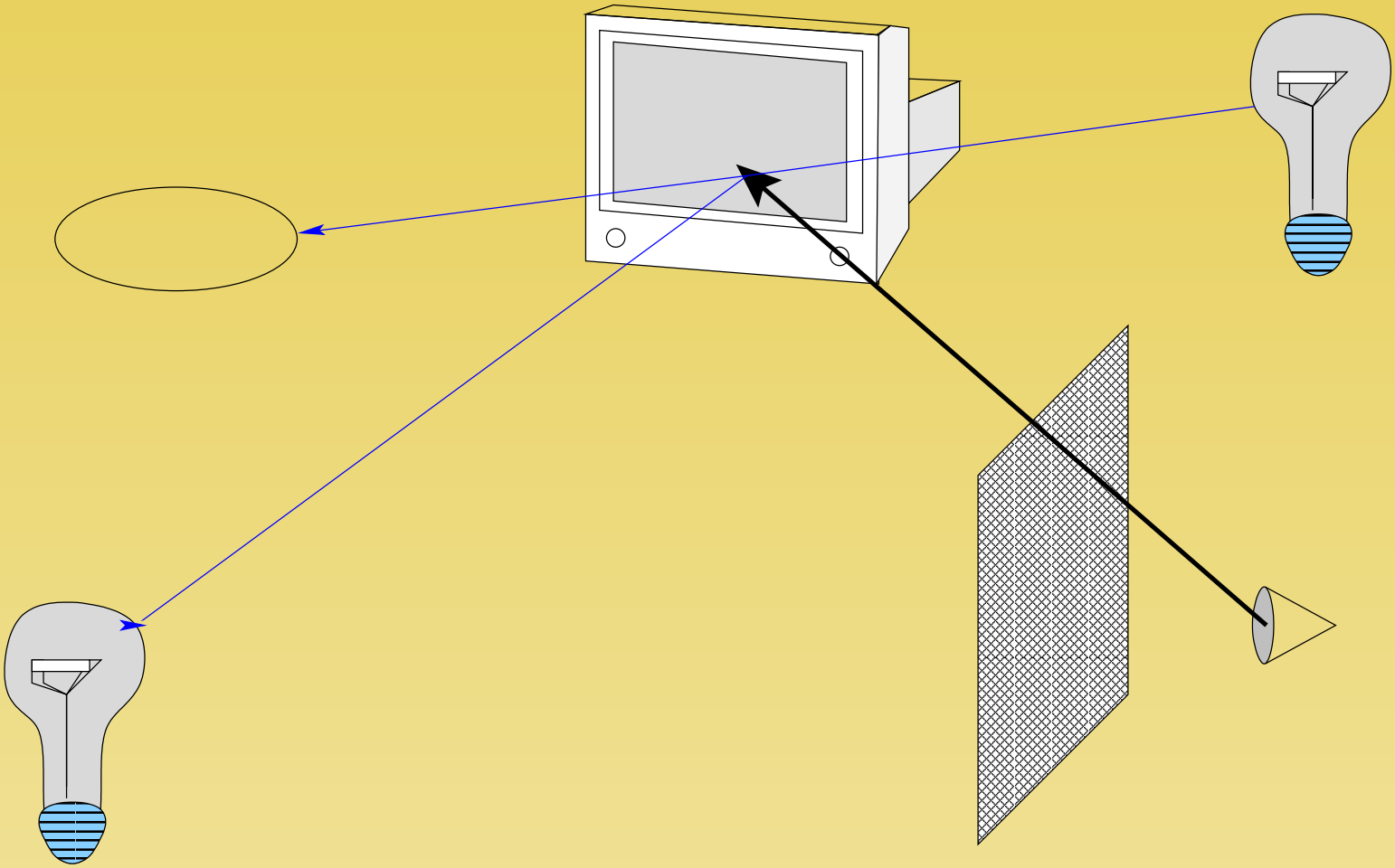
Ray Genealogy

- Primary rays spawn off 3 rays. Two of those can spawn of 3 more, etc.
- When do you stop?
  - ⋆ When ray leaves the scene.
  - ⋆ When the contribution is small enough - after each bounce the contribution is attenuated by the k's in the illumination model. So we set a recursion level.

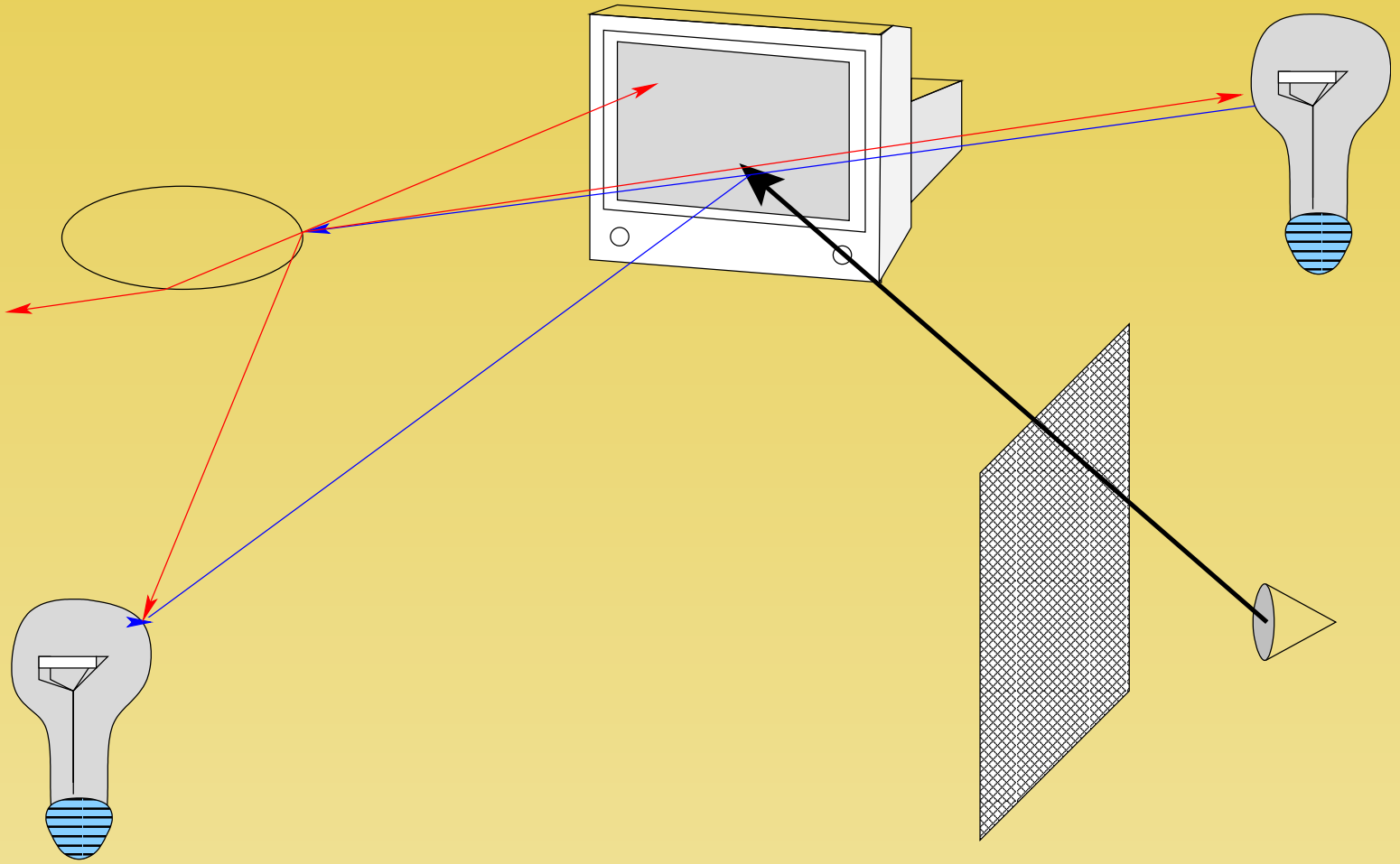
Ray Genealogy



Ray Genealogy



Ray Genealogy



Ray Genealogy

- 1 Ray/pixel at $1k \times 1k$ image $= 1M$ rays.
- Say on average 6 secondary rays.$\rightarrow 7M$ rays
- $100k$ objects; 10 ops/intersection $\rightarrow 7,000,000M$ ops
- 2GHz processor 5 cycles per op $\rightarrow 400MFLOPS$
- How long to render? $70000/4 = 17500s \approx 5hr$
- Will this image look really good for our 5hrs?

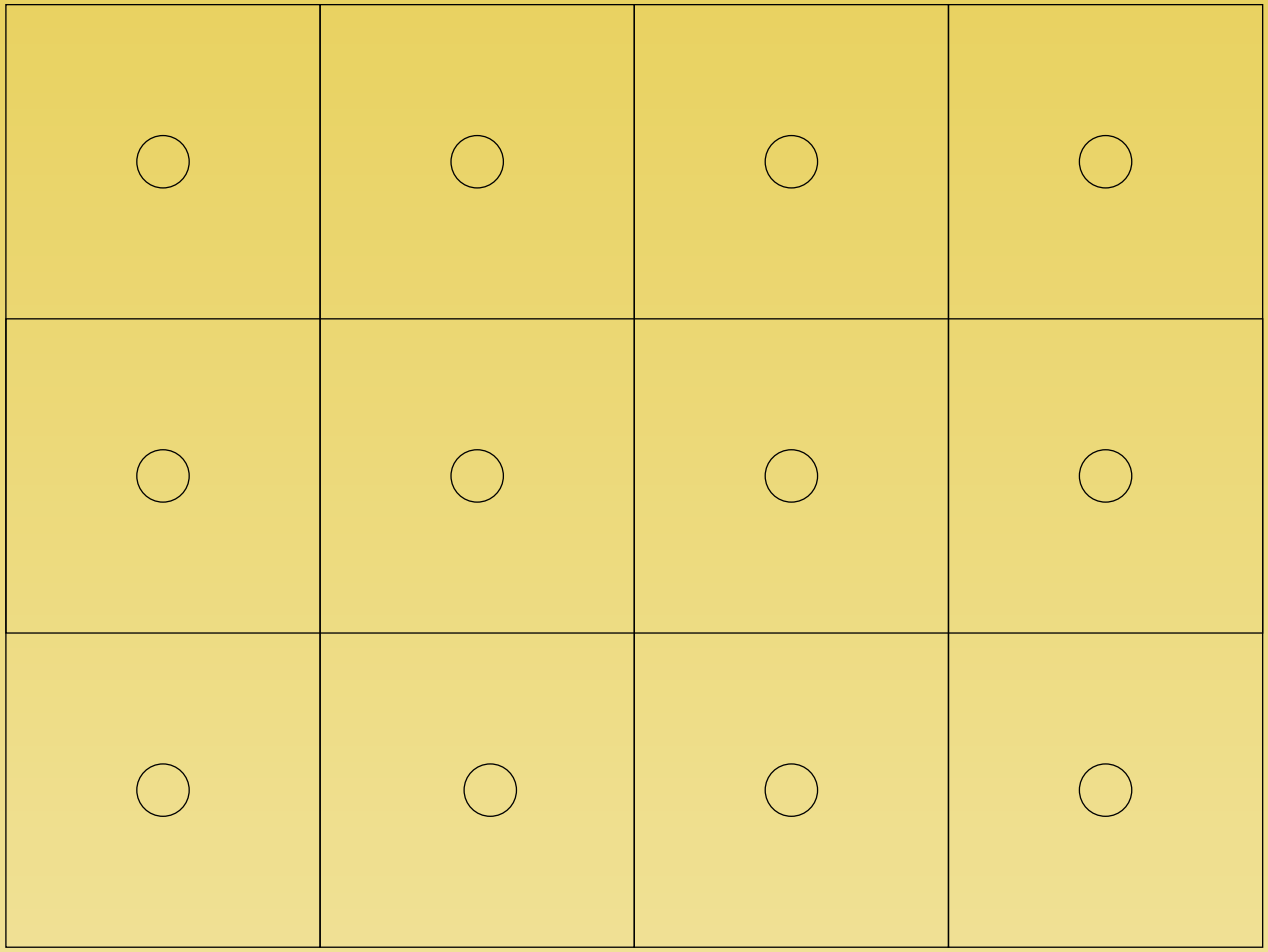
Anti-aliasing

- The pixels represent discrete samples of our image.
- Since we don’t have enough samples, higher frequencies get aliased (renamed) to lower ones.
- How can we remove the aliasing?

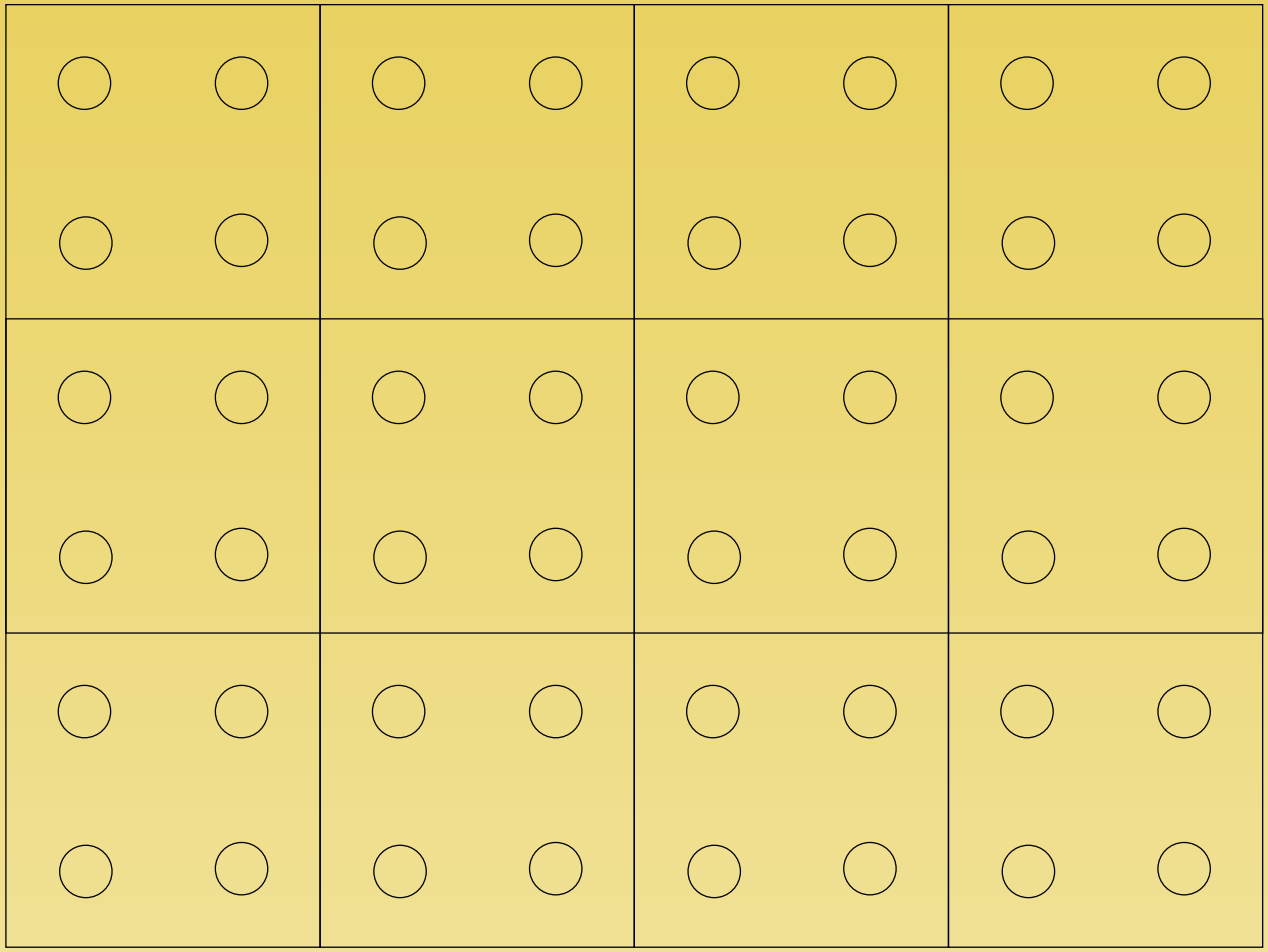
Super-sampling

- A simple method is to shoot more than one ray per pixel and average the values.

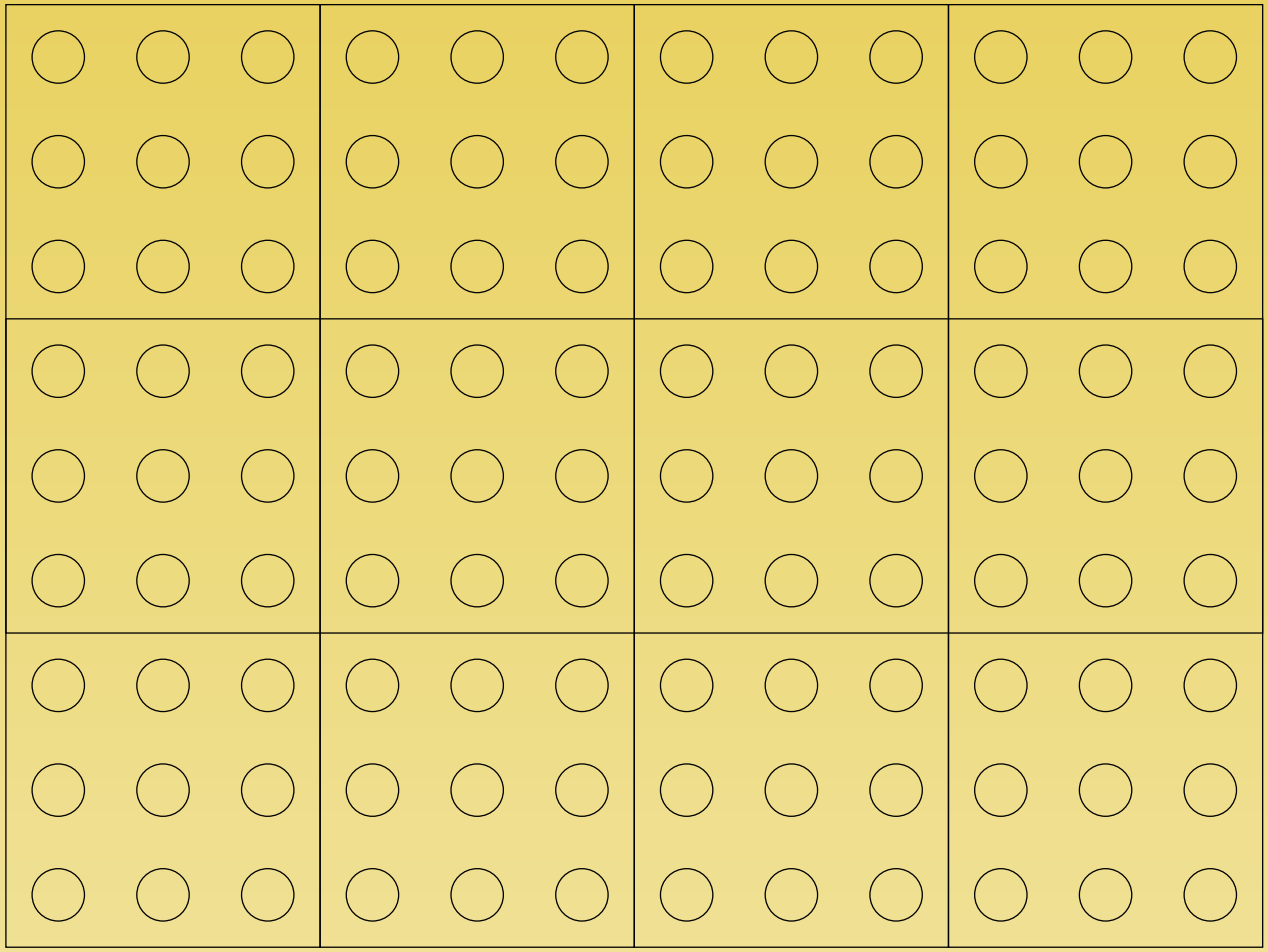
Super-sampling



Super-sampling



Super-sampling



Super-sampling

- We can do this forever.
- Can fire many many rays and still have aliasing.
- Now what?

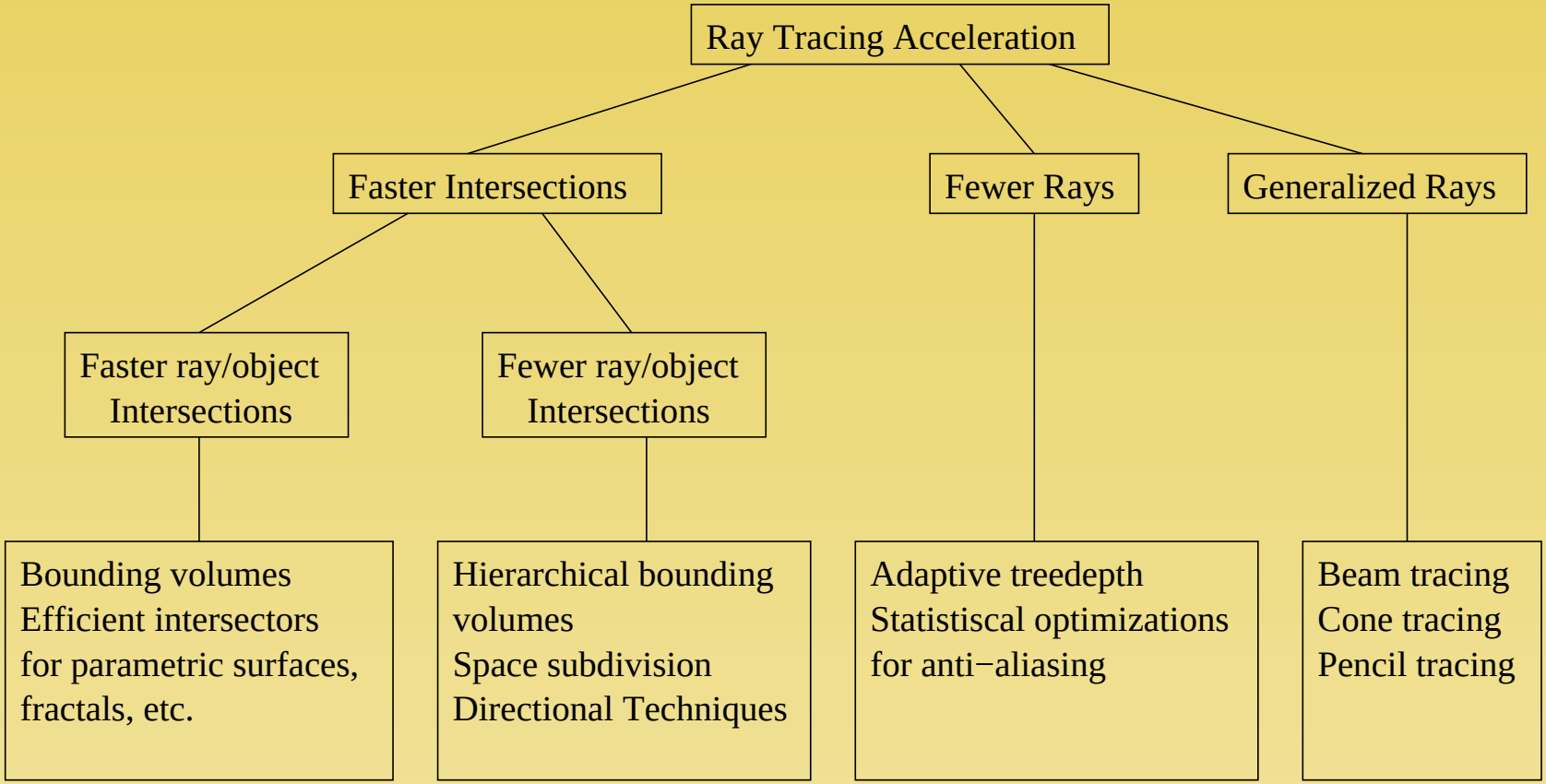
Ray Genealogy

- 1 Ray/pixel at $1k \times 1k$ image $= 1M$ rays.
- Say on average 6 secondary rays.$\rightarrow 7M$ rays
- $100k$ objects; 10 ops/intersection $\rightarrow 7,000,000M$ ops
- 2GHz processor 5 cycles per op $\rightarrow 400MFLOPS$
- How long to render? $70000/4 = 17500s \approx 5hr$
- Will this image look really good for our 5hrs?
- If we shoot 9 rays/pixel that gives us $45hr$.

