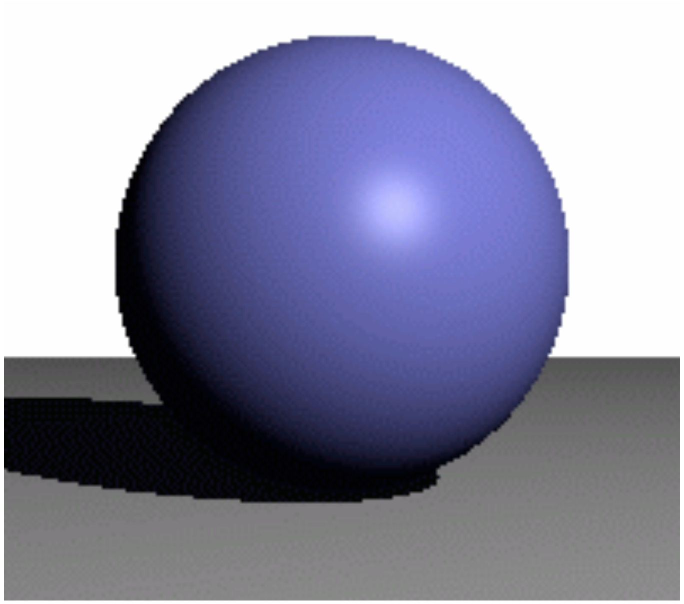
CSE 681

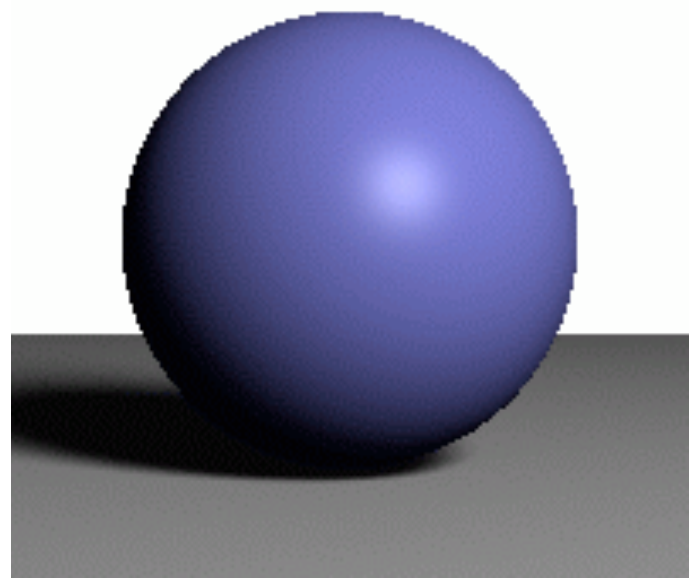
Distributed Ray Tracing

Shadows

- Assumption: The light source is a point
  - **Realistic: Soft shadows**



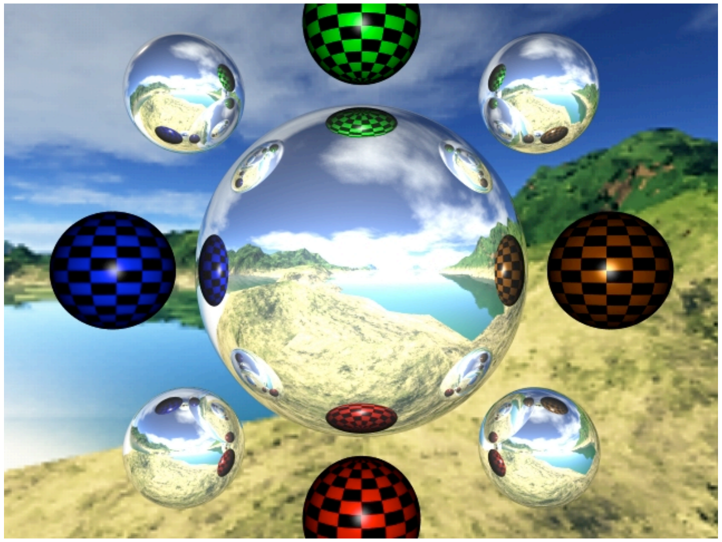
Point Light Source

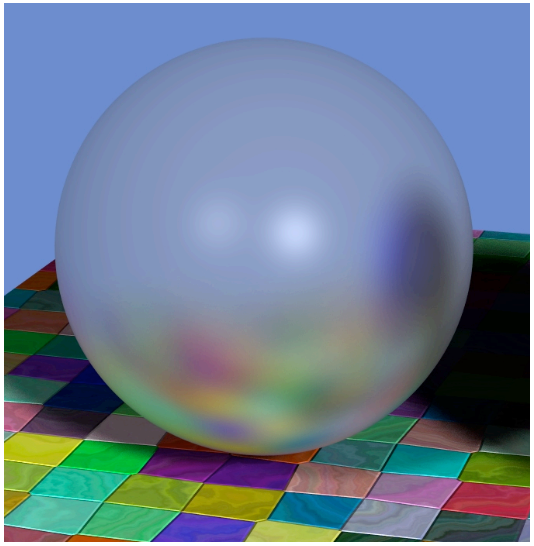


Area Light Source

Reflections

- Assumption: The surface is a perfect mirror, so the only reflection on a surface comes from the reflection vector
  - **Realistic: Glossy reflection**



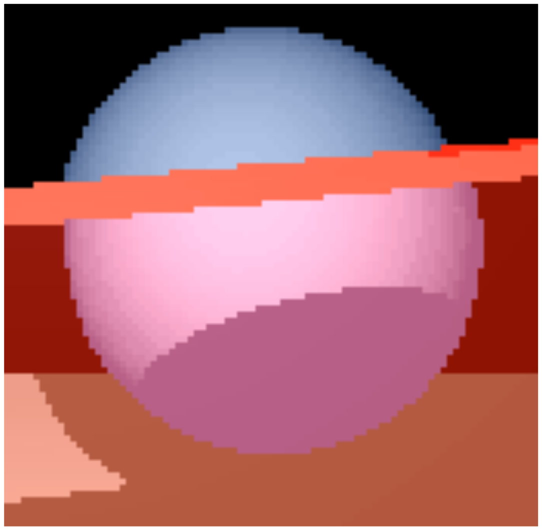


Justin Legakis

Andrew Zaferakis - http://www.cs.unc.edu/~andrewz/comp236/hw1/index.html

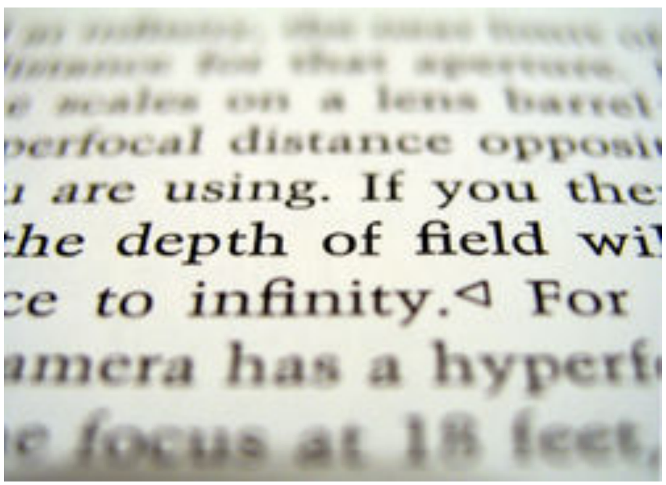
Refraction

- Assumption: Perfectly clear material, so the only refraction contribution comes from the transmittance vector
  - **Realistic: "Blurry" refraction**

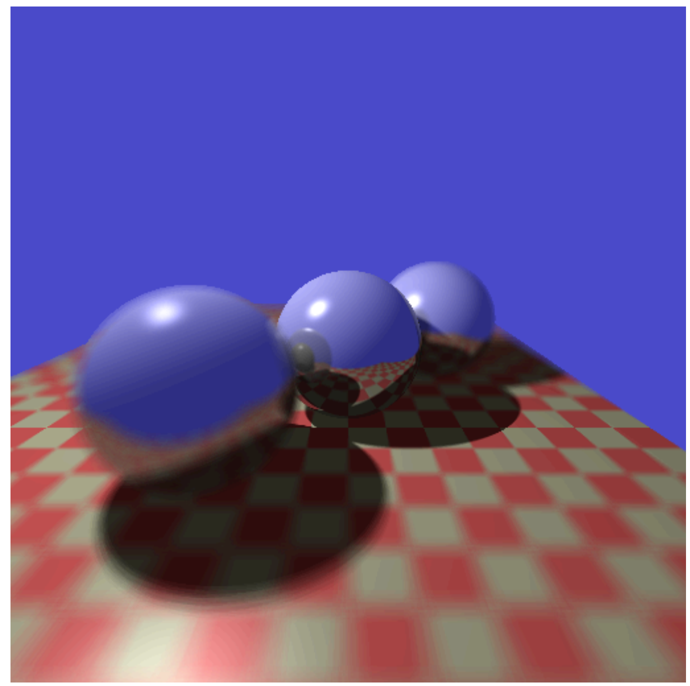


Depth of Field

- Assumption: Pinhole camera model
  - **Realistic: Focus depends upon focal length of a “real” camera lens**

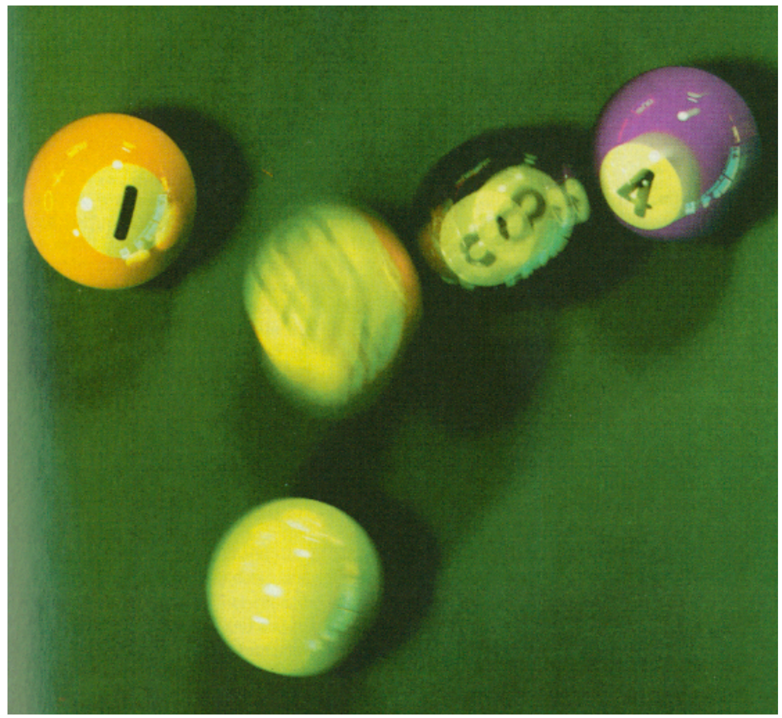


wikipedia.com



Motion Blur

- Assumption: Exposure time is instantaneous
  - **Realistic: Integrate (average?) frames over time**



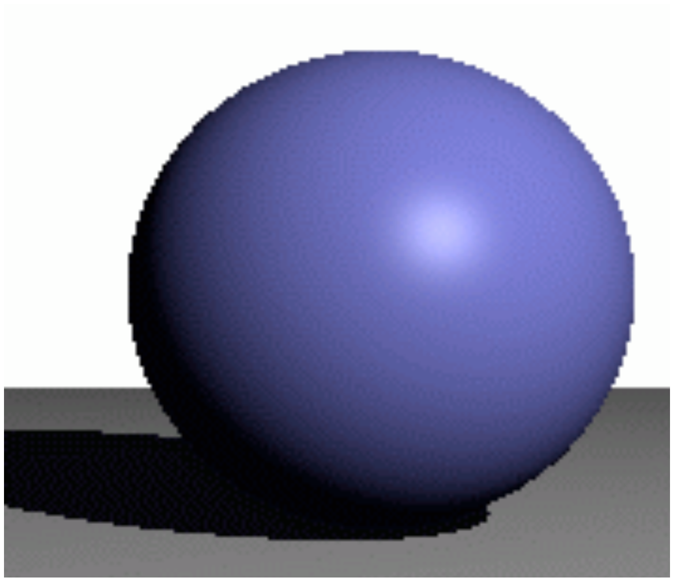
Distributed Ray Tracing (DRT)

- Improvements to this image:
  - Anti-aliased edges
  - Soft shadows
  - Glossy reflection
  - "Glossy" translucency
  - Objects in/out of focus according to a lens
  - Motion blur of fast moving objects (not shown here)