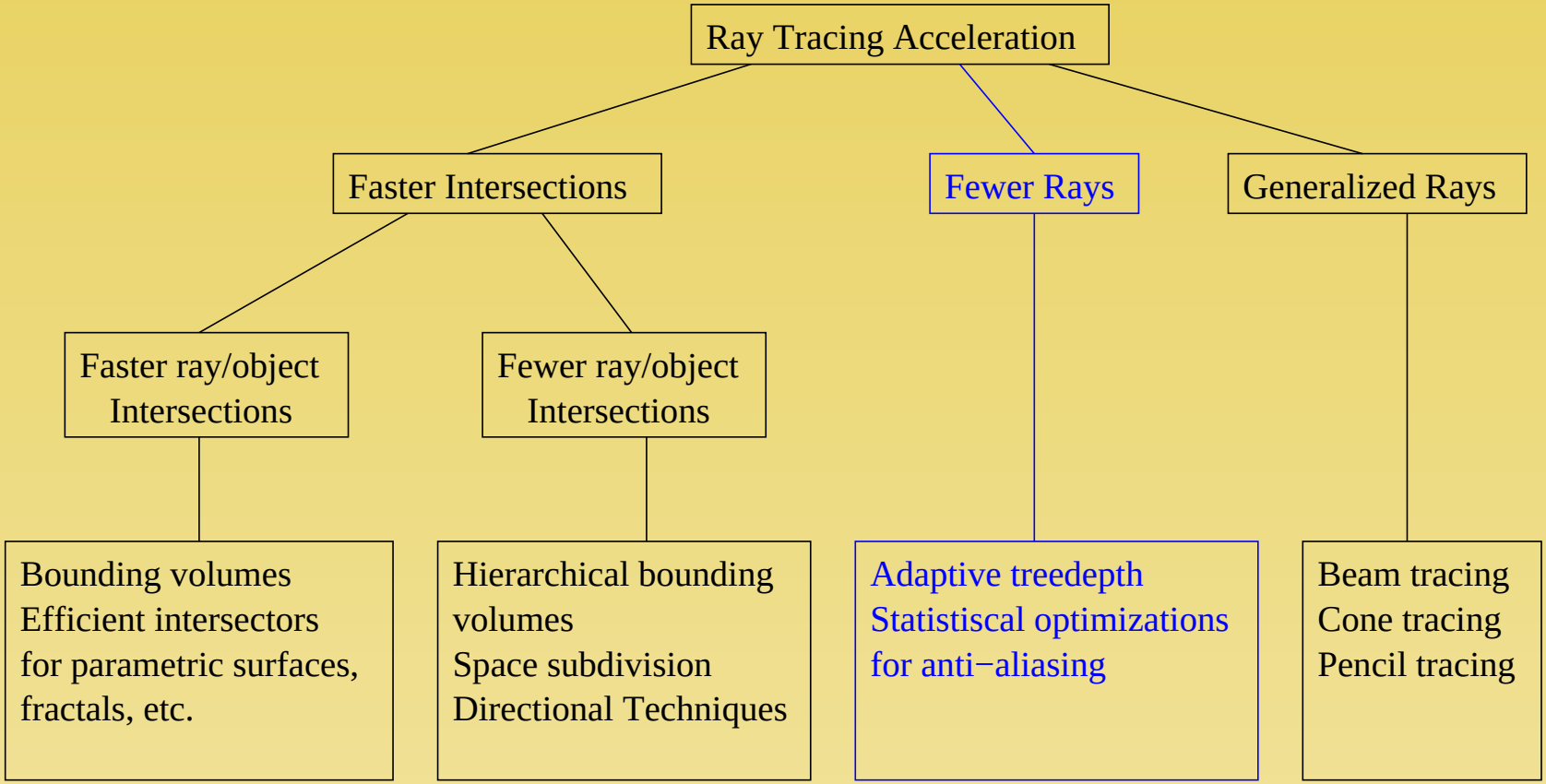
Ray Genealogy

- 1 Ray/pixel at $1k \times 1k$ image $= 1M$ rays.
- Say on average 6 secondary rays.$\rightarrow 7M$ rays
- $100k$ objects; 10 ops/intersection $\rightarrow 7,000,000M$ ops
- 2GHz processor 5 cycles per op $\rightarrow 400MFLOPS$
- How long to render? $70000/4 = 17500s \approx 5hr$
- Will this image look really good for our 5hrs?
- If we shoot 9 rays/pixel that gives us $45hr$.
- So where are we spending the time?