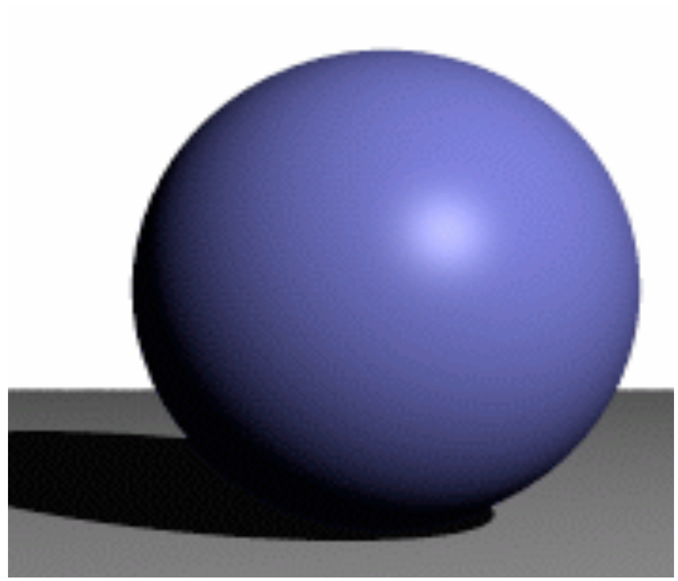
- **Main idea**: Replace our single ray approximations with a *distribution of rays*