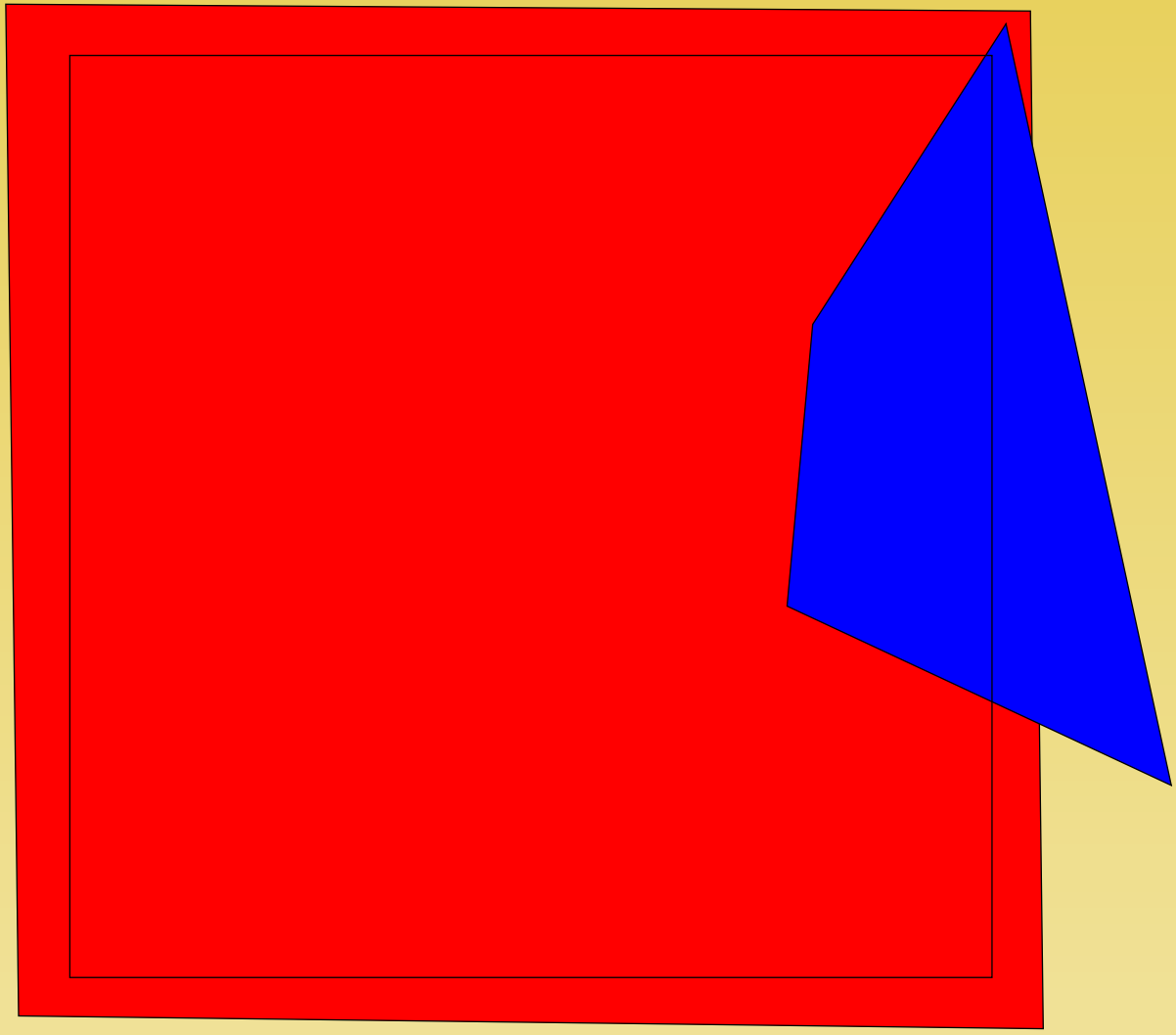
Ray Tracing Acceleration Classification



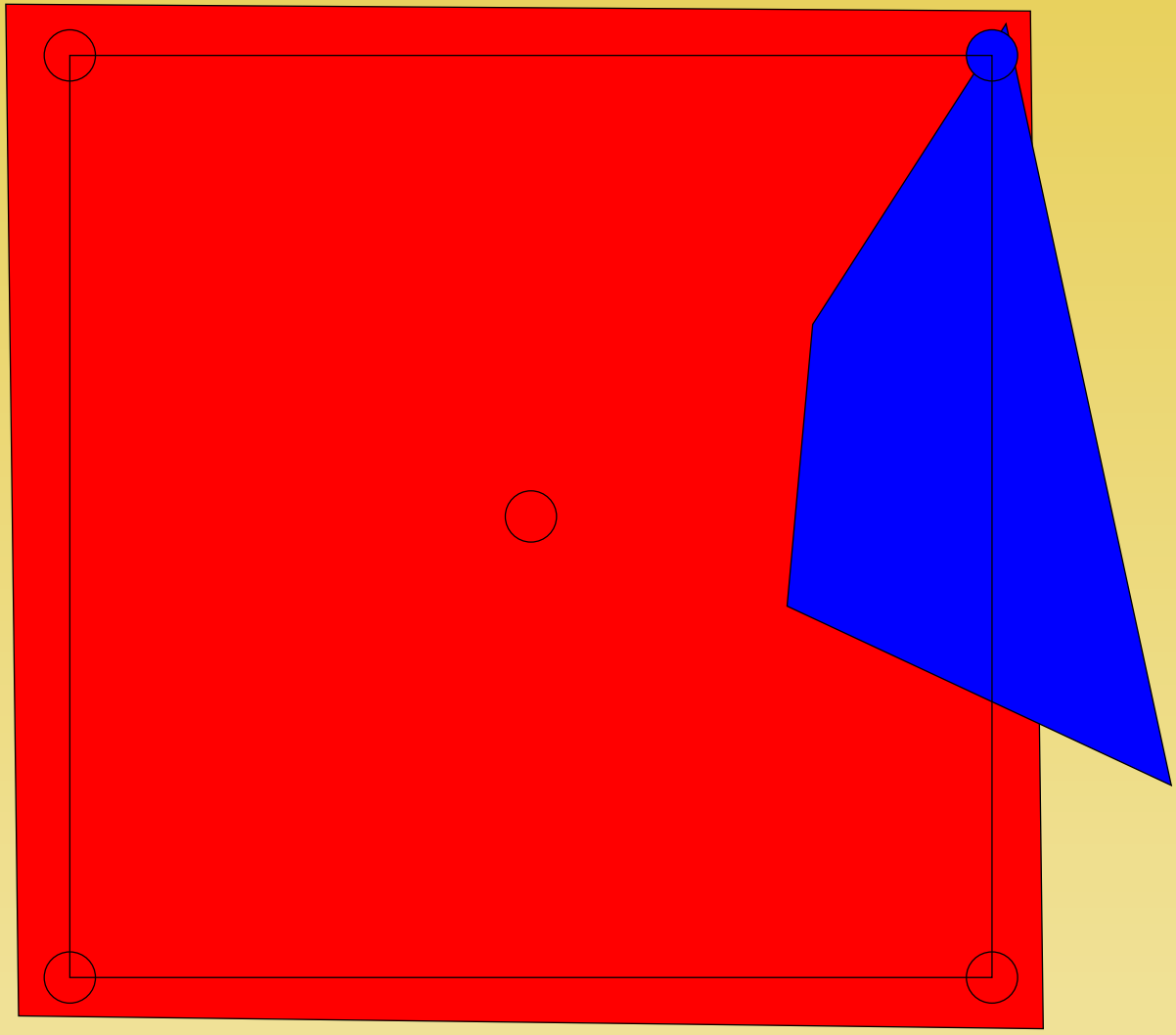
Ray Tracing Acceleration Classification



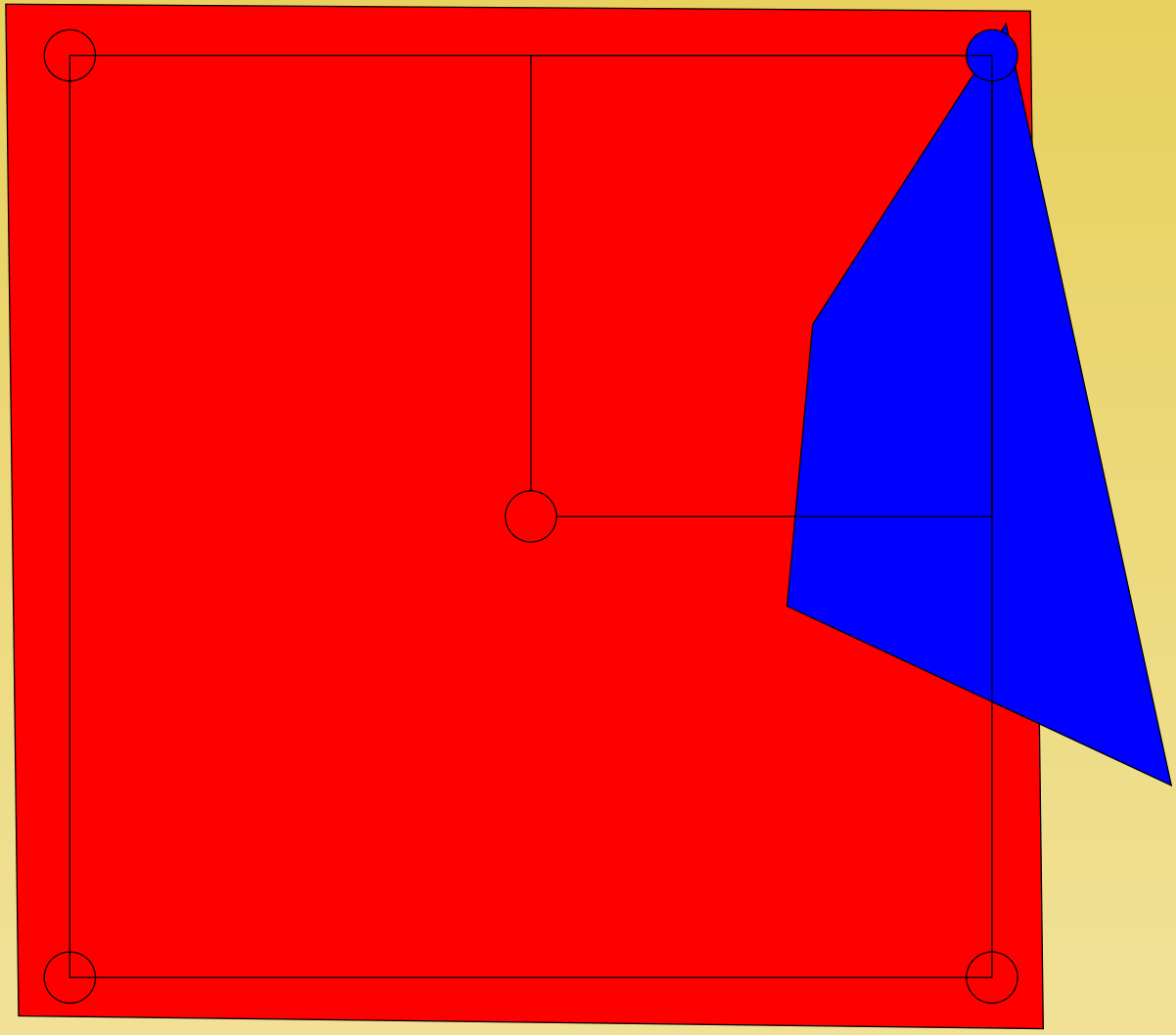
Adaptive Super-sampling



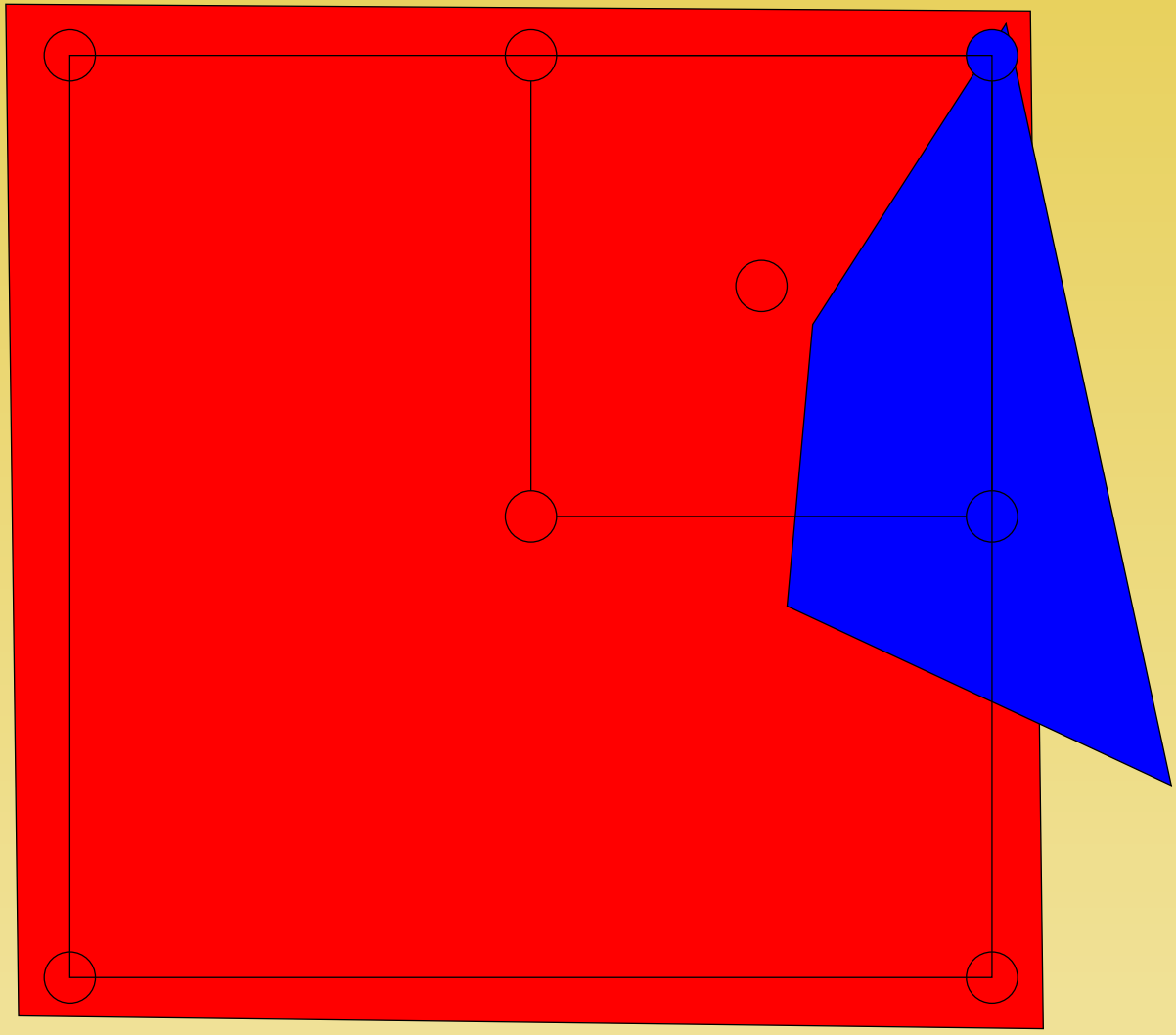
Adaptive Super-sampling



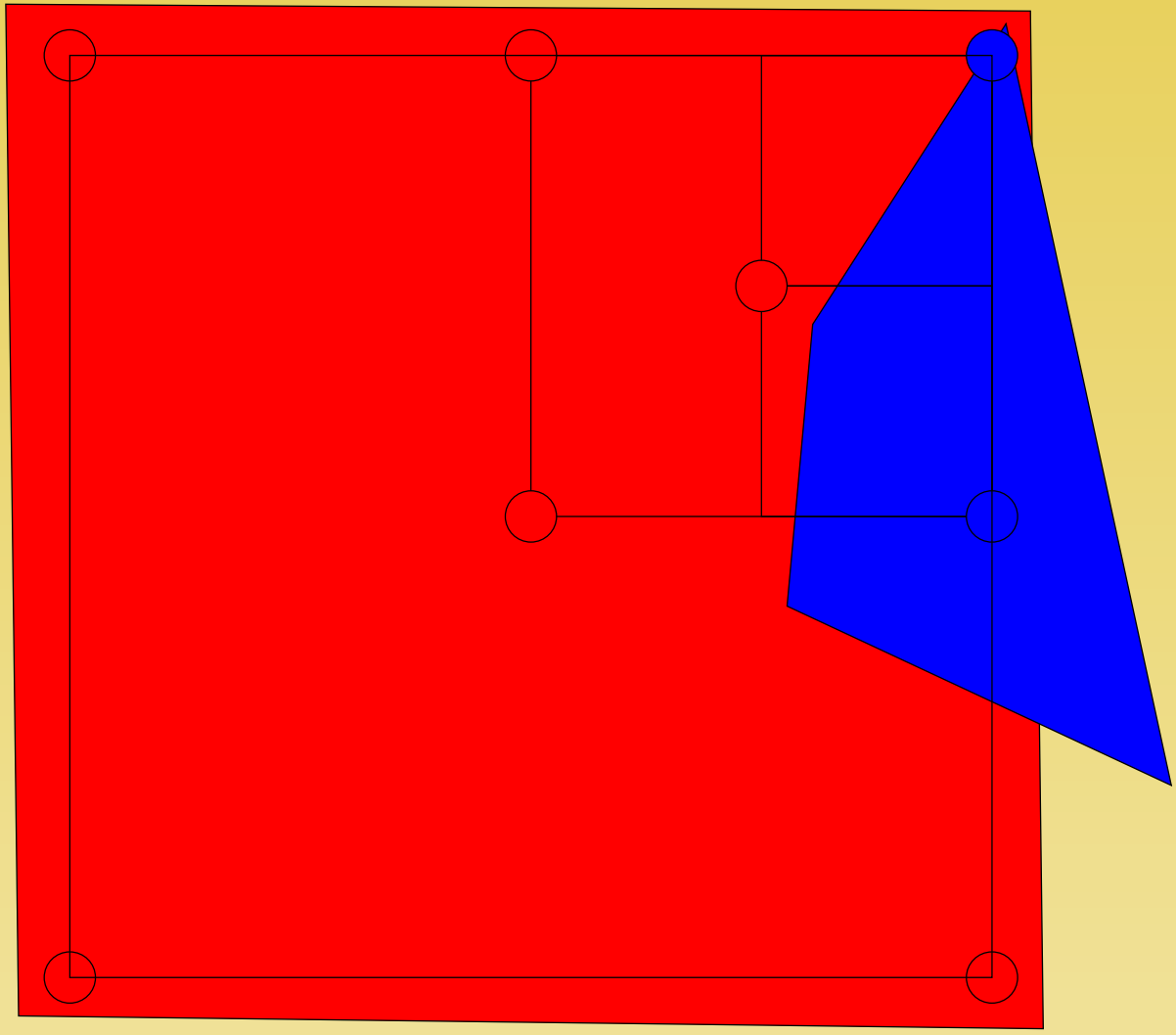
Adaptive Super-sampling



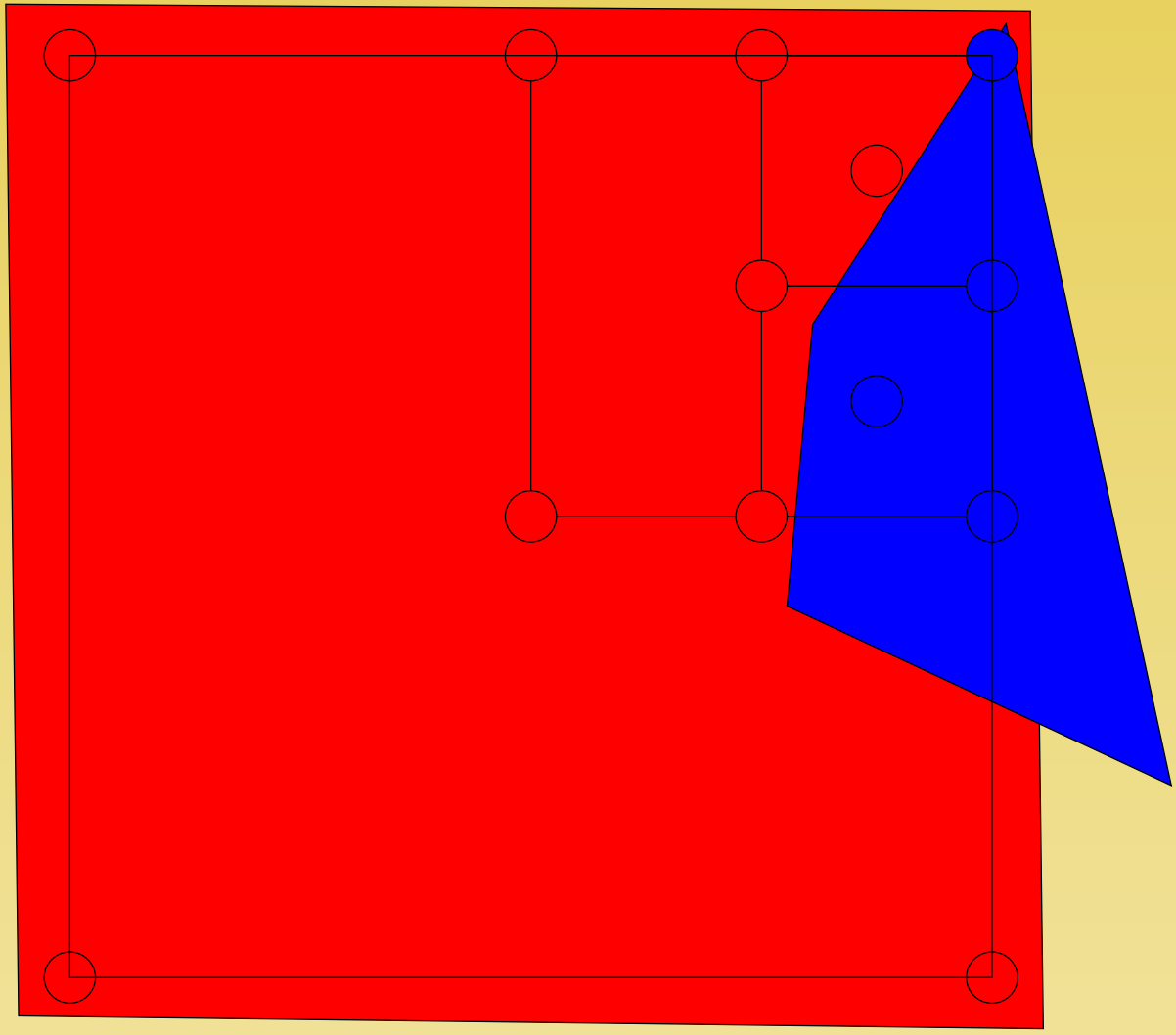
Adaptive Super-sampling



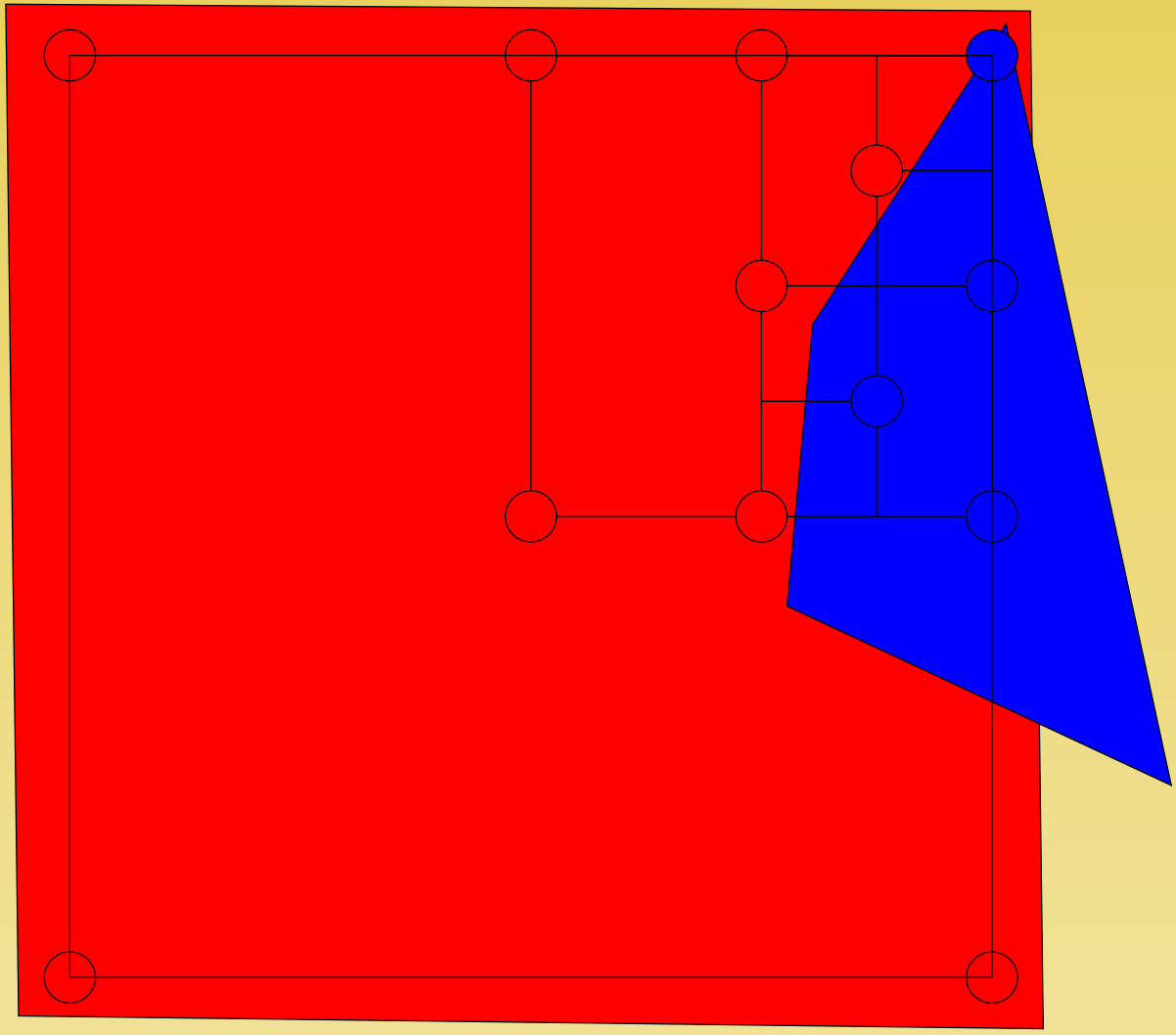
Adaptive Super-sampling



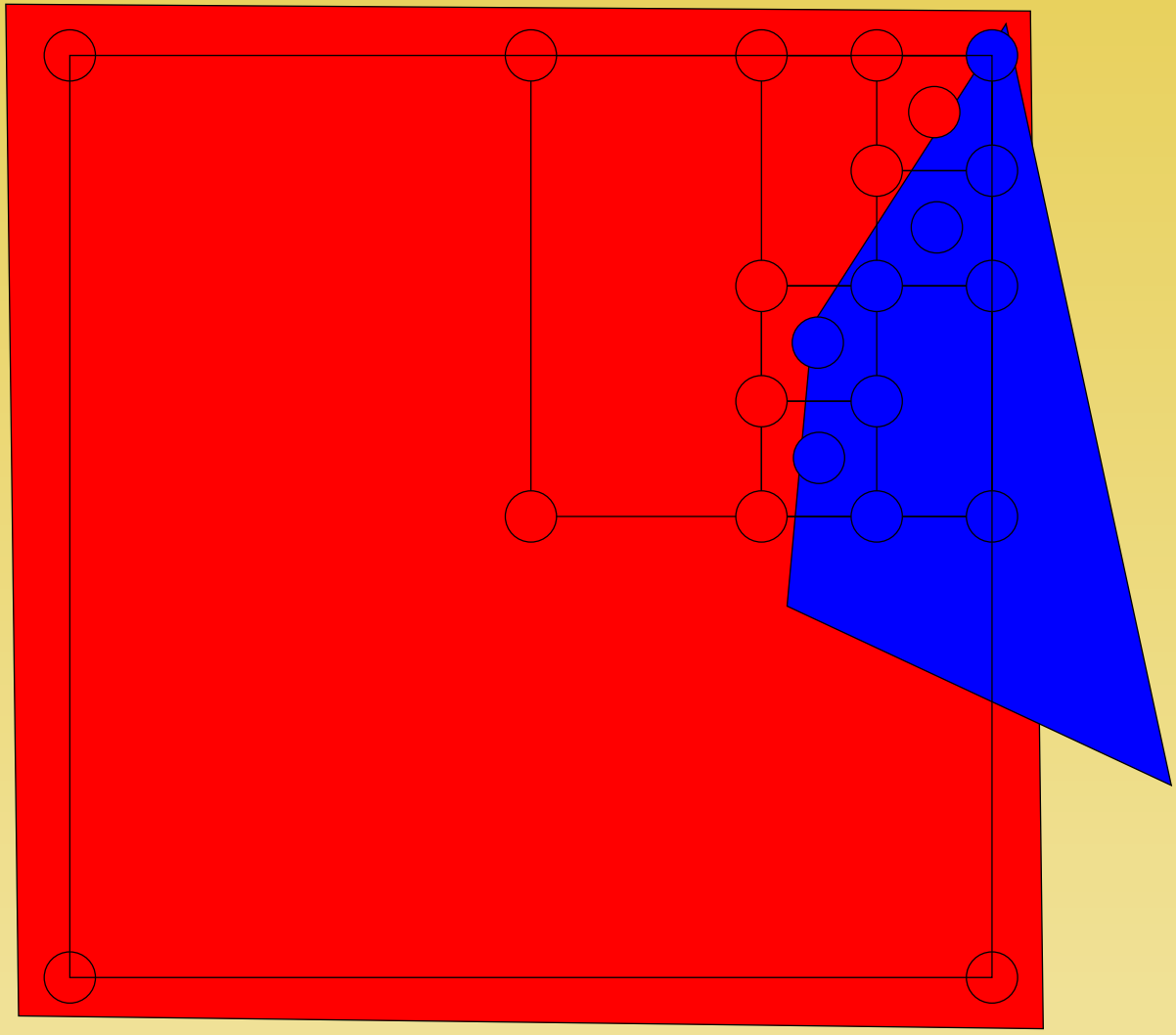
Adaptive Super-sampling



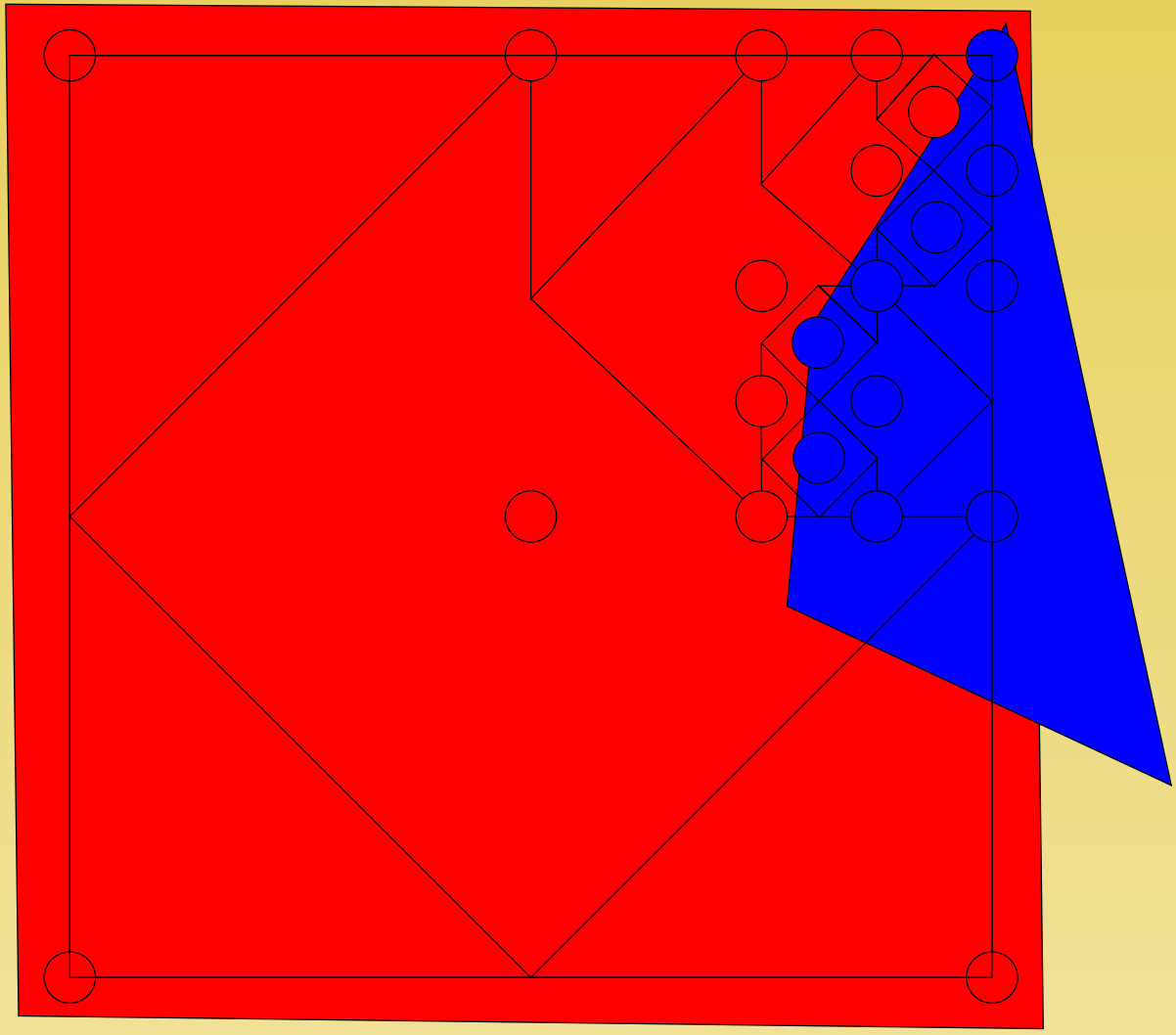
Adaptive Super-sampling



Adaptive Super-sampling



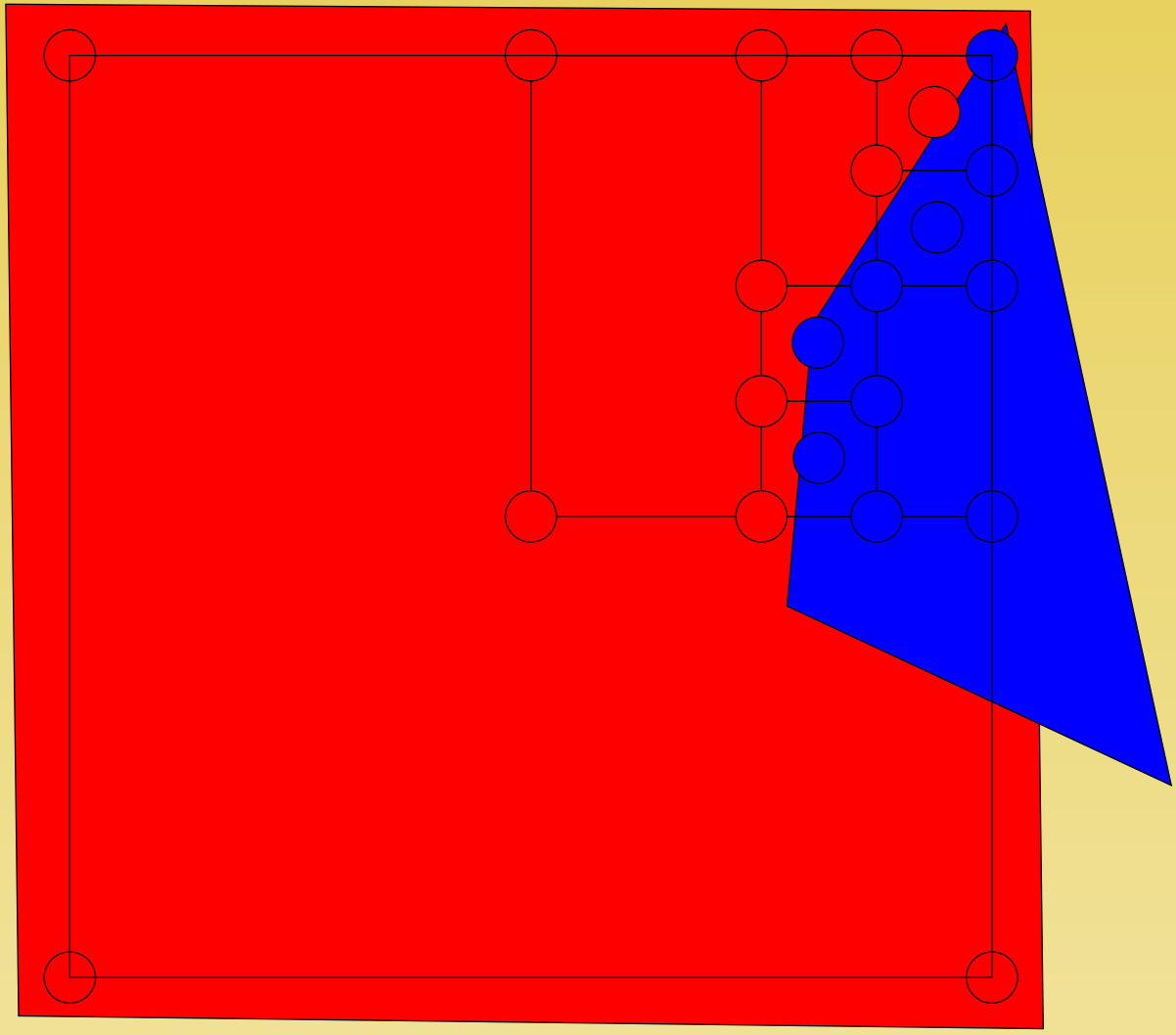
Adaptive Super-sampling



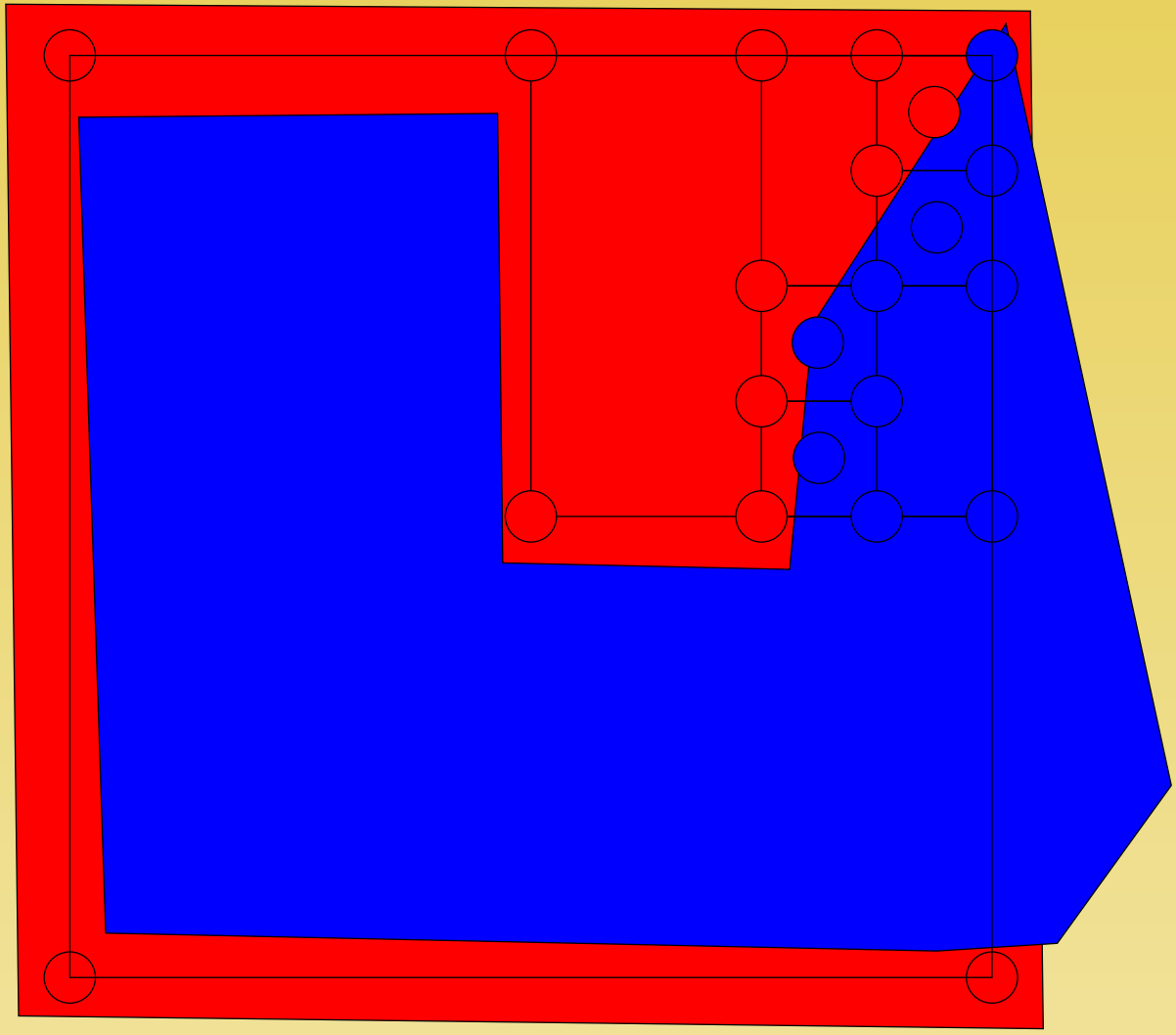
Adaptive Super-sampling

- Tries to only fire rays that are needed.
- But the assumption that if the rays are the same that the correct pixel value has been determined is wrong.

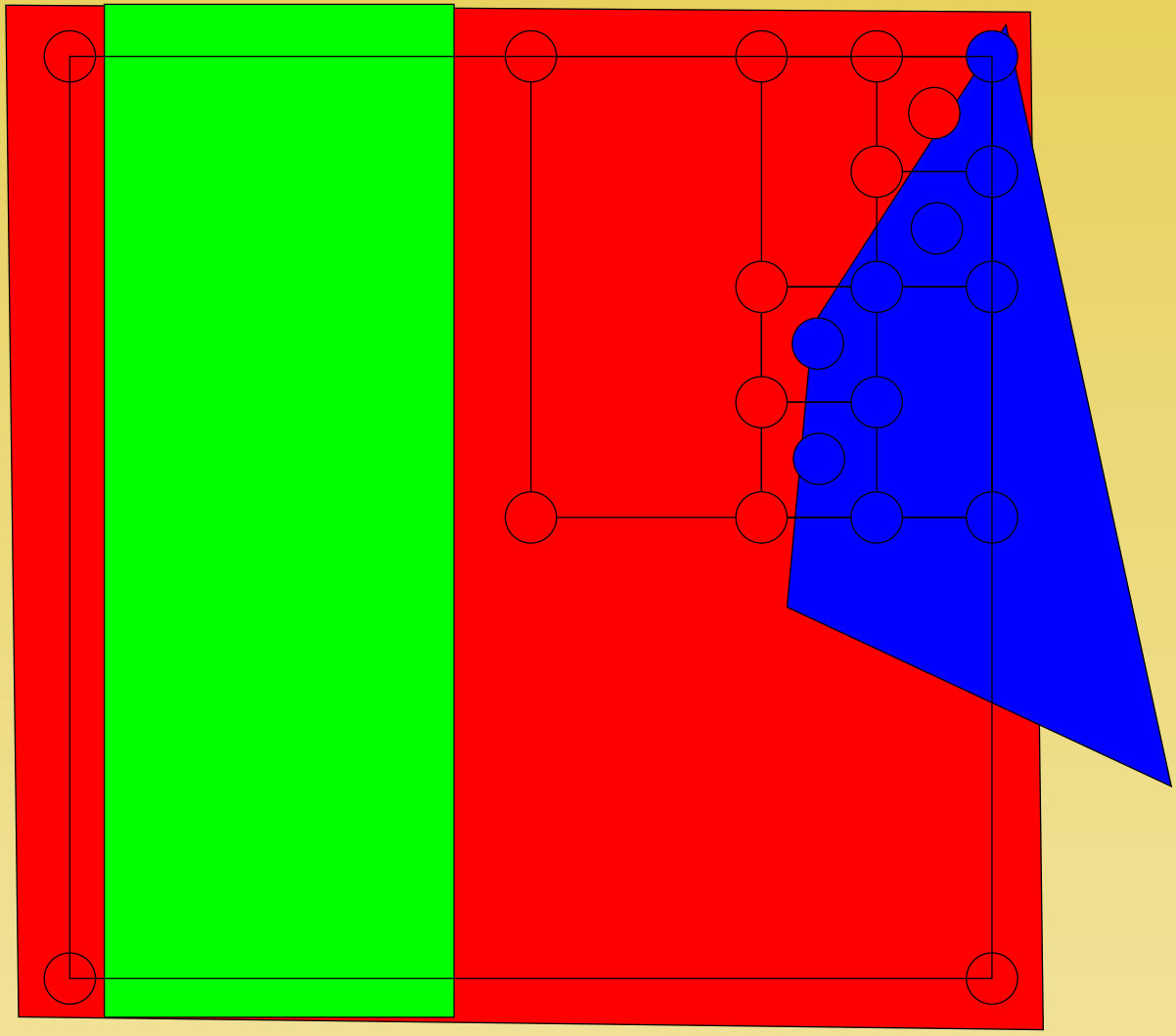
Adaptive Super-sampling



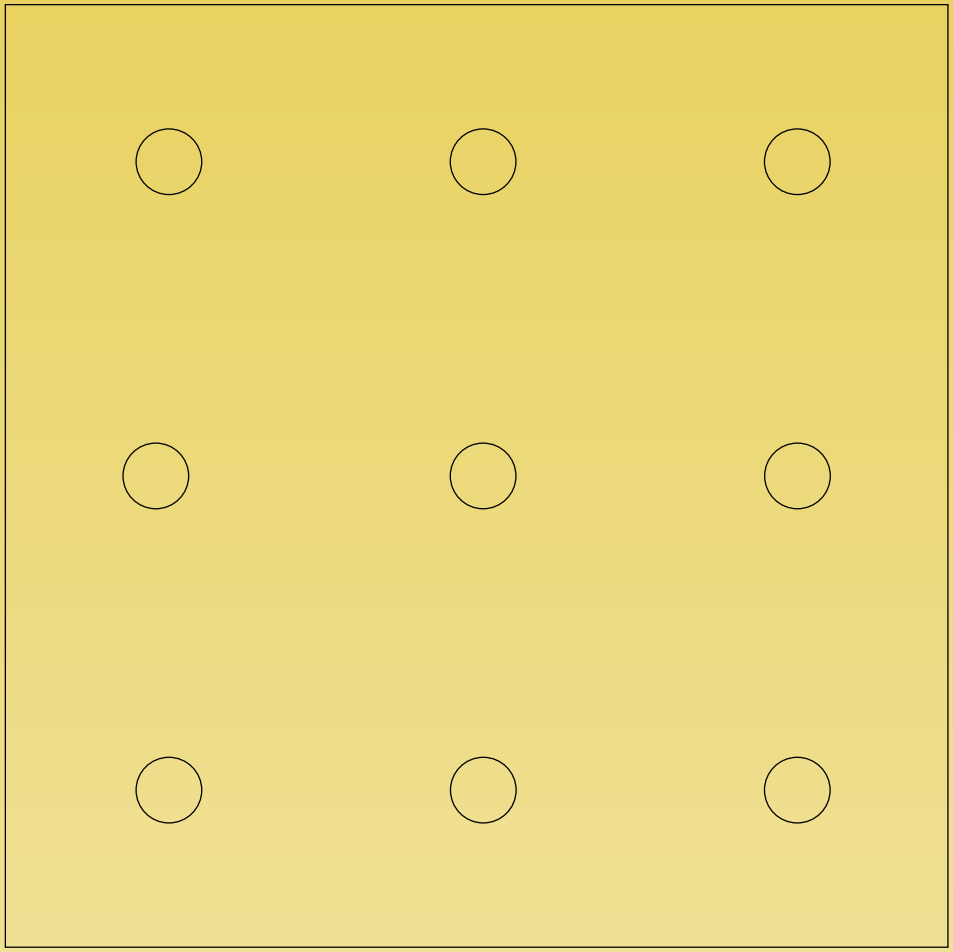
Adaptive Super-sampling



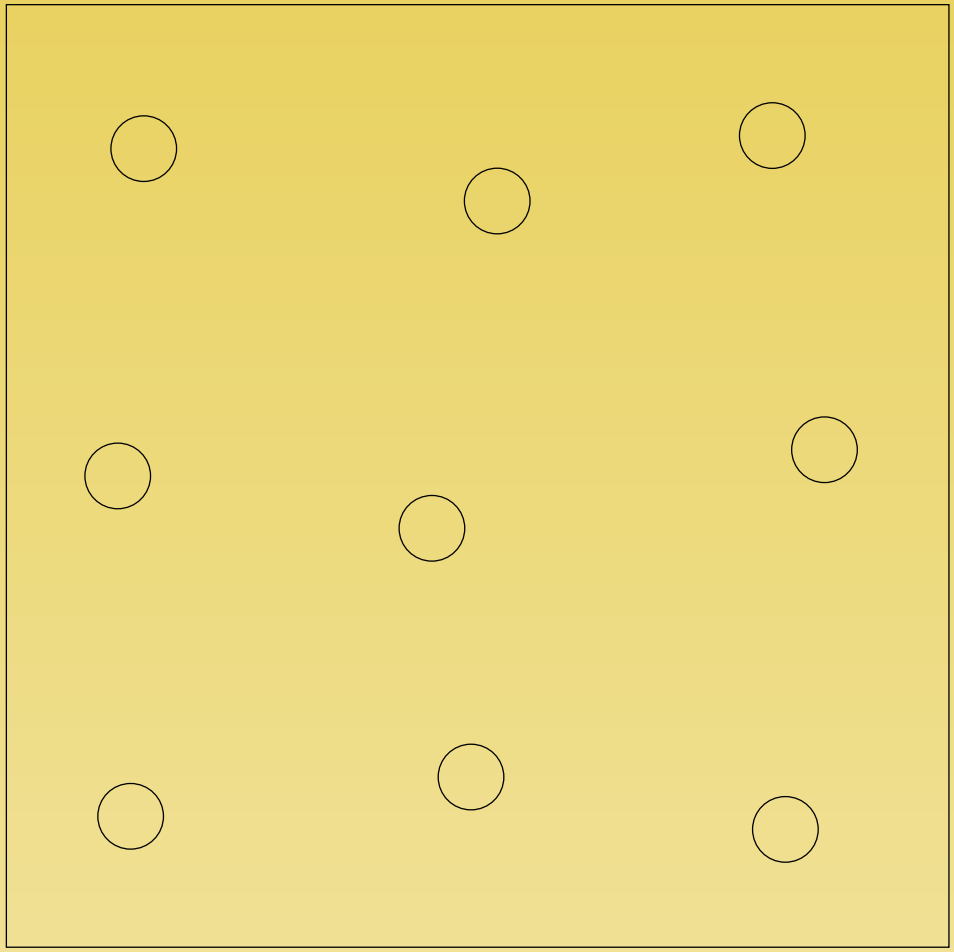
Adaptive Super-sampling



Distributed/Stochastic Ray Tracing



Distributed/Stochastic Ray Tracing



Motion blur

Cook, Porter, Carpenter 1984

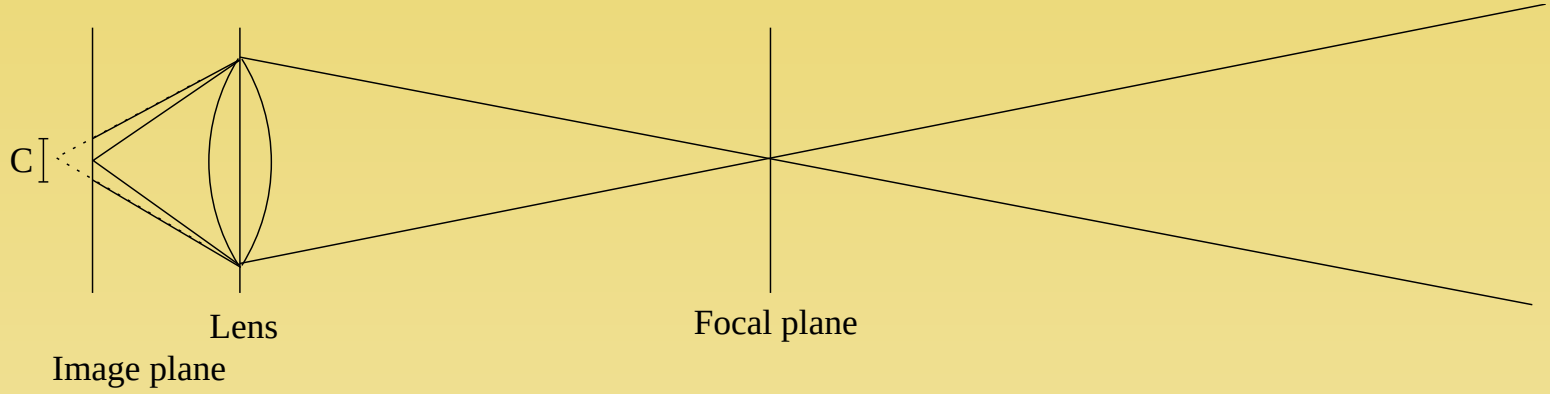
motion blur

- Sample the image temporally; temporal anti-aliasing.
- Can perform with other rendering techniques.
- Modeling the shutter of a camera.
- In ray tracing, can combine with spatial anti-aliasing by giving each ray a jittered time.

Cook, Porter, Carpenter 1984

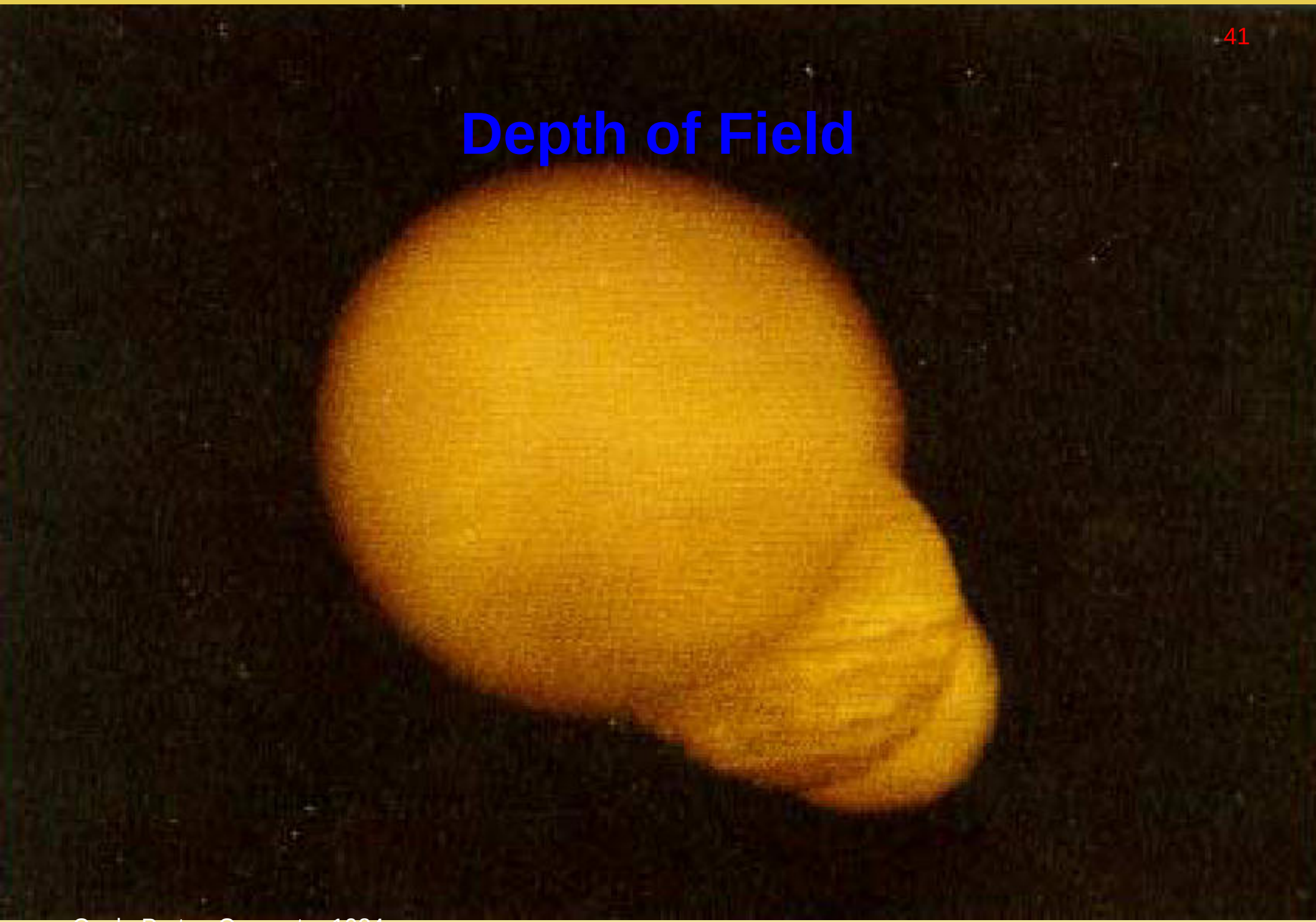
Depth of field

- Model the lens of a camera.
- All points in the scene project as a circle on the image plane, called the *circle of confusion*. Objects at the focal distance are sharp, others are blurred.



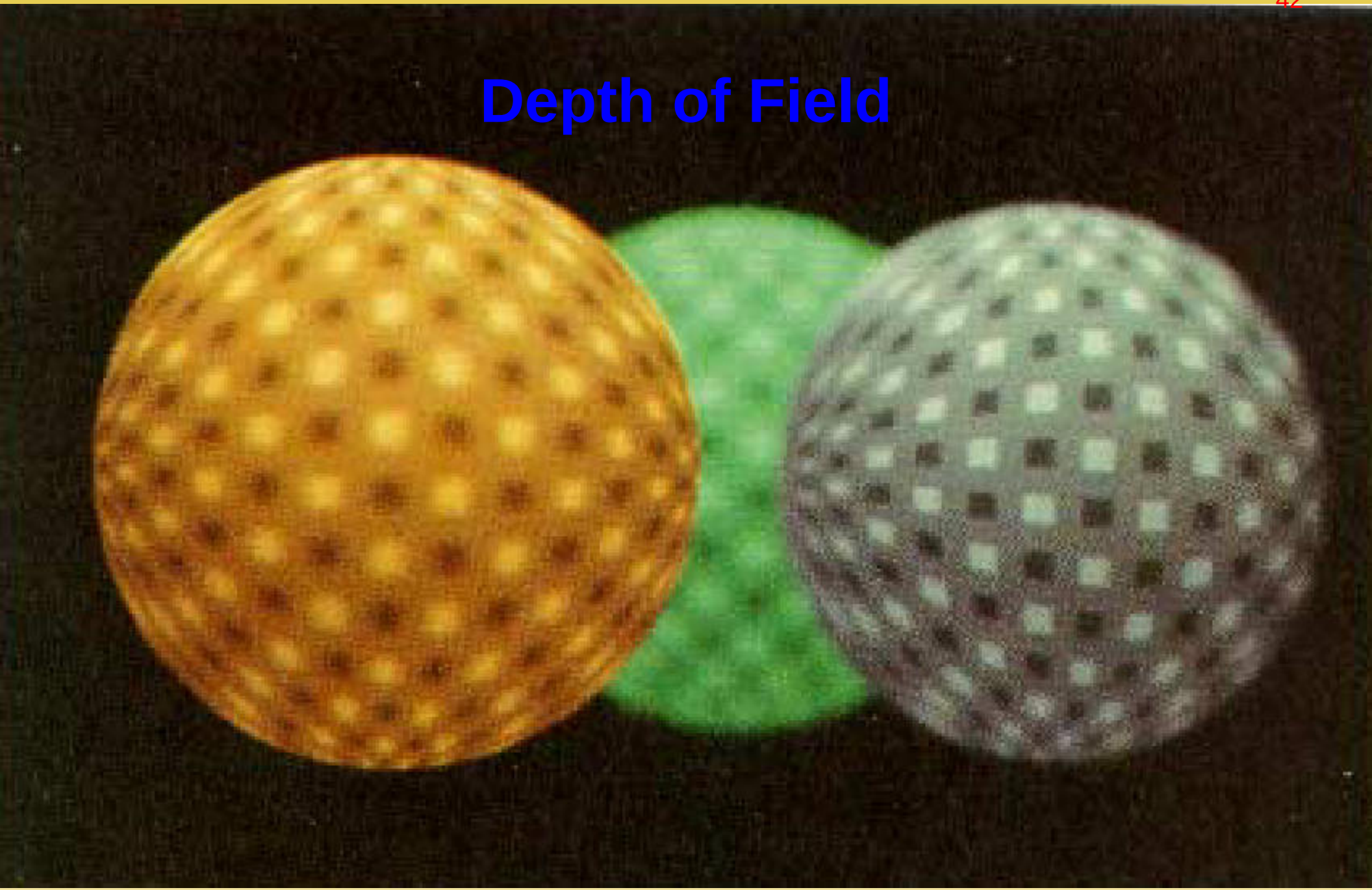
Cook, Porter, Carpenter 1984

Depth of Field



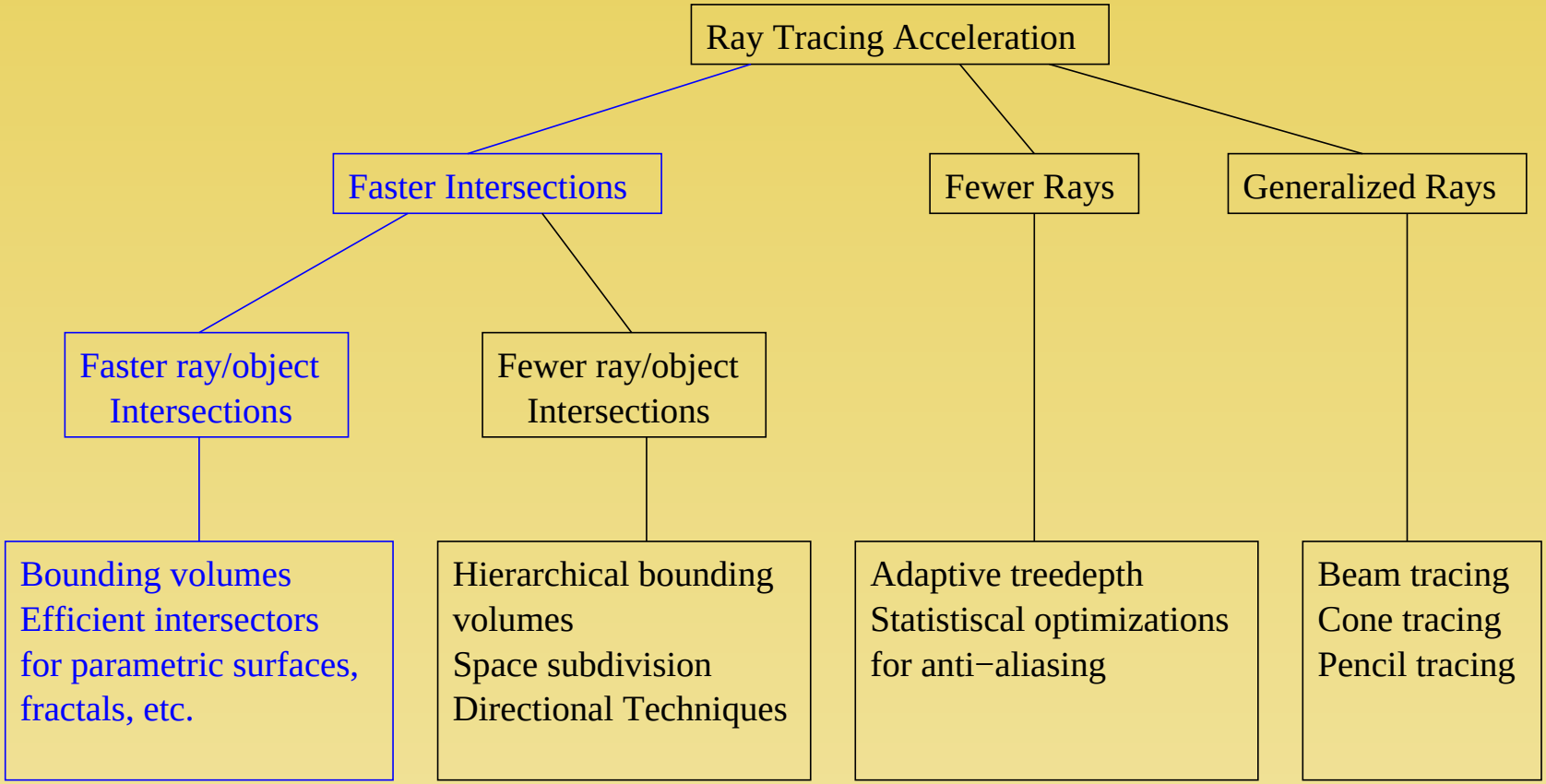
Cook, Porter, Carpenter 1984

Depth of Field



Cook, Porter, Carpenter 1984

Ray Tracing Acceleration Classification



Bounding Volumes

- How does the number of intersection tests compare to the number of primitives?

Bounding Volumes

- How does the number of intersection tests compare to the number of primitives?
- How many times will a ray hit a sphere?

Bounding Volumes

- How does the number of intersection tests compare to the number of primitives?
- How many times will a ray hit a sphere?
- How many times will a ray hit a sphere made of 40 triangles?

Bounding Volumes

- How does the number of intersection tests compare to the number of primitives?
- How many times will a ray hit a sphere?
- How many times will a ray hit a sphere made of 40 triangles?
- How many times will a ray hit a sphere made of 4000 triangles?

Bounding Volumes

- Enclose objects/primitives inside volume with simpler intersection test.
- For objects that are intersected do we have an increase or decrease in computation?

Bounding Volumes

- Enclose objects/primitives inside volume with simpler intersection test.
- For objects that are intersected do we have an increase or decrease in computation?
- For objects away from ray do we have an increase or decrease in calculations?

Bounding Volumes

- Enclose objects/primitives inside volume with simpler intersection test.
- For objects that are intersected do we have an increase or decrease in computation?
- For objects away from ray do we have an increase or decrease in calculations?
- Which case happens more often?

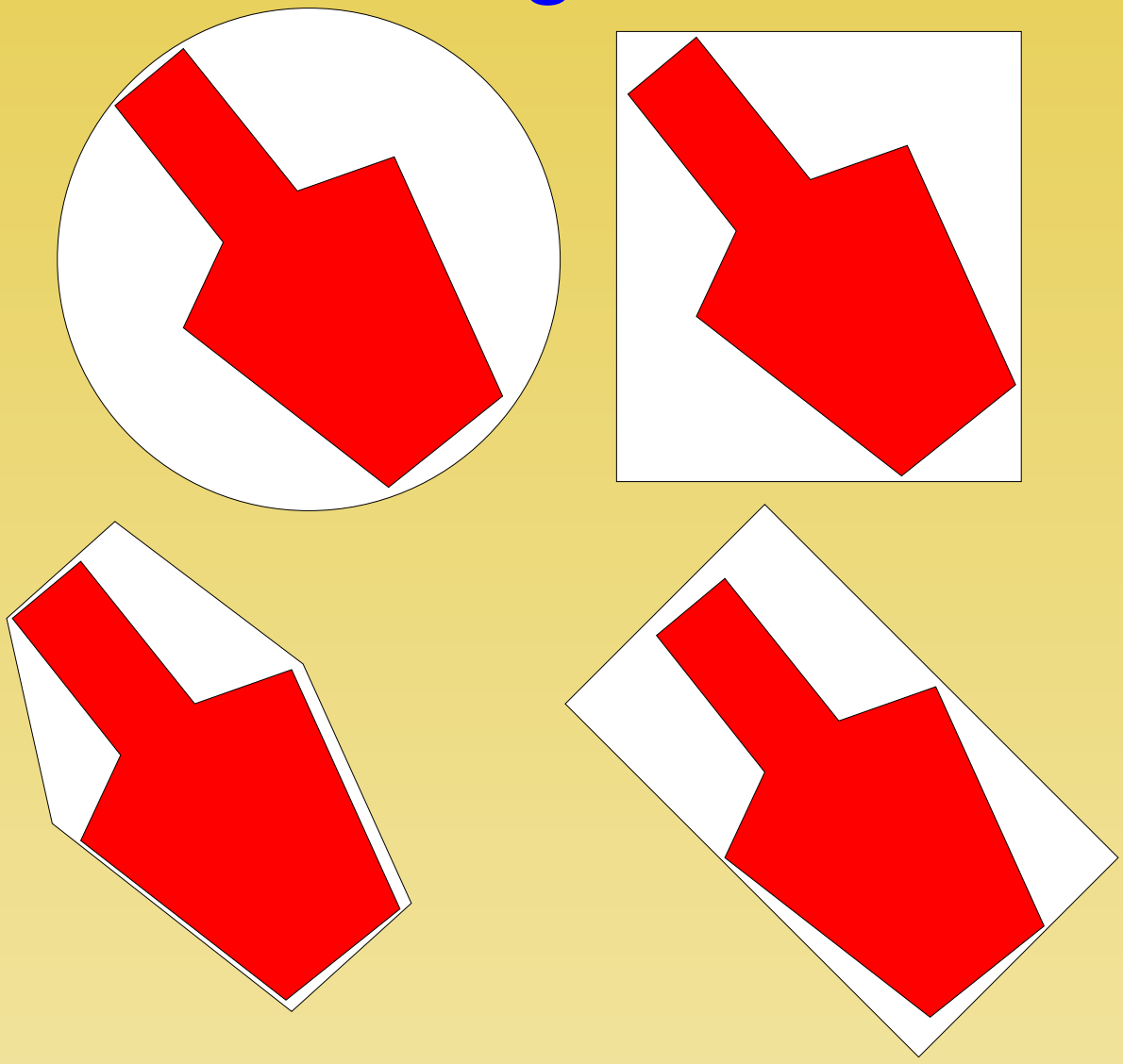
Bounding Volumes

- Spheres
- Axes aligned boxes
- Boxes (parallelepipeds)
- Slabs (pairs of parallel planes)

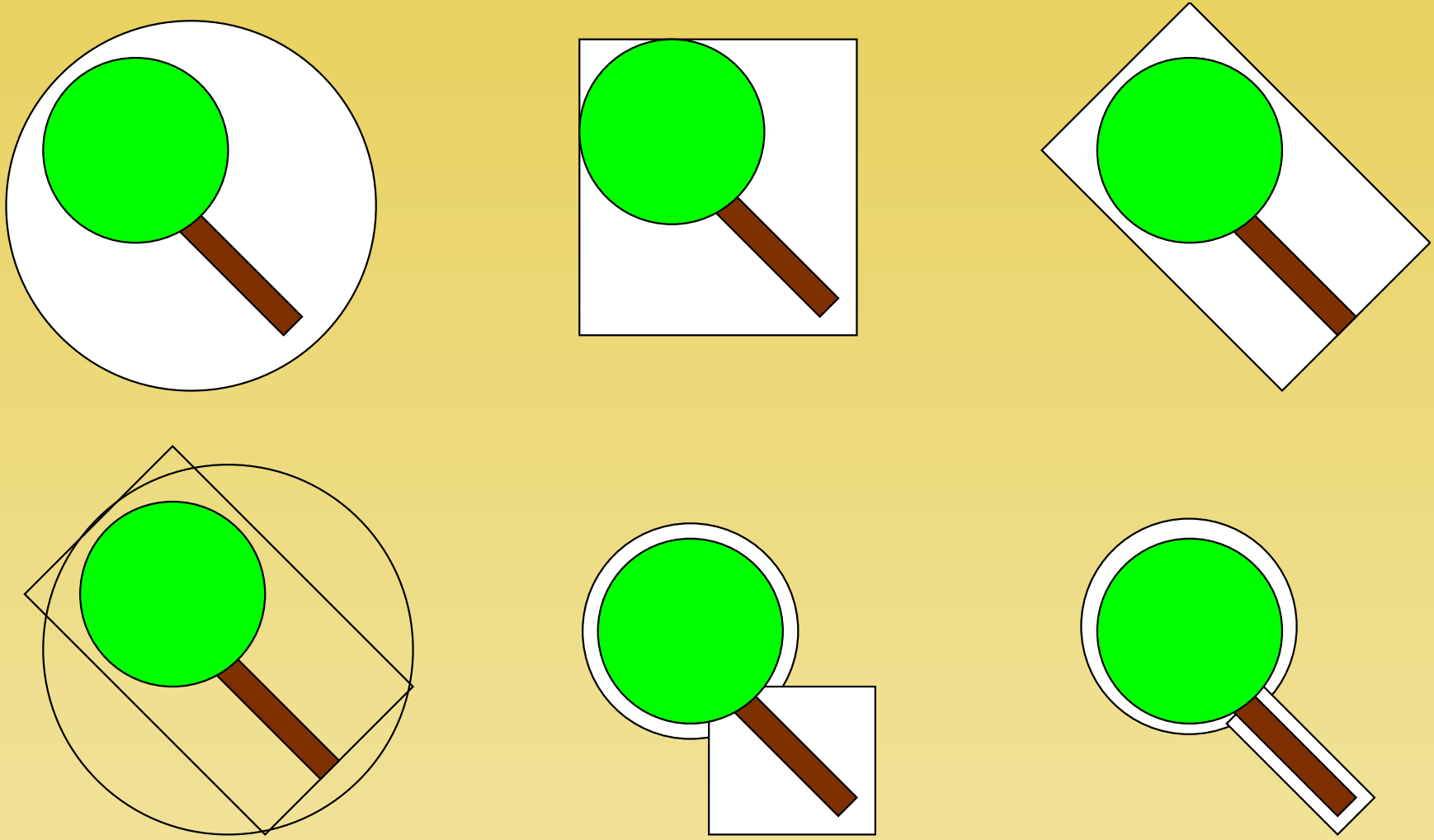
$$Cost = n * B + m * I$$

where n - number of rays,
m - number of rays that intersect bounding volume,
I cost of intersecting object within,
B cost of intersecting bounding volume.