DRT: Supersampling

- Anti-aliasing: remove jagged edges



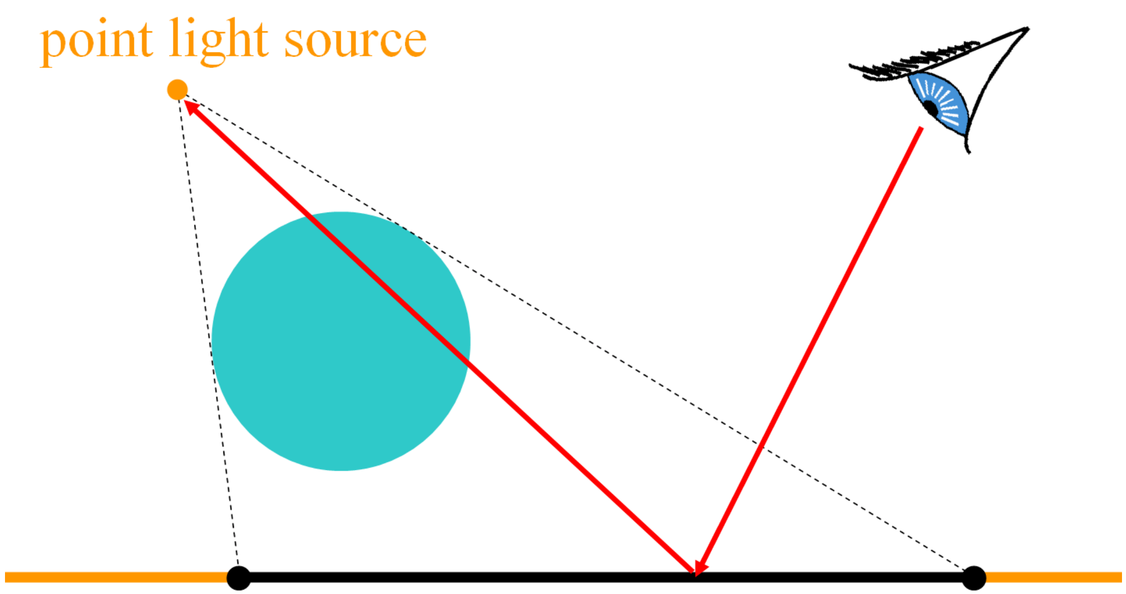
One sample/pixel



Multiple samples/pixel

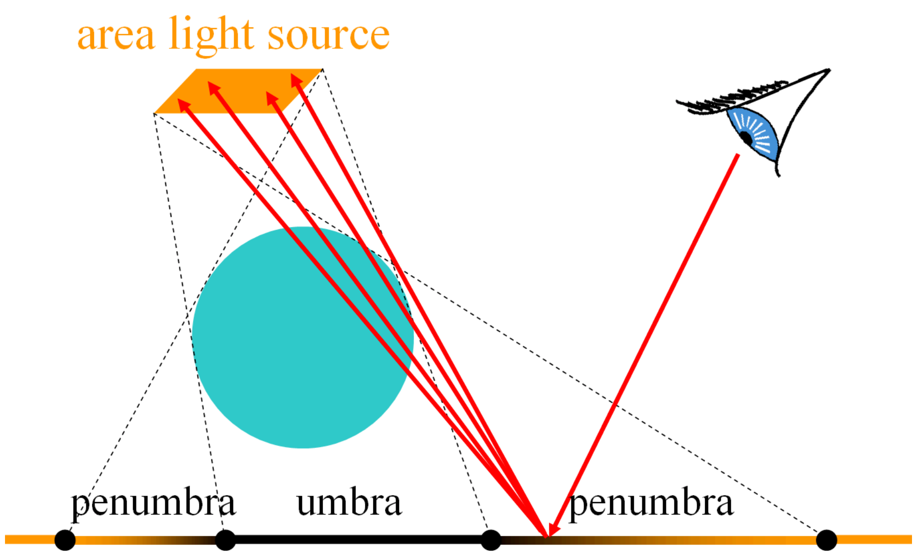
DRT: Soft Shadows

- *Problem*: Point light source
  - Only send a single shadow ray



DRT: Soft Shadows

- *Solution*: Use area light source and trace rays back to some point on the light's surface
  - Soft shadow = umbra + penumbra
    - *Umbra* results from total occlusion of a light source
    - *Penumbra* results from a partially occluded. light source
  - The distribution of the shadow rays is proportional to the energy intensity