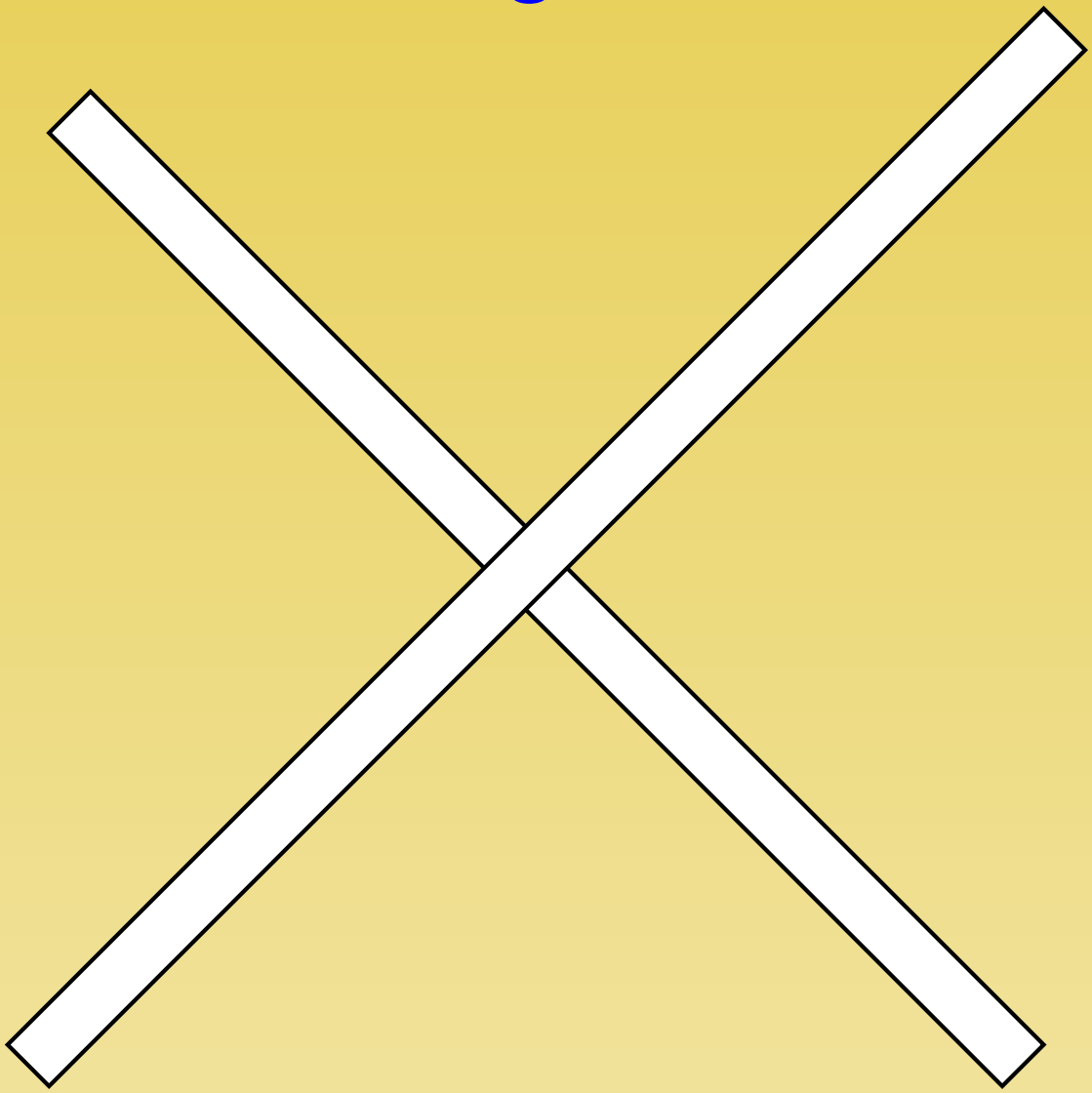
Bounding Volumes



Bounding Volumes



Bounding Volumes



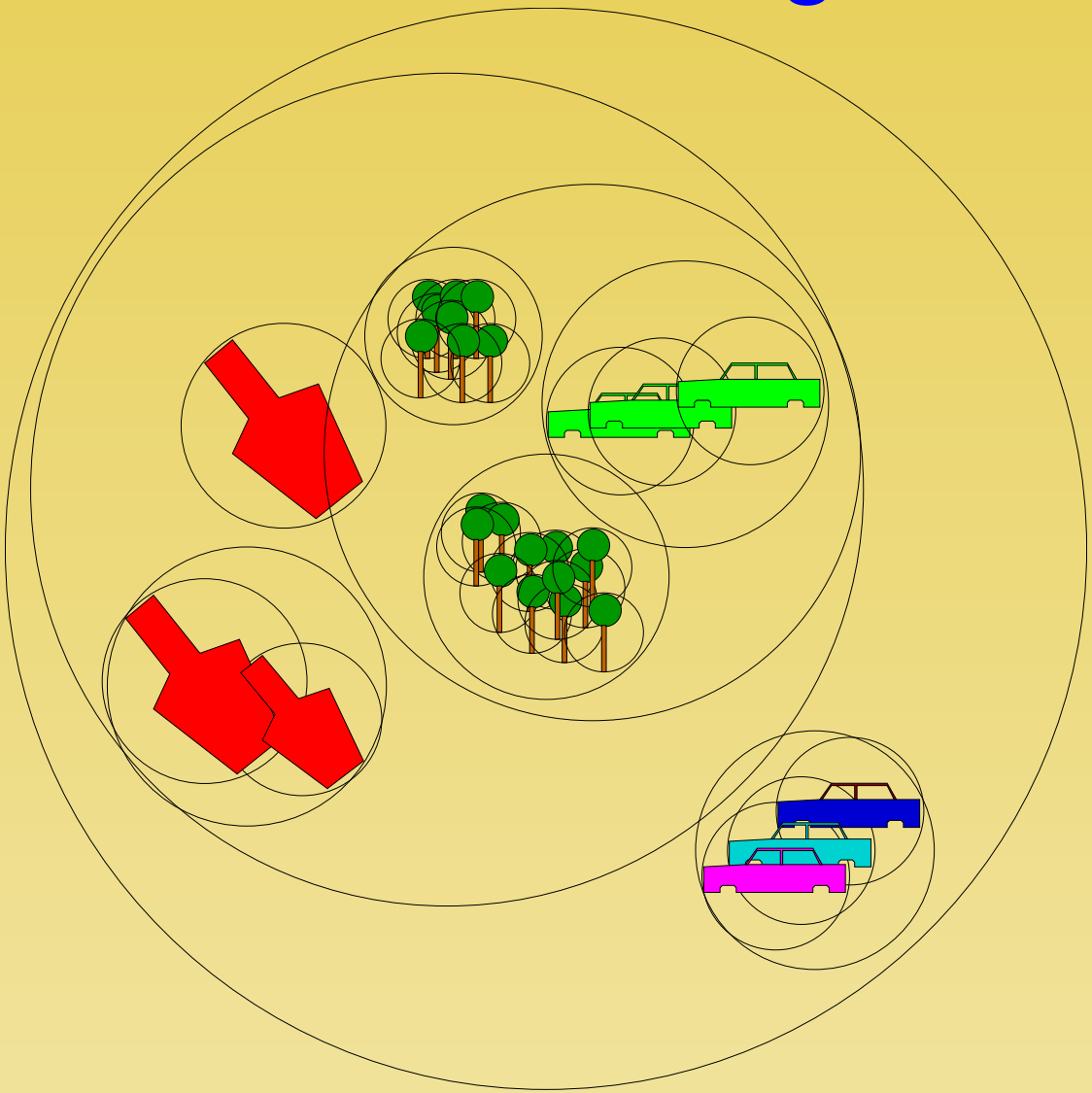
Bounding Volumes

- Trade off complexity versus closeness of fit.
- Transformed bounding volumes.
- Intersection of bounding volumes.
- Union of bounding volumes.

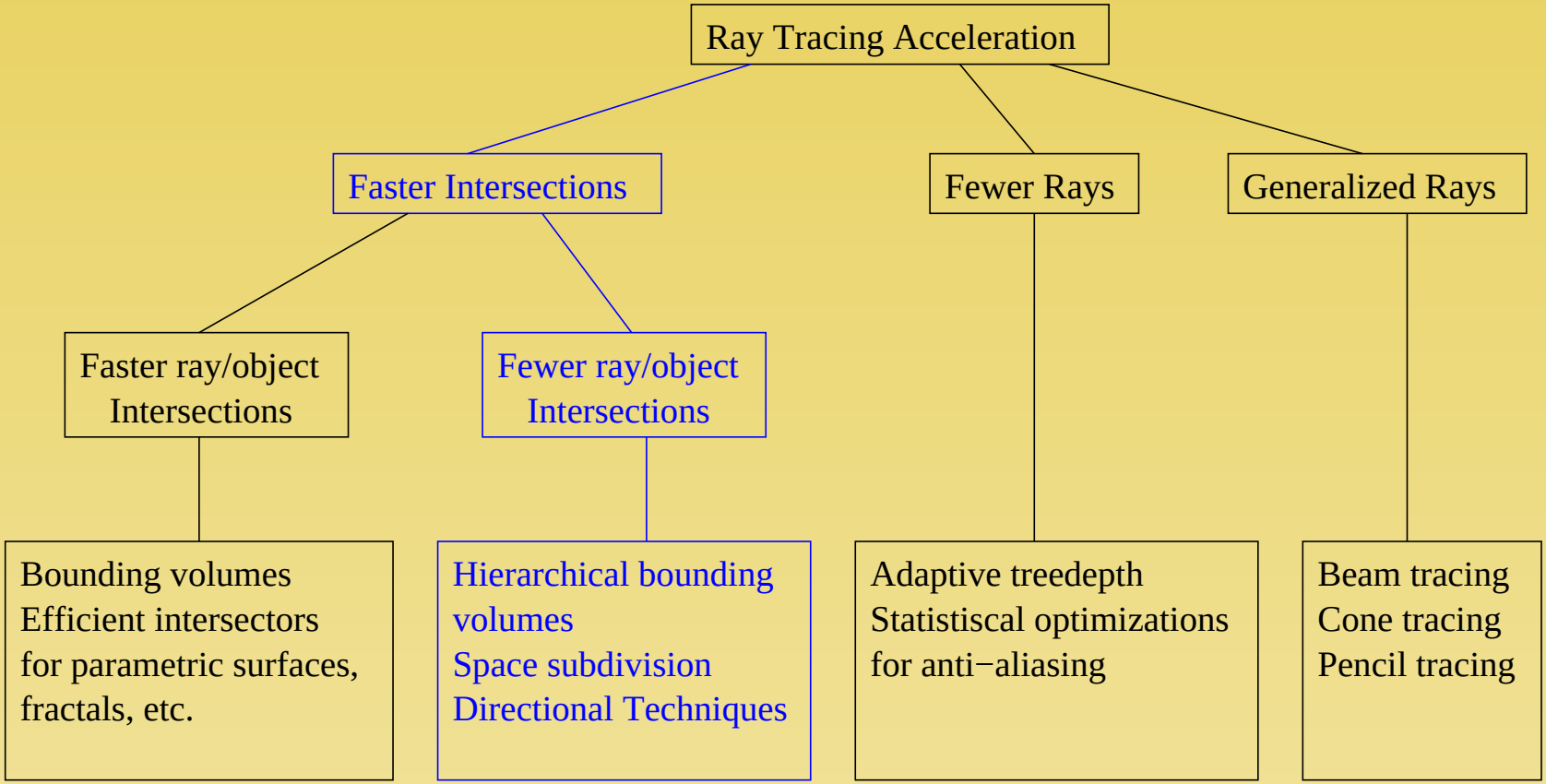
Bounding Volumes

- Trade off complexity versus closeness of fit.
- Transformed bounding volumes.
- Intersection of bounding volumes.
- Union of bounding volumes.
- What about hierarchies?

Hierarchical Bounding Volumes



Ray Tracing Acceleration Classification



Bounding Volumes: Why Hierarchies

- What is the complexity of the intersection tests?

Bounding Volumes: Why Hierarchies

- What is the complexity of the intersection tests?
- What if you create a tree-like bounding volumes?

Bounding Volumes: Why Hierarchies

- What is the complexity of the intersection tests?
- What if you create a tree-like bounding volumes?
- Using hierarchies are theoretically $\mathcal{O}(\log(n))$.
- When enclosing several volumes with a new volume, the cost of doing the extra check must pay off.
- Top-down or bottom-up approach.
- Minimize volume/surface area.
- Hierarchies are not always simple to construct.

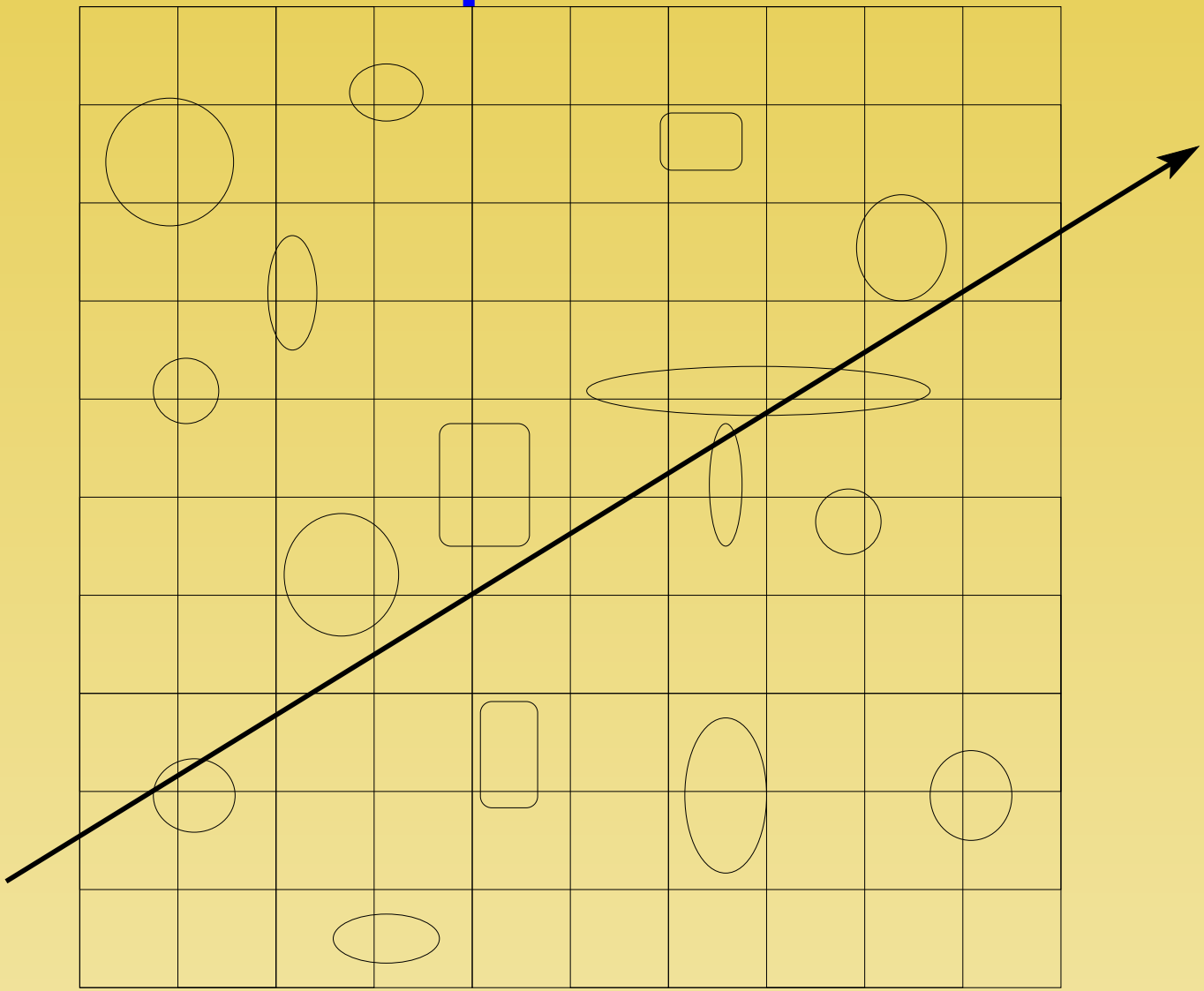
Bounding Volumes

- No correct volume for all cases. Often a combination is best.
- No automatic way to determine the best volumes, can do a good job though.
- Placement of volumes usually requires help for really good results.

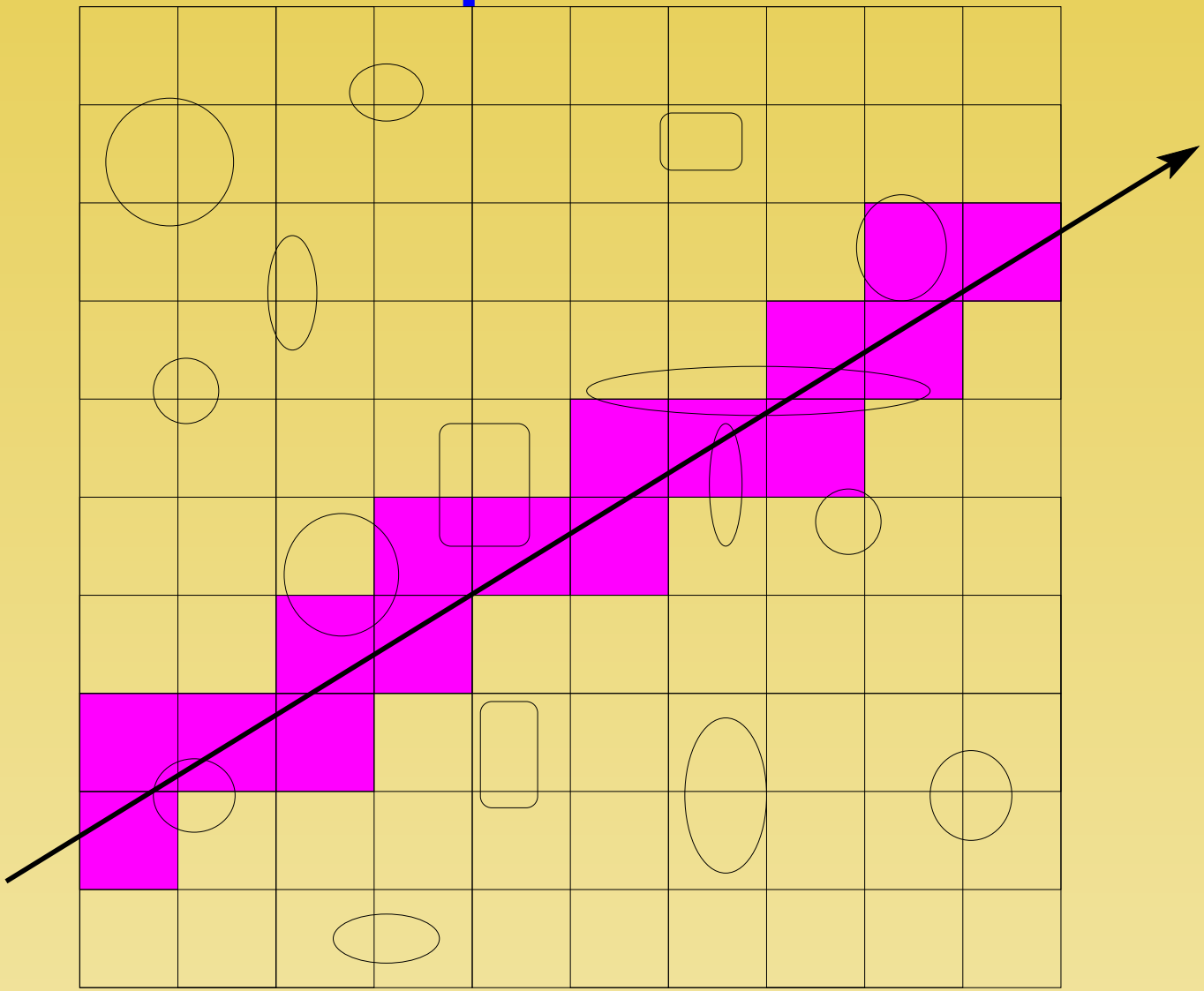
Spatial Subdivision

- Bounding volumes divide the space based on the objects.
- Instead lets just divide the space.
- Divide the space into *voxels*.

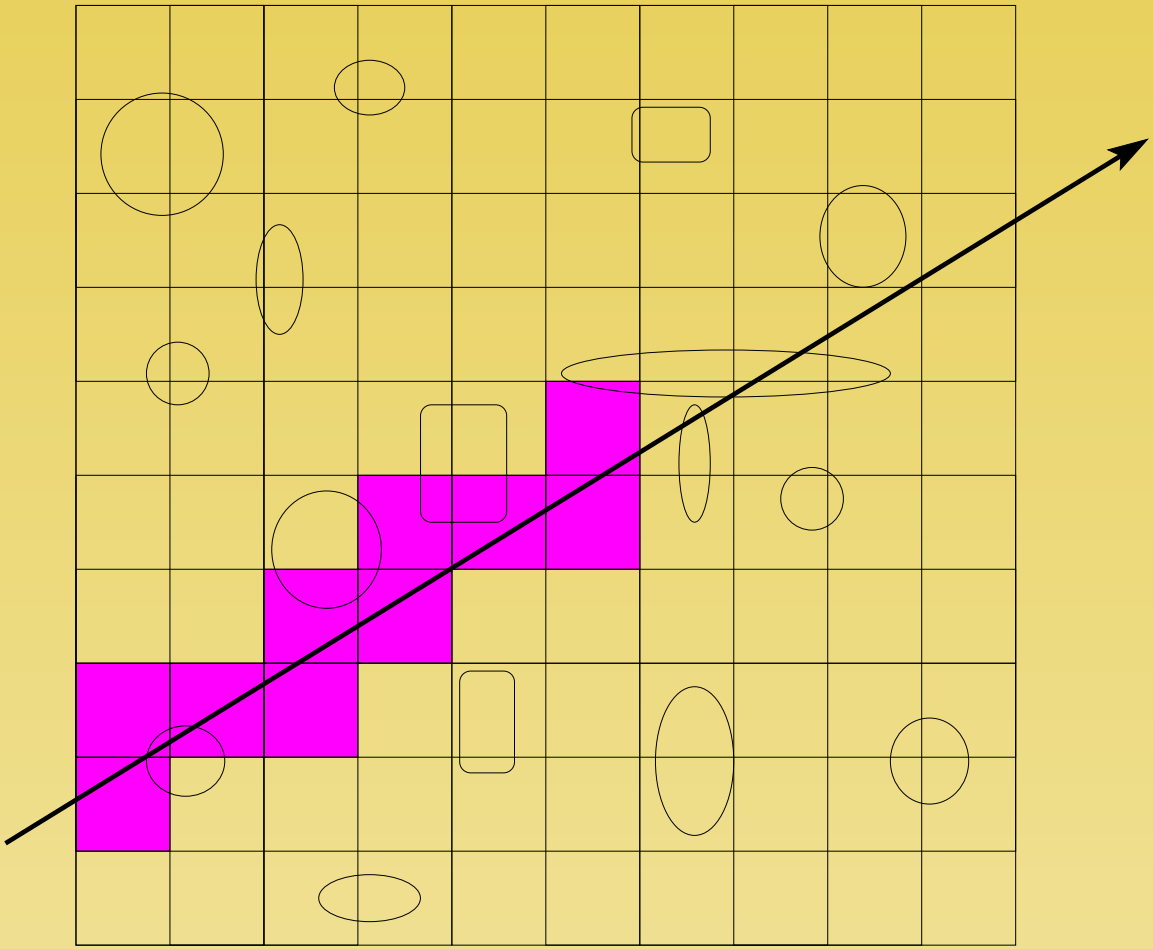
Uniform Space Subdivision



Uniform Space Subdivision

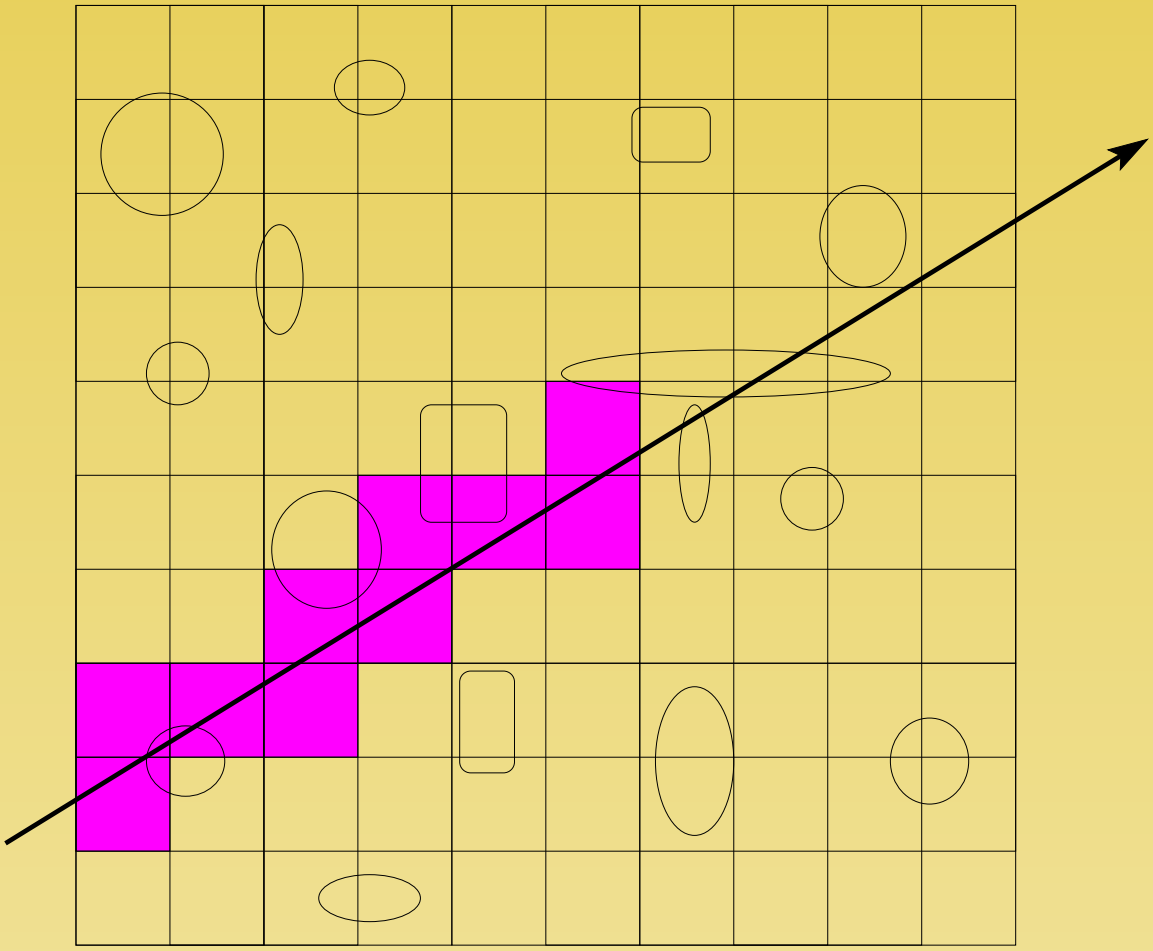


Uniform Space Subdivision



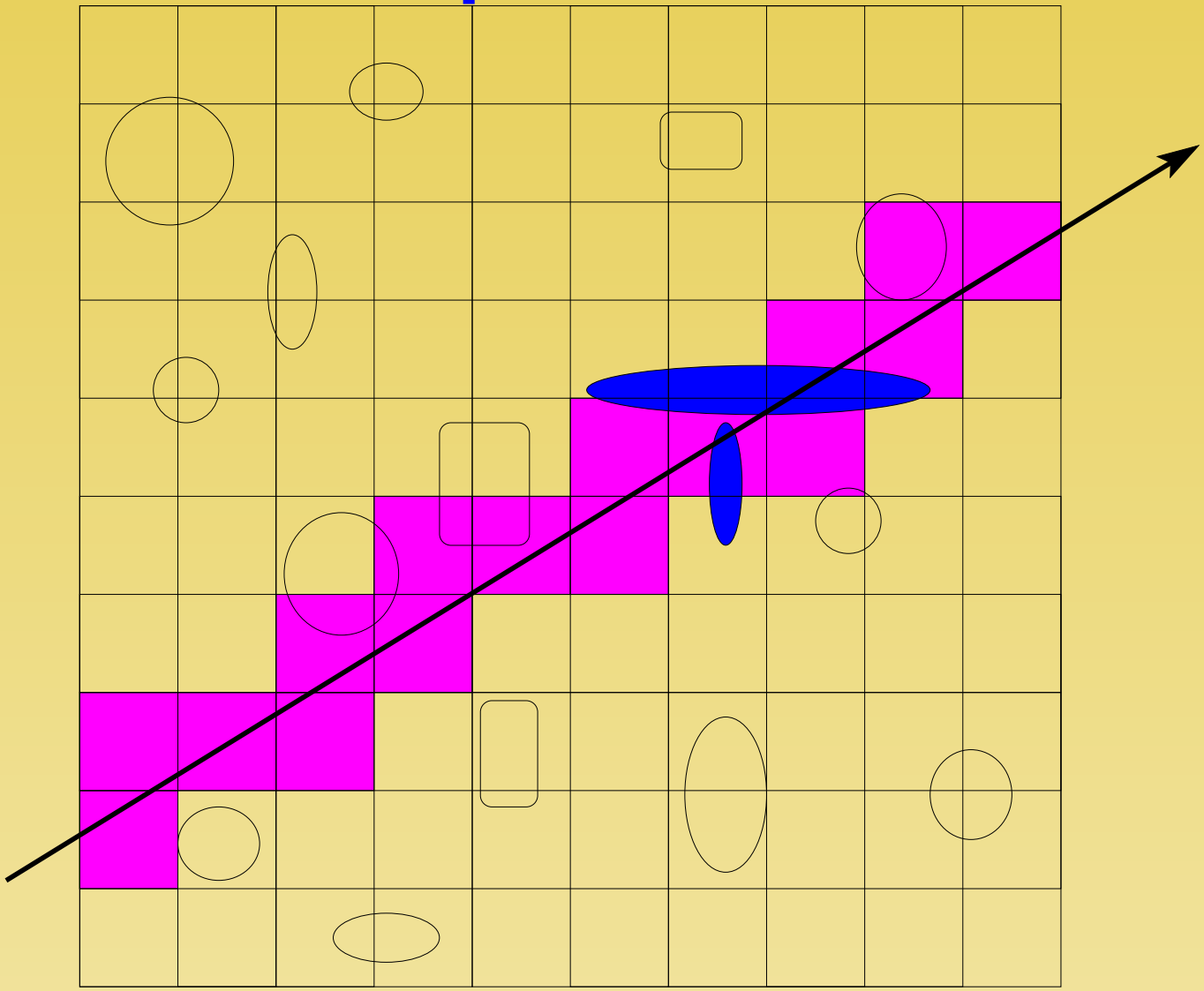
How do we determine the next voxel to test?

Uniform Space Subdivision

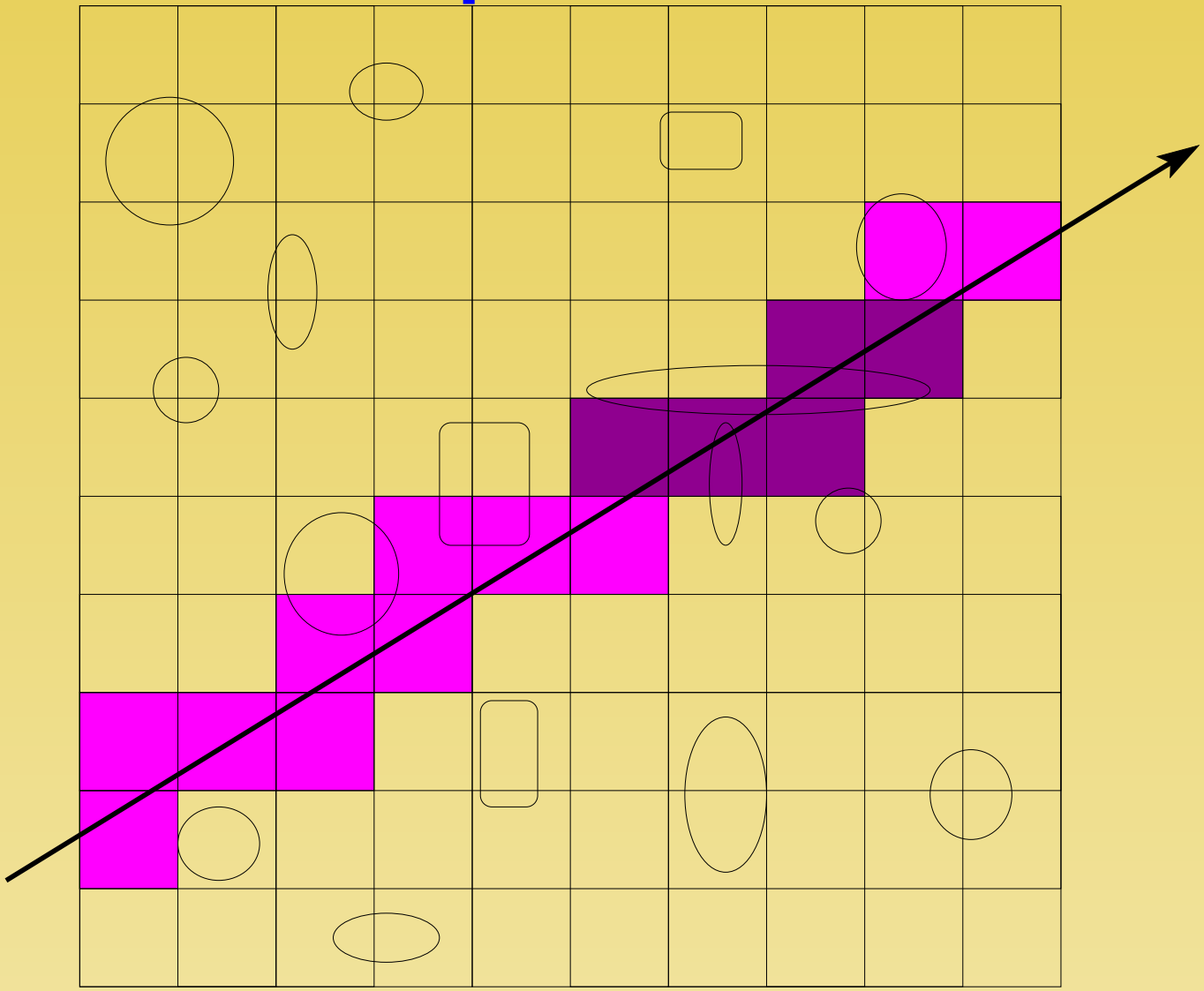


How do we determine the next voxel to test?3D-DDA 3D Digital Differential Analyzer

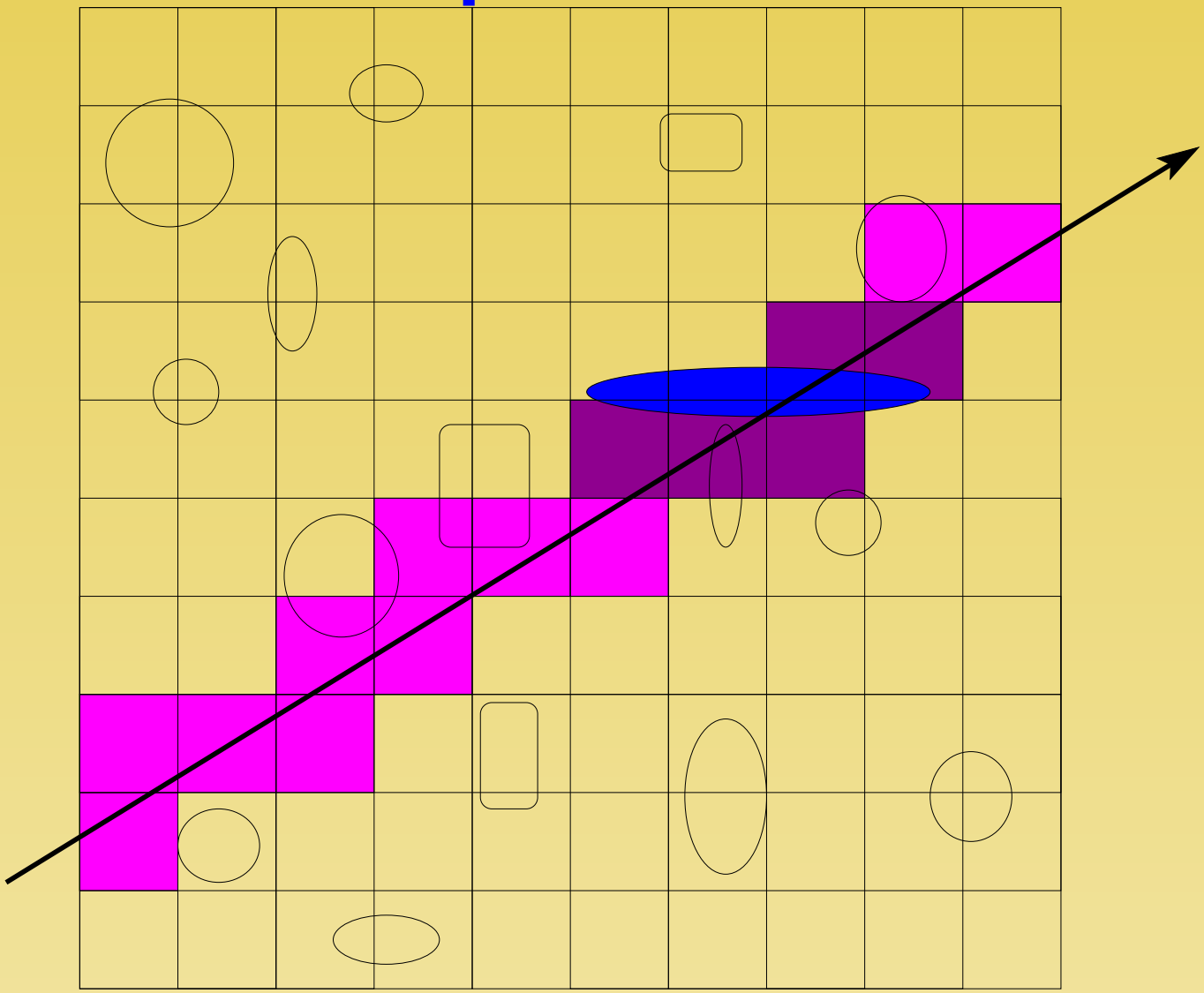
Uniform Space Subdivision



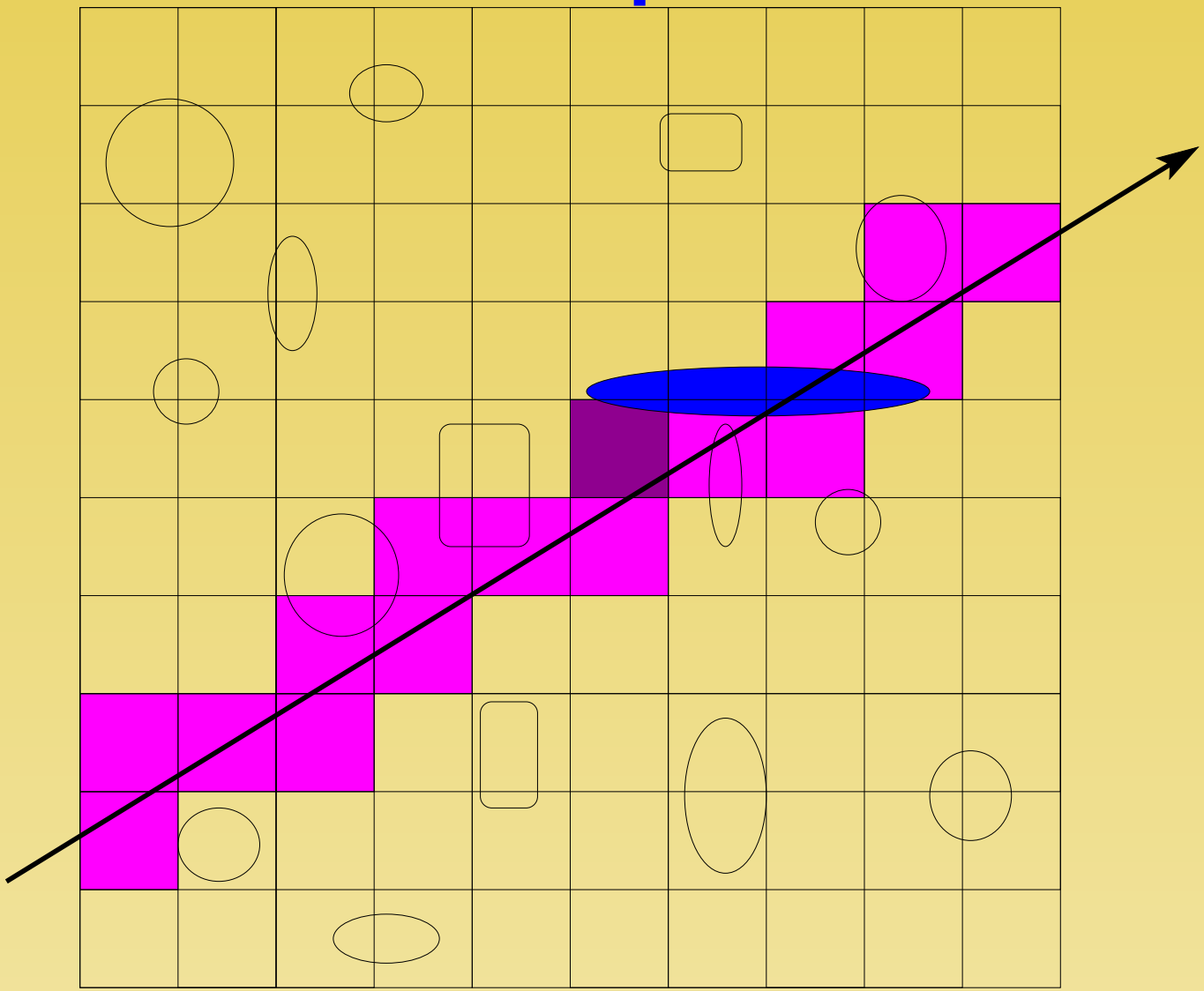
Uniform Space Subdivision



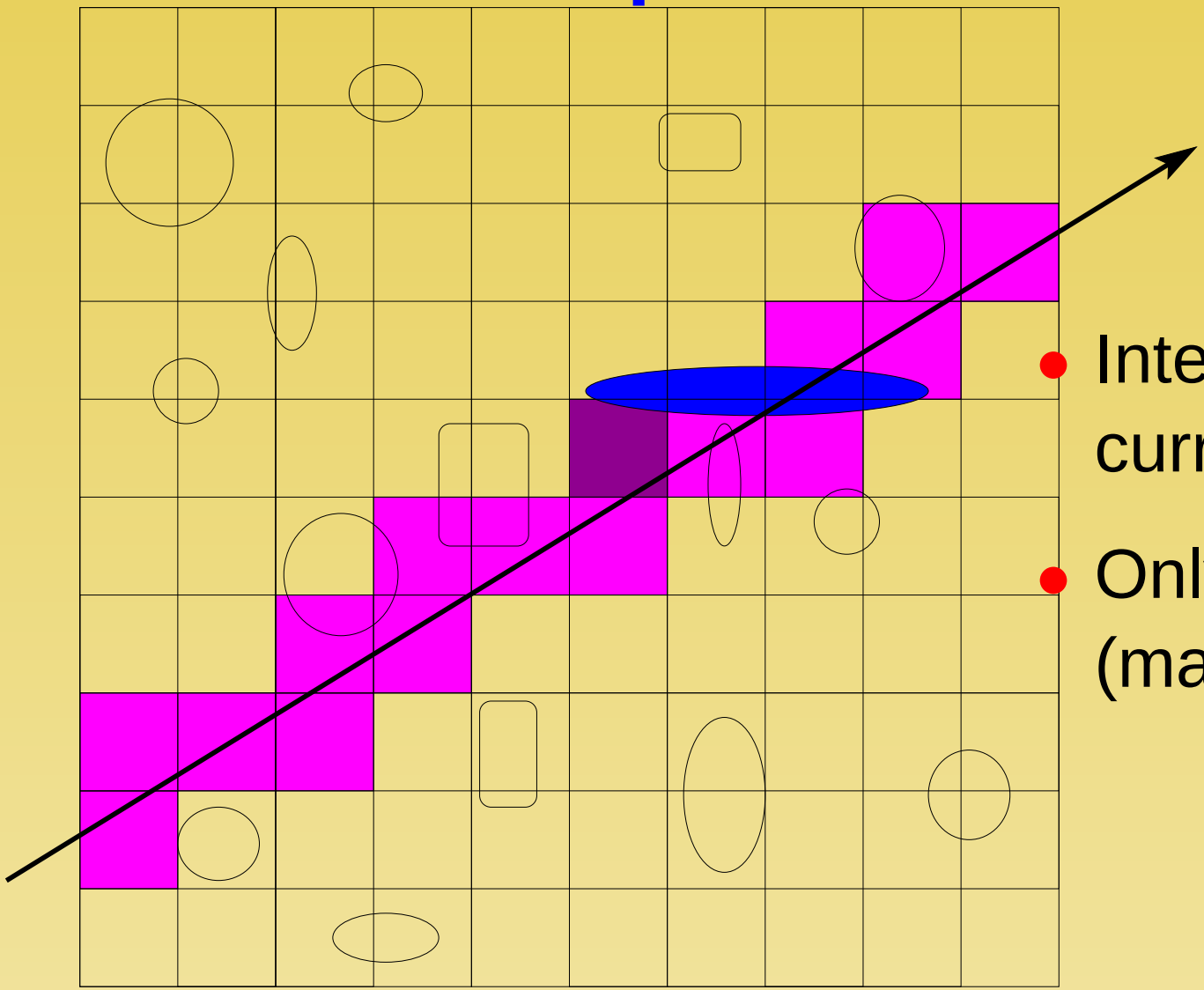
Uniform Space Subdivision



Uniform Space Subdivision



Uniform Space Subdivision

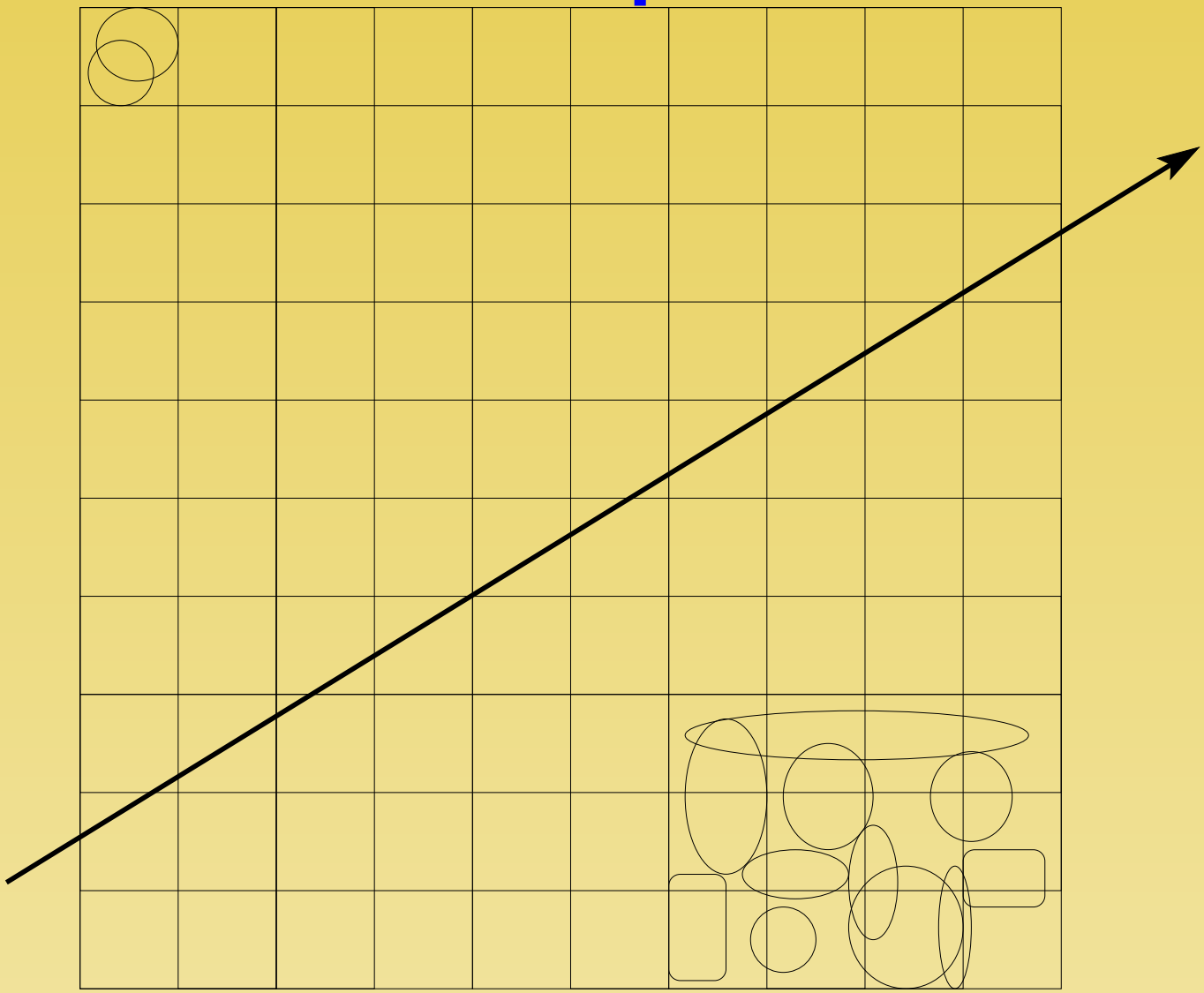


- Intersection in current voxel.
- Only test once (mailboxes).

Uniform Space Subdivision

- To construct, start with scene bounding box then sub-divide.
- What affect does the size of the voxels have.