Sampling the Area Light

- Stochastic sampling on the light source's surface provides anti-aliasing in the penumbra
- The light source may be treated as a sphere and random positions chosen on the sphere's surface to send a population of shadow rays
- Usually, the light source is modeled as a plane oriented towards the scene

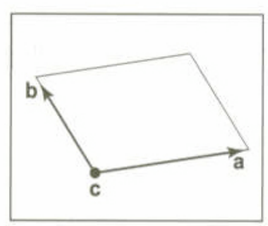
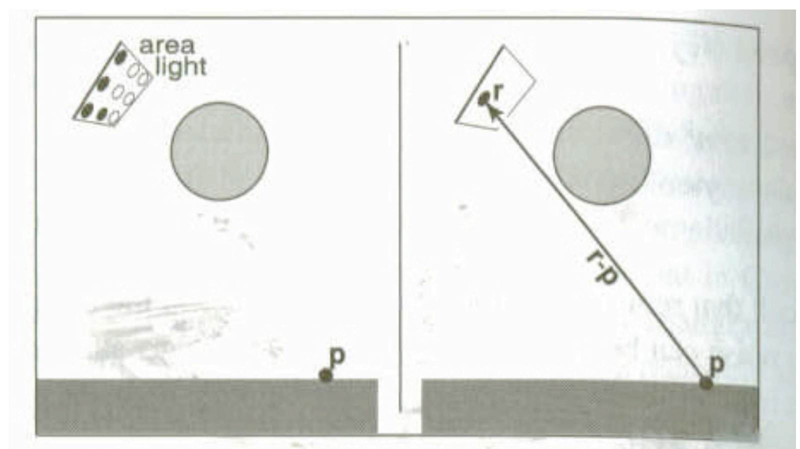
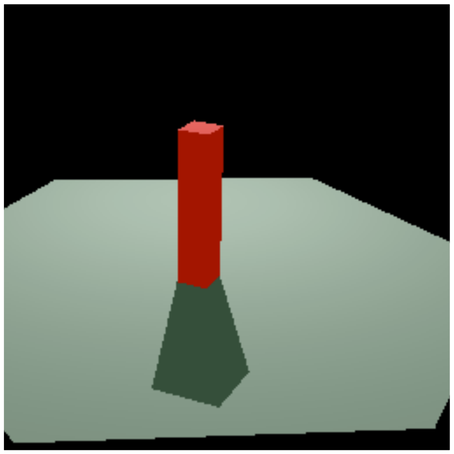


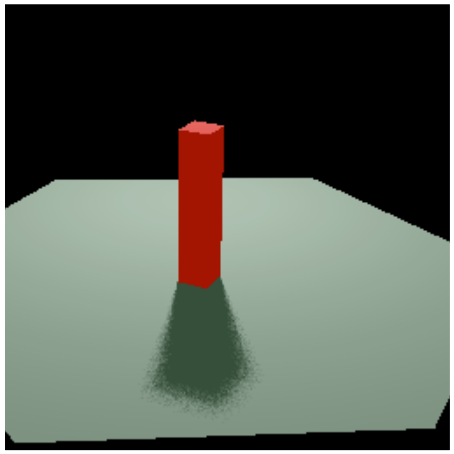
Figure 10.32. The geometry of a parallelogram light specified by a corner point and two edge vectors.