Uniform Space Subdivision

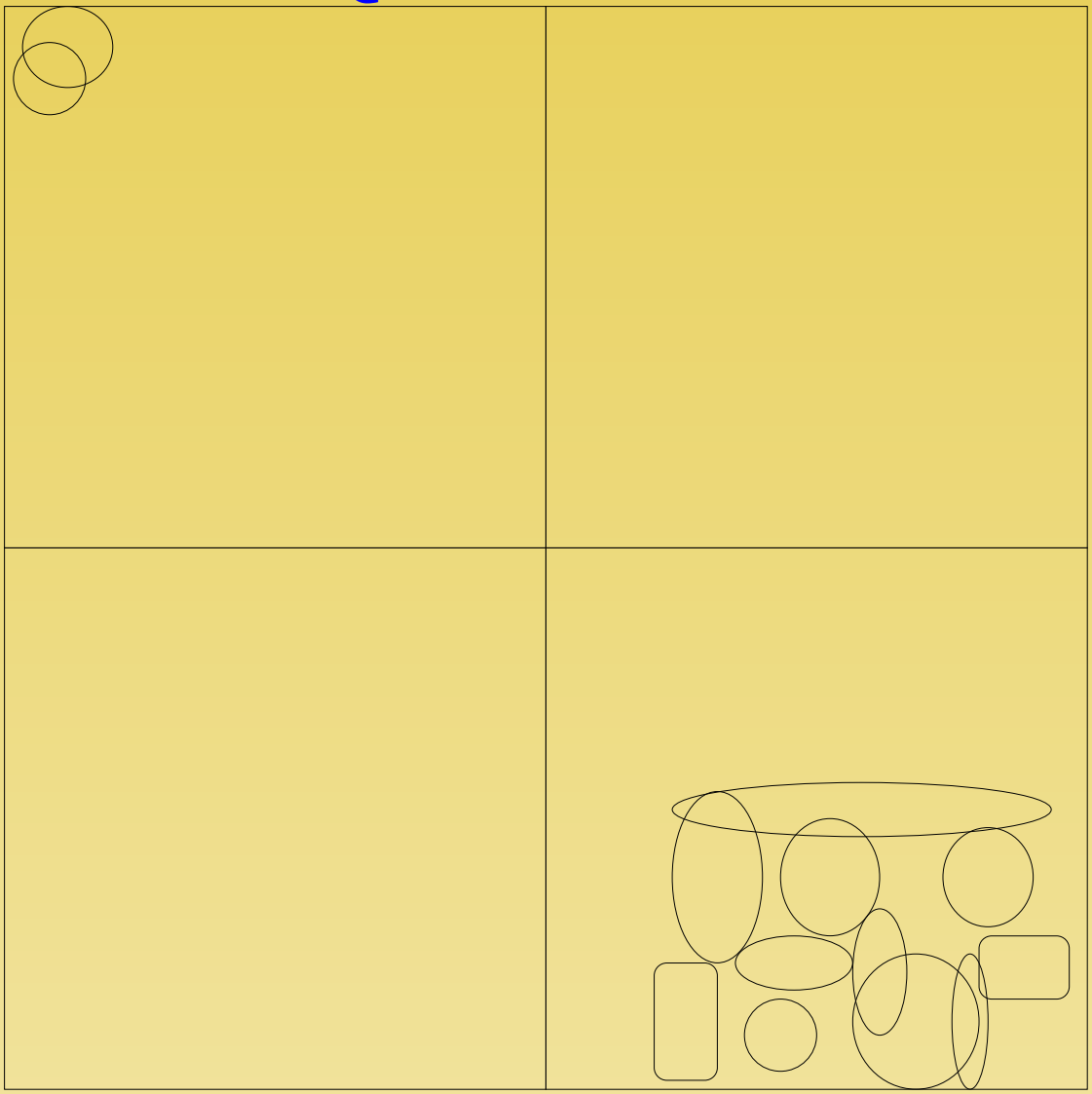


Why is this bad?

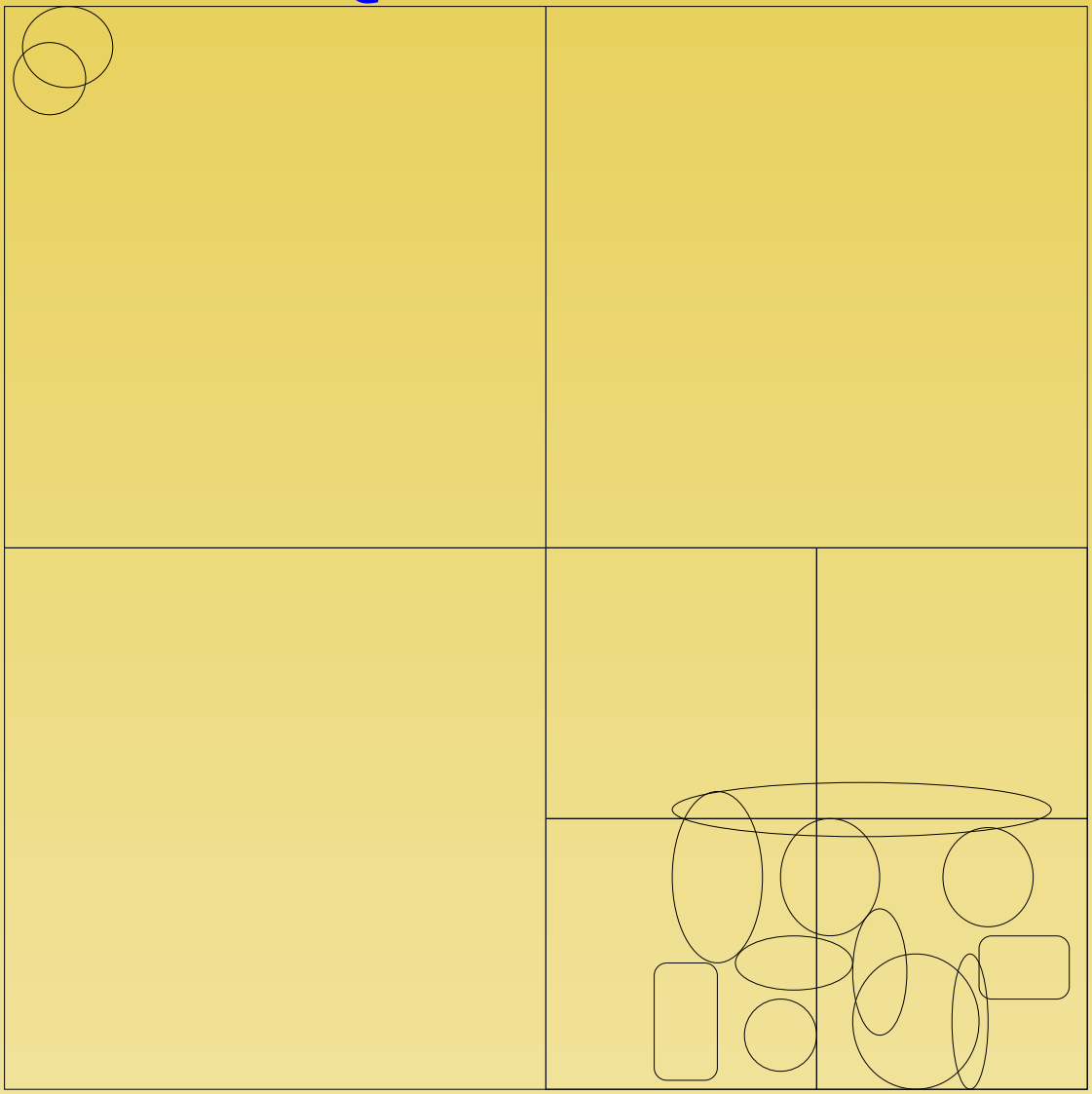
Nonuniform Subdivision (Hierarchical)

- Instead of having lots of empty little cells, lets have just a few empty big cells.
- This gives us a tree structure (hierarchy again!)
- Less voxels.
- But at what cost?
- Octrees.
- KD trees.
- BSP trees.

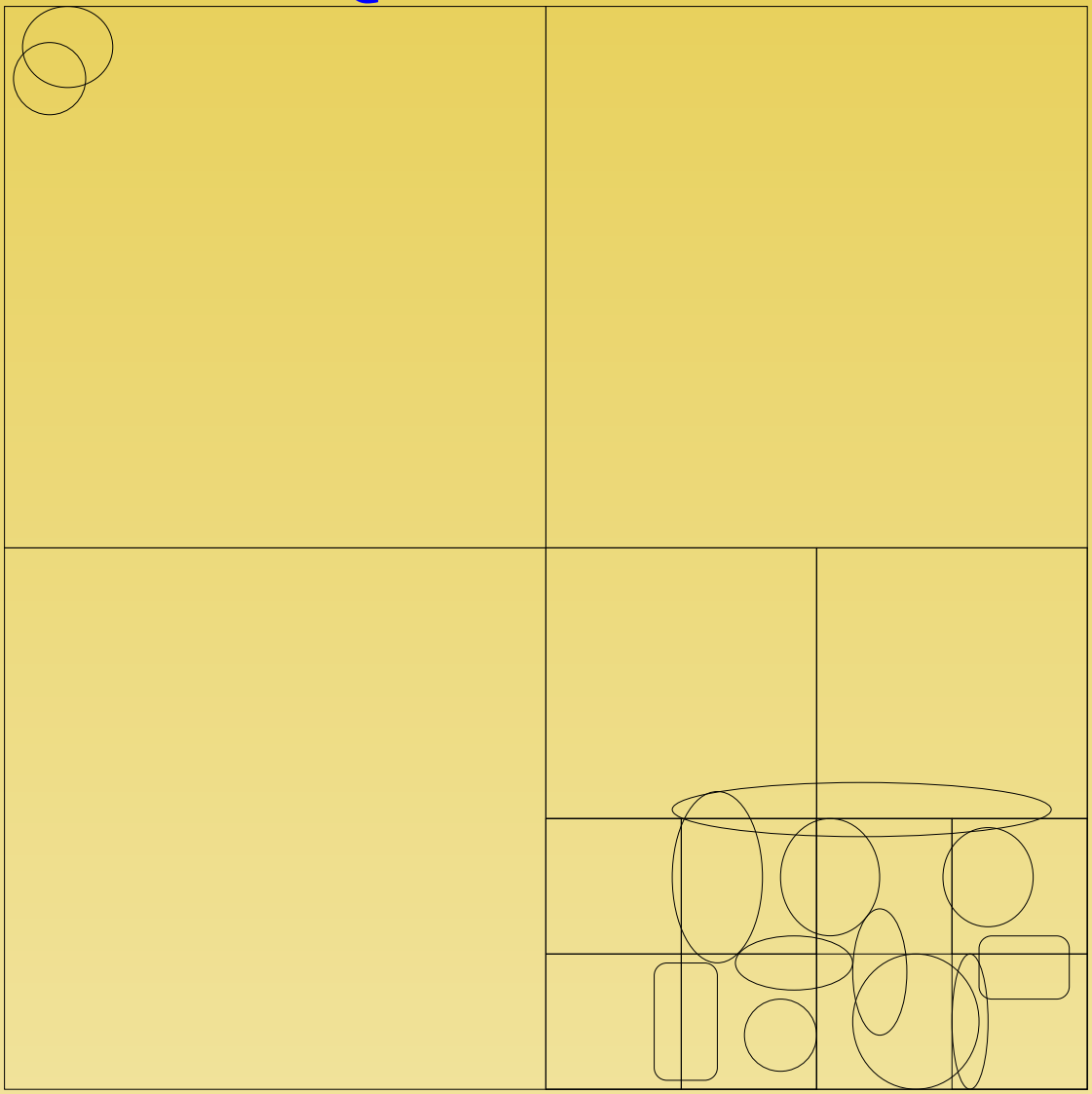
Quadtrees



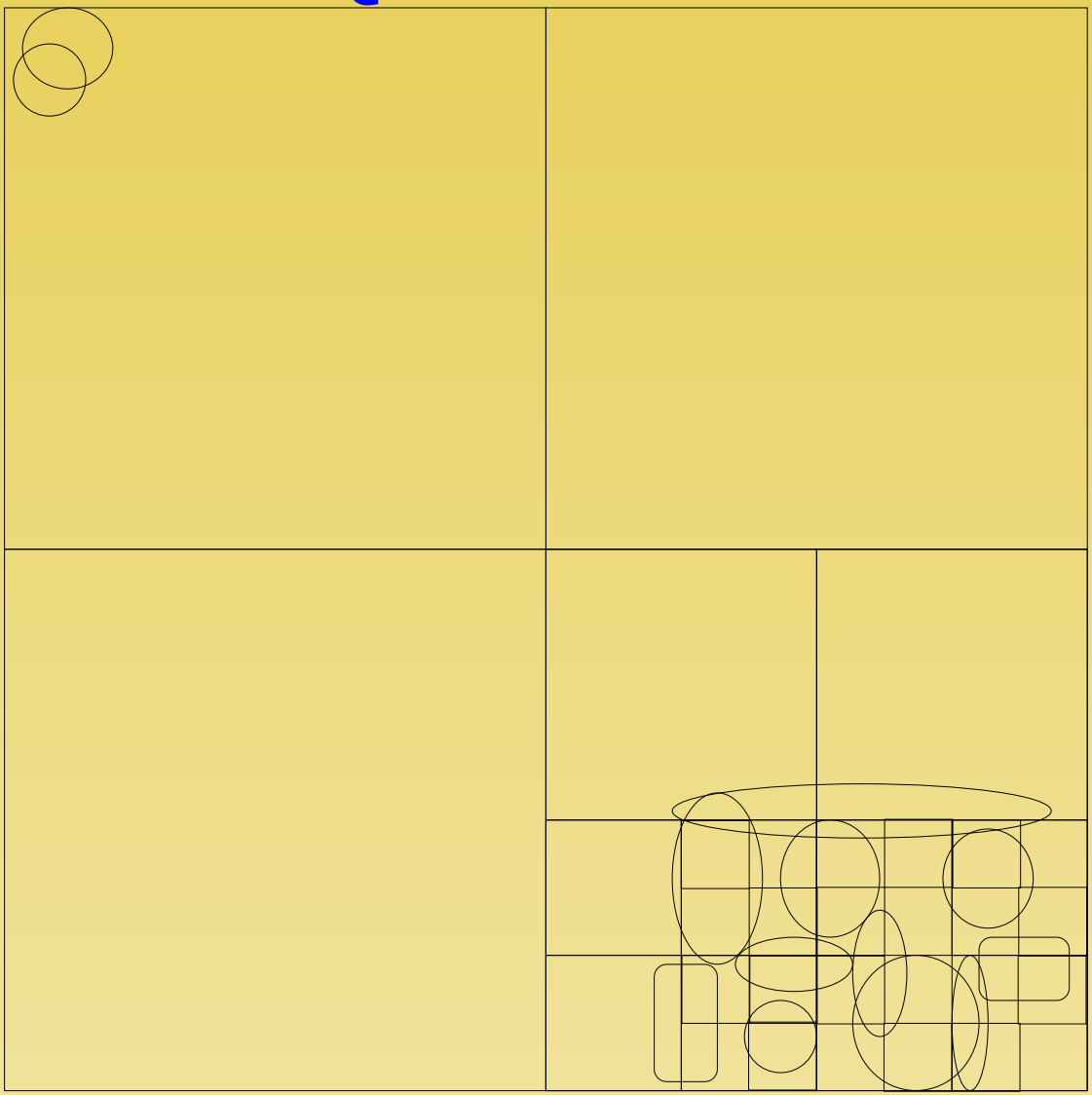
Quadtrees



Quadtrees



Quadtrees



Octrees

- Divide until a cell (voxel) has less than some number of objects (often 1) or until a minimum size is reached.
- Can subdivide on the fly to help improve efficiency of quadtree

Octrees: Dynamic Subdivision

- Don't divide it voxel too small or fewer than x objects.
- Don't divide if less than N rays (4 is good) and none of them have hit an object.
- Don't divide if $MK \geq N$, (M - num rays through voxel that hit something, k (2 or higher) user defined weight. (If a voxel is working why subdivide?)

Octrees: Ray Traversal Algorithms

- Point location from root (Glassner)
- Neighbor finding (Samet)
- Recursive inorder traversal (Kaplan, Arvo, Jansen)

Hiearchical Grids

- Instead of just one type of grid, why not do a combination?
- Recursive grids - Jevans and Wyvill (1989).
- Hierarchy of uniform grids - Cazals *et al. (1995)*
- *Adaptive grid - Klimaszewski (Ph.D. 1997)*

Recursive grids - Jevans and Wyvill (1989)

- Construct a uniform grid.
- Place objects within the grid.
- Recursively descend into voxels with too many objects creating a new uniform grid.
- Continue until maximum depth or minimum number of objects per voxel.

Hierarchy of uniform grids - Cazals *et al.* *(1995)*

- Make a global grid.
- Compute histogram based on object size.
- Split object into groups using histogram.
- Cluster objects within groups according to distance.
- For clusters with too many objects insert a new sub grid .

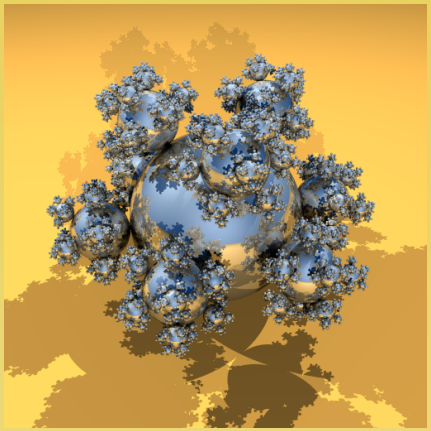
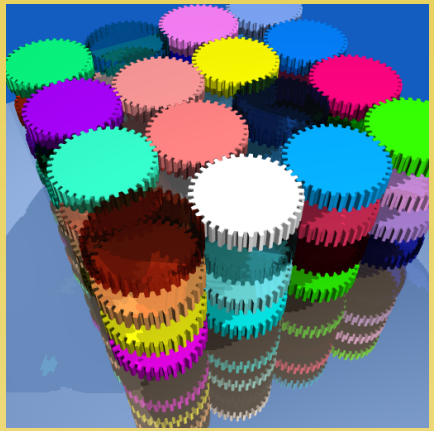
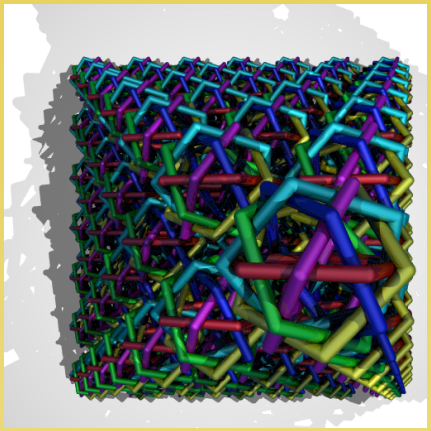
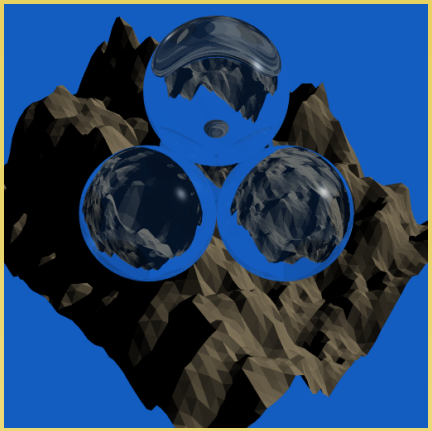
Adaptive grid - Klimaszewski (Ph.D. 1997)

- Cluster two things at a time (objects, clusters)
- Cluster based on distance, surface area of new cluster, and overall size.
- Build a bounding volume hierarchy over clusters.
- Build a grid for leaves of hierarchy.
- Subdivide these grids if too many objects.

Hierarchical Grid Differences

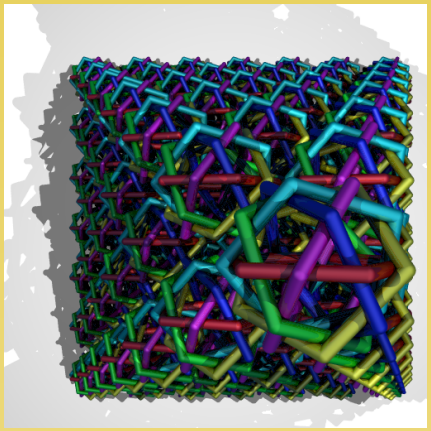
- Recursive grids cannot overlap, others can, so each nested grid tested only once.
- HUG requires mail boxes for traversal using grids as objects.
- Ray traversal very similar but the BVH must be traversed for adaptive grids.
- data from http://www.acm.org/tog/resources/ RT-News/html/rtnv12n1.html#art3

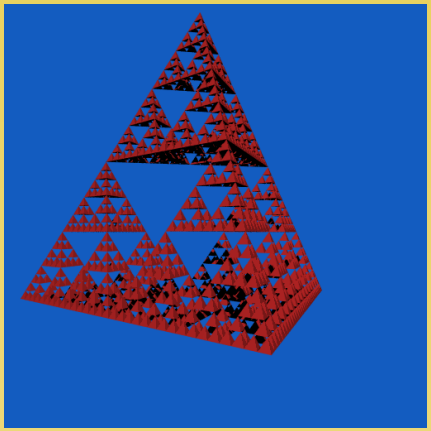
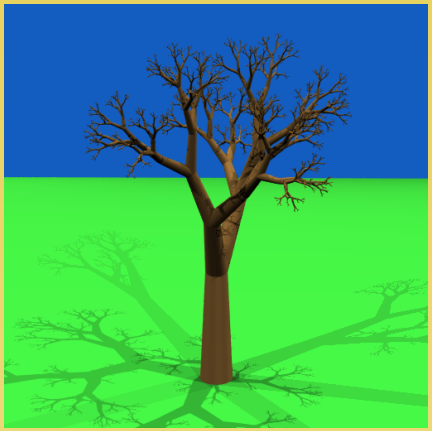
Hierarchical Grid Comparison (startup)

	balls	gears	lattice	mount
Uniform - D = 1.0	0.19	0.38	0.38	0.26
Uniform - D = 20.0	0.39	1.13	1.27	0.4
Recursive	0.39	5.06	0.16	1.98
HUG	0.4	1.04	0.3	0.16
Adaptive	2.79	5.88	0.6	2.54

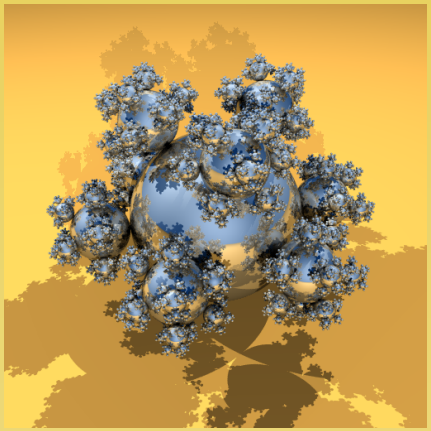
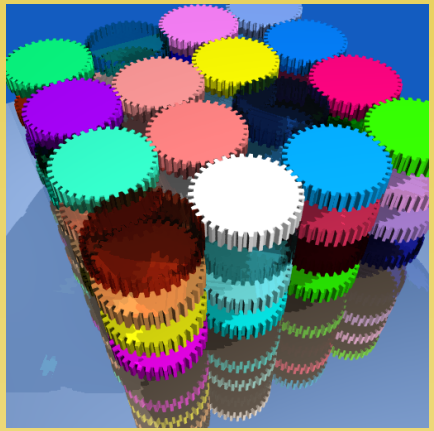
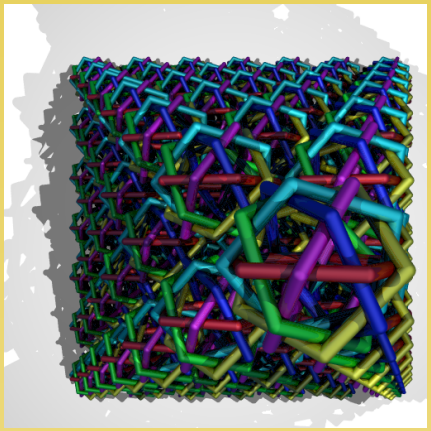
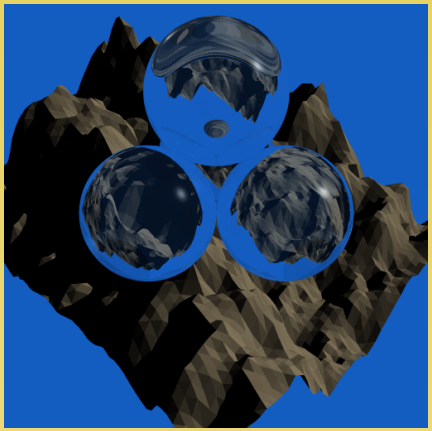
Hierarchical Grid Comparison (startup)



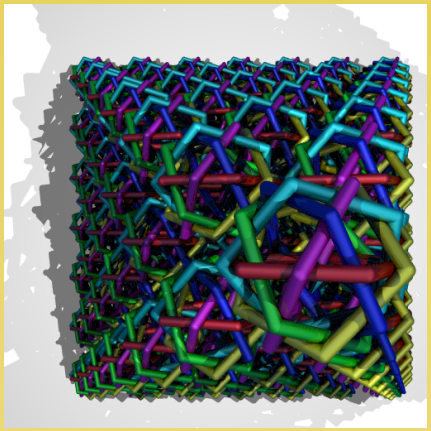
	rings	teapot	tetra	tree
Uniform - D = 1.0	0.35	0.3	0.13	0.22
Uniform - D = 20.0	0.98	0.65	0.34	0.33
Recursive	0.39	1.55	0.47	0.28
HUG	0.45	0.53	0.24	0.48
Adaptive	1.82	3.16	1.19	1.46

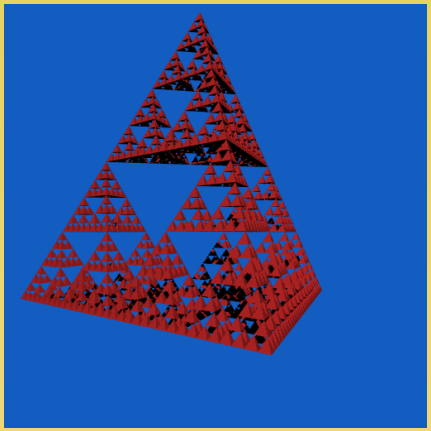
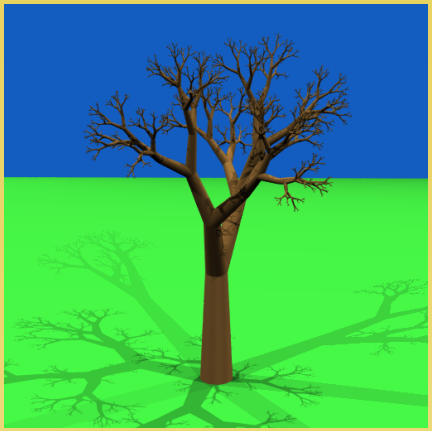
Hierarchical Grid Comparison

	balls	gears	lattice	mount
Uniform - D = 1.0	244.7	201.0	54.68	28.99
Uniform - D = 20.0	38.52	**192.3**	**34.21**	**25.15**
Recursive	36.73	214.9	82.05	30.28
HUG	34.0	242.1	71.62	62.31
Adaptive	**30.51**	330.0	129.6	59.05

Hierarchical Grid Comparison



	rings	teapot	tetra	tree
Uniform - D = 1.0	129.8	28.68	5.54	1517.0
Uniform - D = 20.0	**83.7**	**18.6**	**3.86**	781.3
Recursive	113.9	22.67	7.23	33.91
HUG	116.3	25.61	7.22	33.48
Adaptive	167.7	43.04	8.71	**18.38**

KD Trees

Java Demo Snagged from

http://www.rolemaker.dk/nonRoleMaker/uni/algogem/kdtree.htm

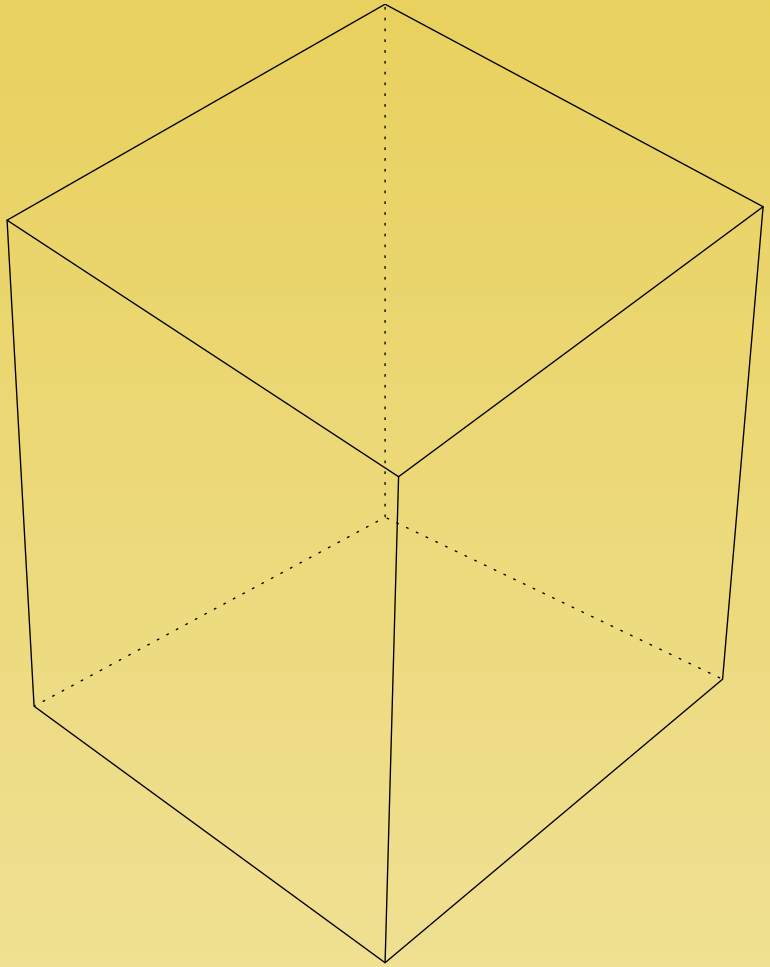
BSP Trees

Java Demo Snagged from
http://symbolcraft.com/graphics/bsp/index.html
http://graphics.lcs.mit.edu/ jcyang/6.838/pset2/

Nonuniform vs Uniform Subdivision

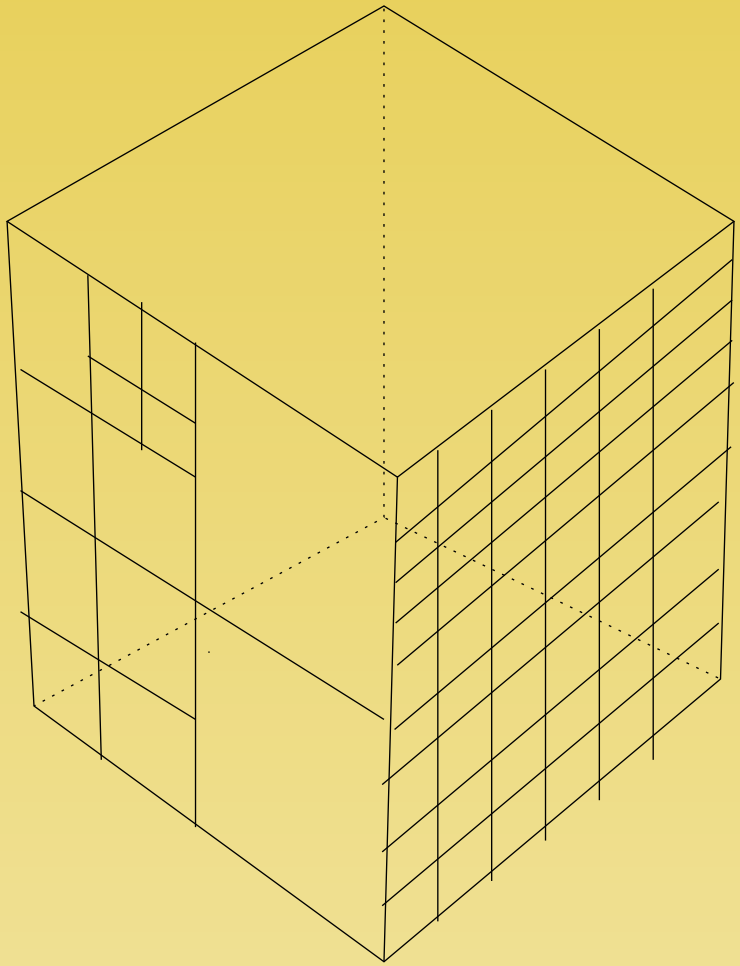
- How does finding next voxel compare?
- Which has more empty voxels?
- Are empty voxels bad?
- Teapot in a stadium.

Directional Technigues



A Ray eminating from the center will pierce one face!

Directional Technigues



- Light Buffer - Haines and Greenburg (1986).
- Ray Coherence - Ohta and Maekawa (1987).
- Ray Classification - Arvo and Kirk (1987).

Light Buffer

- Do we care which object occludes a light ray?

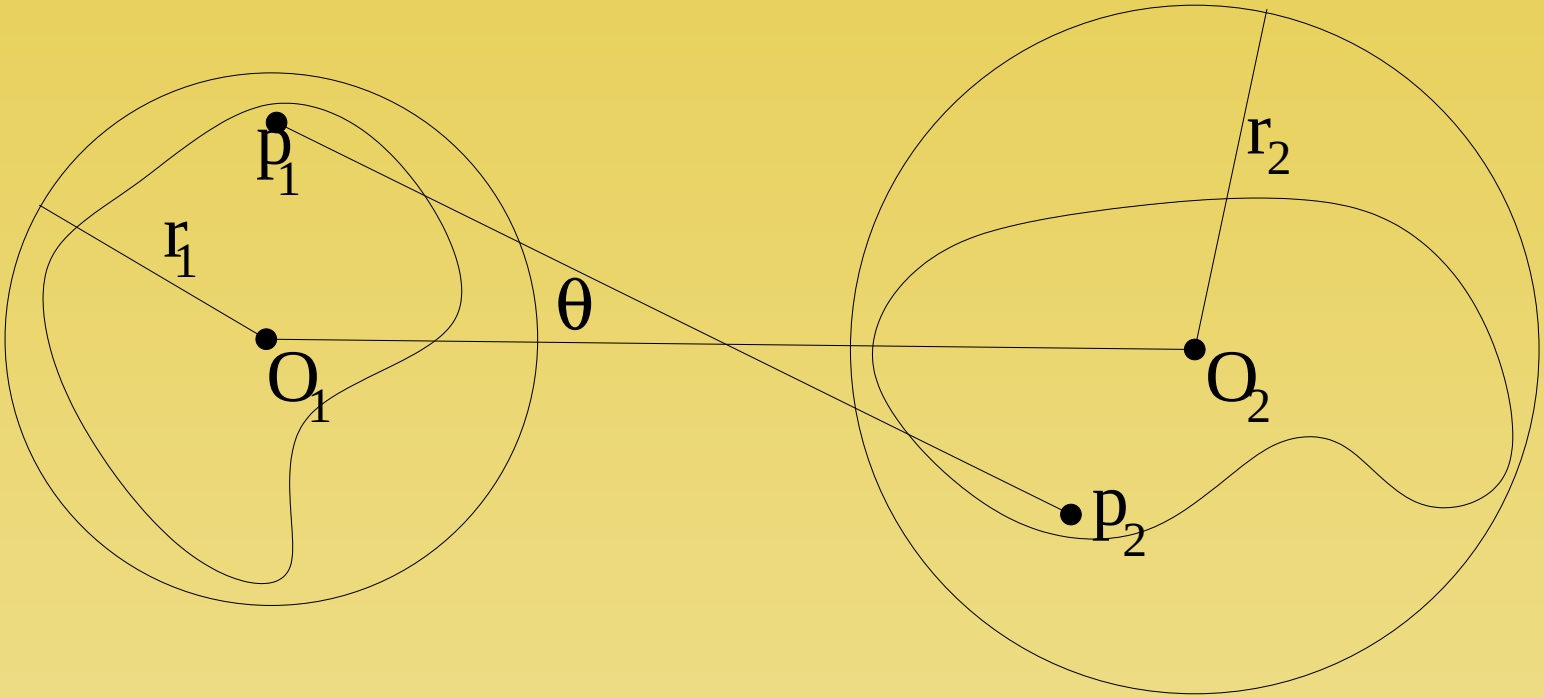
Light Buffer

- Do we care which object occludes a light ray?
- How did you do the test in your ray tracer?

Light Buffer

- Do we care which object occludes a light ray?
- How did you do the test in your ray tracer?
- How could we use the direction information?

Ray Coherence



$$cos\theta > \sqrt{\left(1 - \frac{r_1 + r_2}{||O_1 - O_2||}\right)}$$

Ray Coherence

- Make a direction cube for all ray origins (lights, eye, reflective surface, refractive surface).

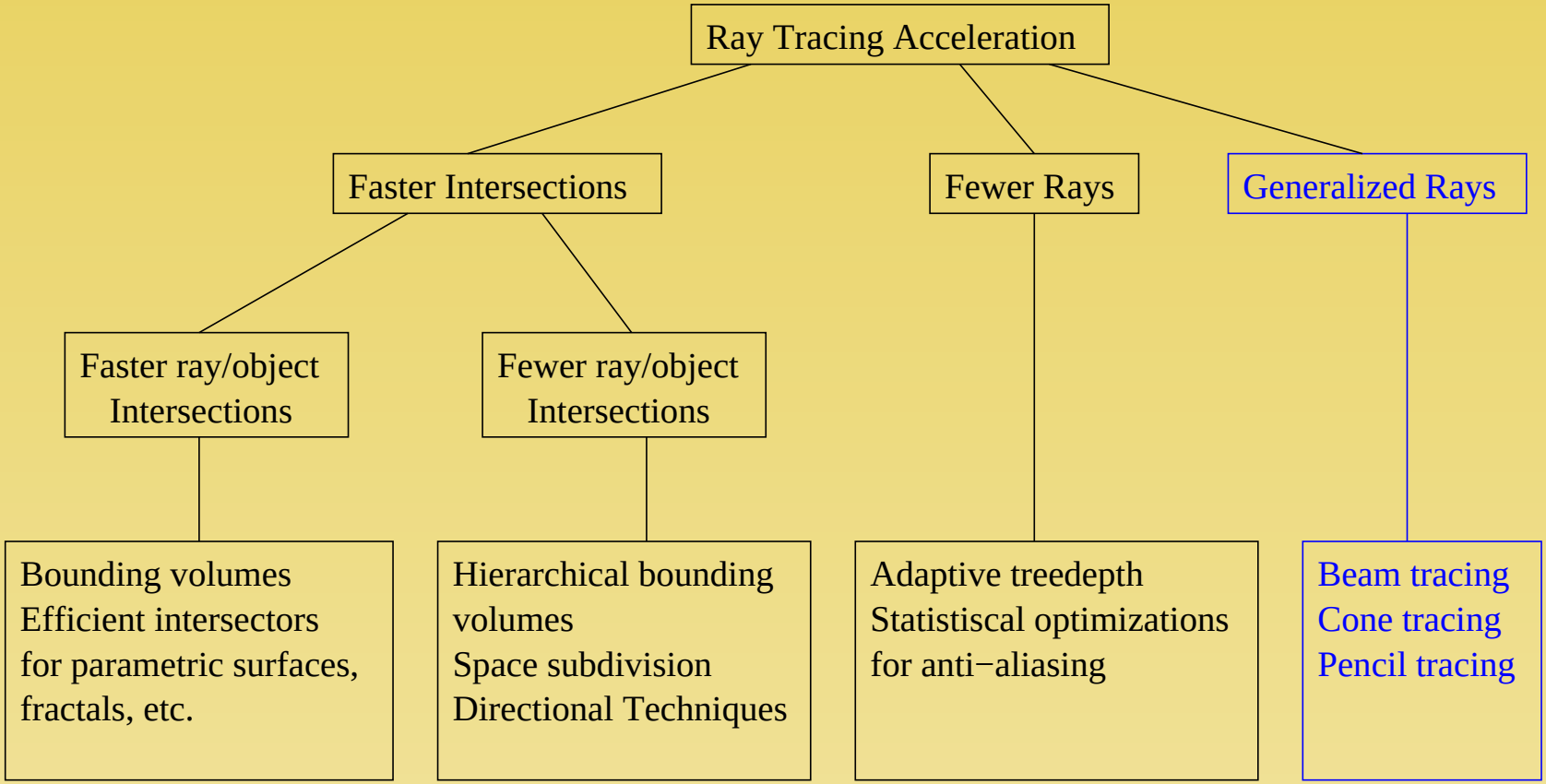
Ray Classification

- Subdivide *Ray space instead of object space.*
- *Gives 5D hypercubes*
- *Given a ray, only objects in its hypercube can possibly be intersected.*
- *No traversal of object space required.*

Coherence

- Object
- Image (viewing dependent object)
- ray coherence similar rays intersect similar objects.
- frame coherence (image + temporal)
- spatial, temporal.

Ray Tracing Acceleration Classification



Generalized Rays

- Better anti-aliasing.
- Exploiting coherency.
- Gains: speed, anti-aliasing, effects
- Lose: flexibility (primitive types), approximation to intersection.

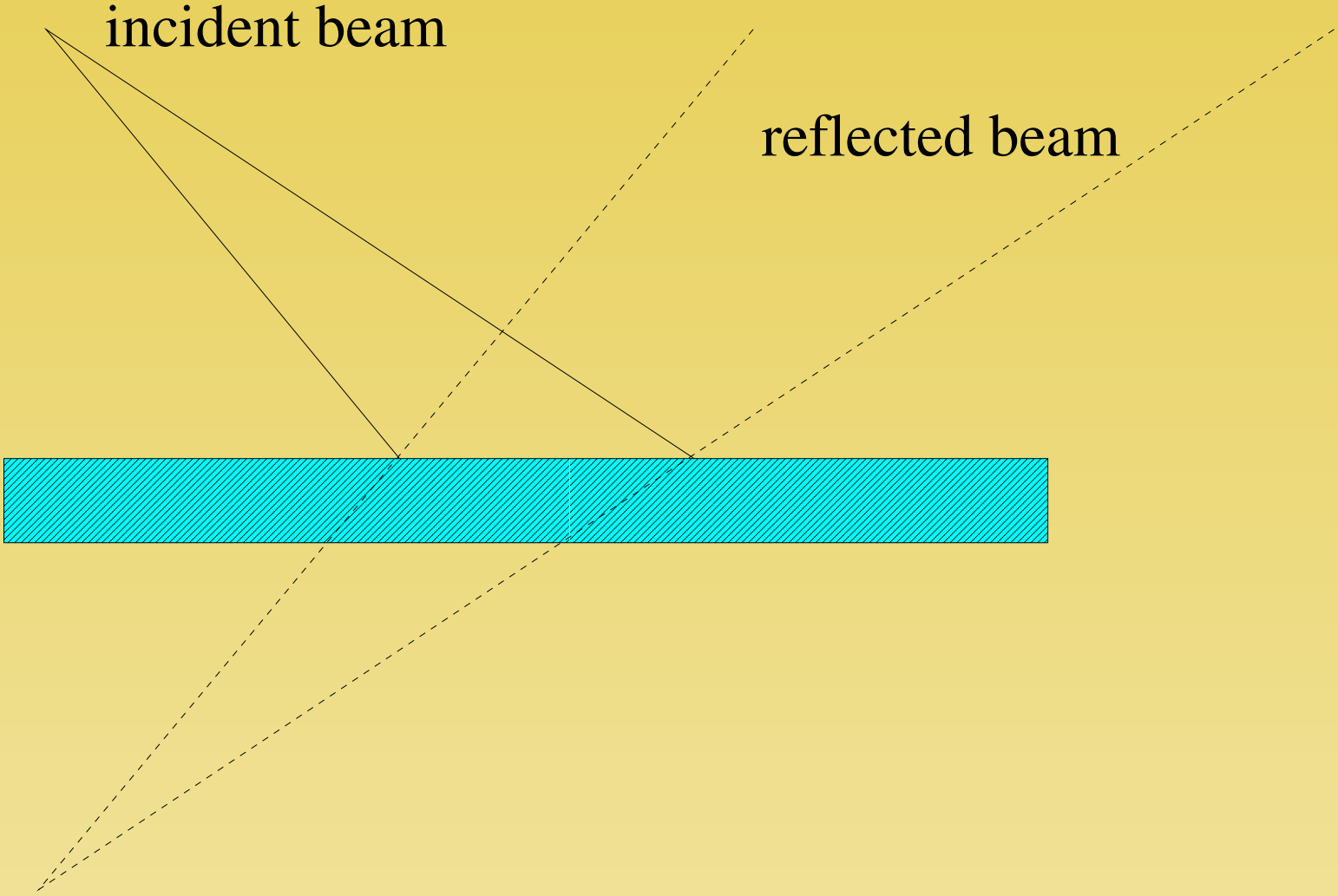
Cone Tracing - Amanatides (1984)

- Ray are generalized as right circular cones, apex, center line, and spread angle.
- Detect intersections and % of cone blocked.
- Maintain list of closest intersections to combine partial intersections.
- New center line for reflections/refractions uses normal, but apex and angle uses curvature.
- Only allows spheres, planes, and polygons.
- Will do penumbrae.

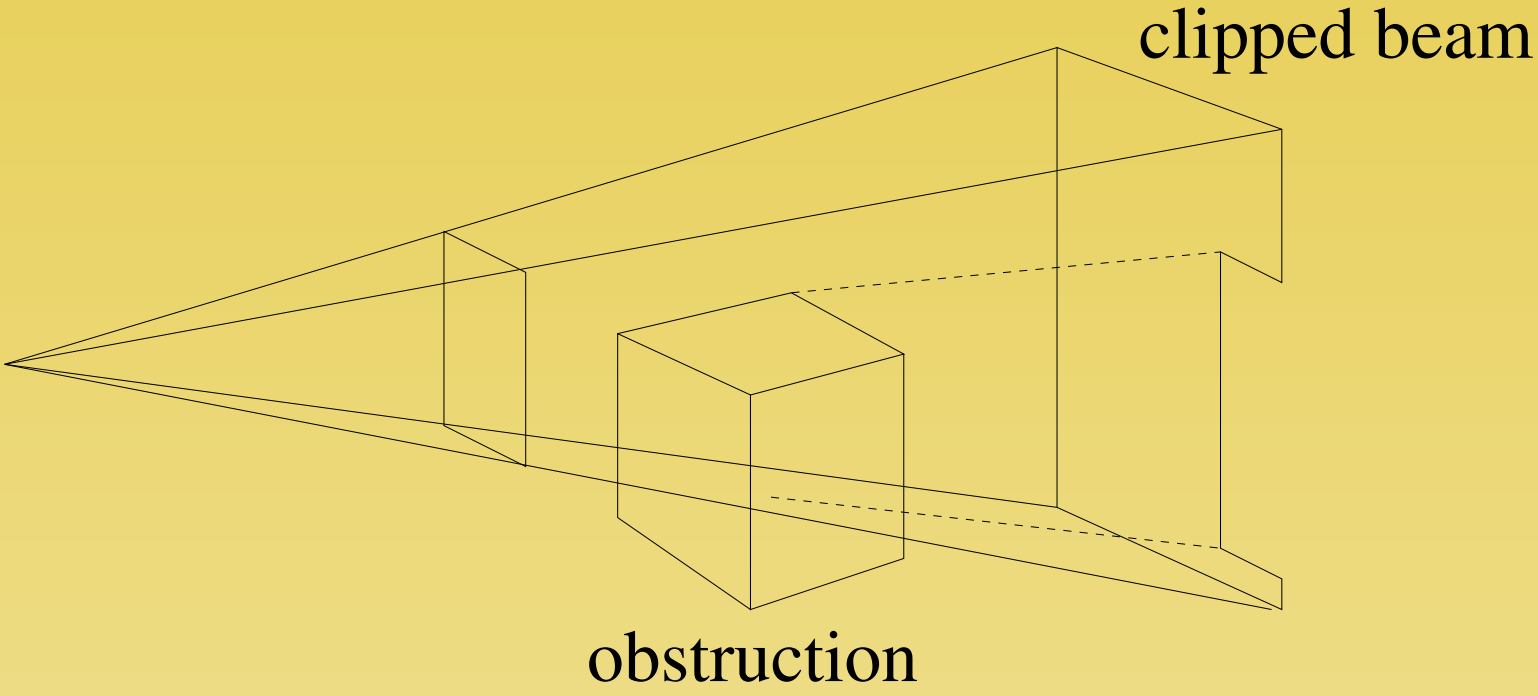
Beam Tracing

- Heckbert and Hanrahan (1984)
- Why trace an infinitesimally thin ray?
- Why not take advantage of ray coherence (almost parallel rays close together will most likely hit the same objects)?
- Based on the Weiler-Atherton hidden surface removal algorithm.

Beam Tracing



Beam Tracing



Beam Tracing

- Only works for polygonal objects.
- Beams that only partially intersect objects become complex.
- Refraction is nonlinear so beam geometry not preserved, must be approximated.
- It is more efficient.
- Produces better quality.
- Can combine with volume hierarchies and spatial subdivision.

Pencil Tracing - Shinya *et al. (1987)*

- Reference rays (axial ray)
- Nearby rays (paraxial rays) form the pencil - 4D vectors in a coordinate system of the axial ray.
- Uses paraxial approximation theory (from optical design) to trace ray.
- Pencil transformations are linear (4x4 matrices)
- Erorr analysis gives max spread angle of pencil.
- Requires smooth surfaces.

Rasterization-based Approximations

- Use rasterization to compute frames.
- Add special effects with ray tracing.
- Approximation so shows artifacts.
- Assumes ray tracing slower.

Image-based Approximation

- Approximate new frames from old frame.
- Add new information with ray tracing as time allows.
- Great for static scenes.
- Artifacts for dynamic scenes.

Ray Tracing-based Approximation

- Try to reduce the cost per pixel with approximations.
- Adaptive sampling (pixel-selected ray tracing: color interpolation in image space for regions with smooth changes in color. Only image regions with high contrast are sample at high rates.
- Interpolate in ray space with error bounds.

Accelerating ray tracing.

- BSP-trees, octrees, hierarchical grids, bounding volume hierarchies, hybrids.
- New methods deal with running on newer hardware, which doesn't deal well with recursion.
- Parallel techniques (load balancing, latency hiding, reduced comm.)
- Hardware implementation.

Frustum Culling

- remove bounding volumes that are outside the viewing frustum.
- Good for space partitioning?

First hit speedup

Use some form of visibility algorithm such as Z-buffer. Assumes it is faster than ray casting.

Selective ray tracing of certain objects.

Parallel Rendering

- Raytracing is a ridiculously parallel algorithm.
- Through each pixel at a different processor.
- Scales nicely, limited by load balancing and synchro-nization
- Can employ frameless rendering.

Interactive Ray Tracing

- Logarithmic on number of primitives.
- Flexible
- More realism
- Shading done only after visibility.
- Parallel scalability.

Questions

- Which is faster? Ray-tracing or rasterization?

Questions

- Which is faster? Ray-tracing or rasterization?
- Which is better? Bounding-volumes or space subdivision?

Questions

- Which is faster? Ray-tracing or rasterization?
- Which is better? Bounding-volumes or space subdivision?
- Which is better? uniform or nonuniform subdivision?

Questions

- Which is faster? Ray-tracing or rasterization?
- Which is better? Bounding-volumes or space subdivision?
- Which is better? uniform or nonuniform subdivision?
- Which is better? Super-sampling or adaptive supersampling?

Questions

- Which is faster? Ray-tracing or rasterization?
- Which is better? Bounding-volumes or space subdivision?
- Which is better? uniform or nonuniform subdivision?
- Which is better? Super-sampling or adaptive supersampling?
- Which is better? Adaptive supersampling or jittering?

Questions

- Which is faster? Ray-tracing or rasterization?
- Which is better? Bounding-volumes or space subdivision?
- Which is better? uniform or nonuniform subdivision?
- Which is better? Super-sampling or adaptive supersampling?
- Which is better? Adaptive supersampling or jittering?