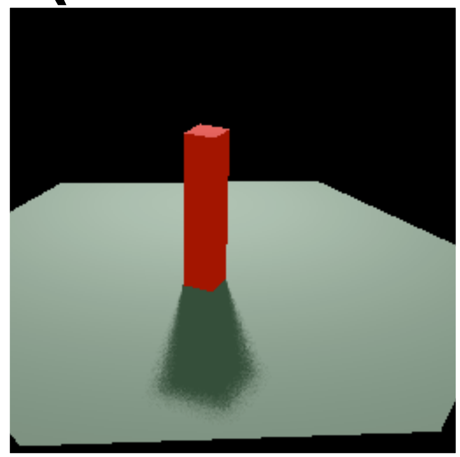
Soft Shadows (Penumbras)



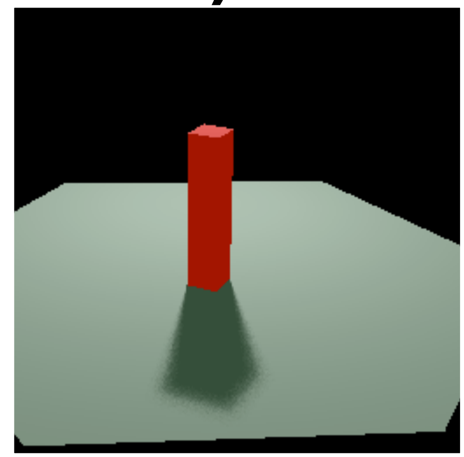
1 Ray



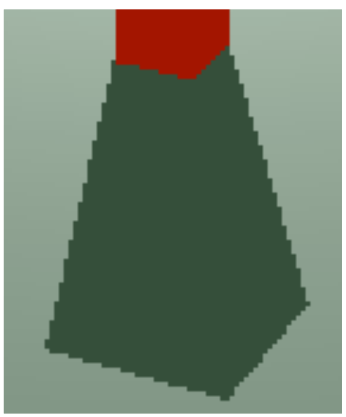
10 Rays

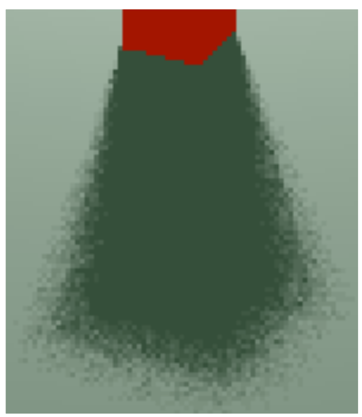


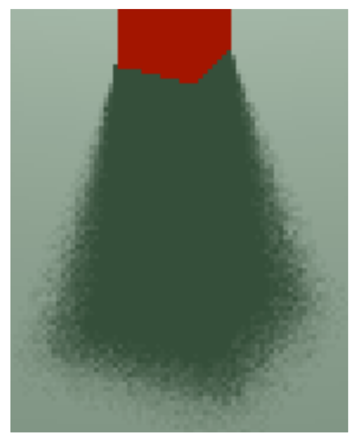
20 Rays

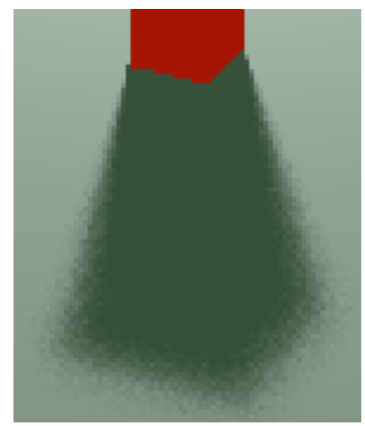


50 Rays





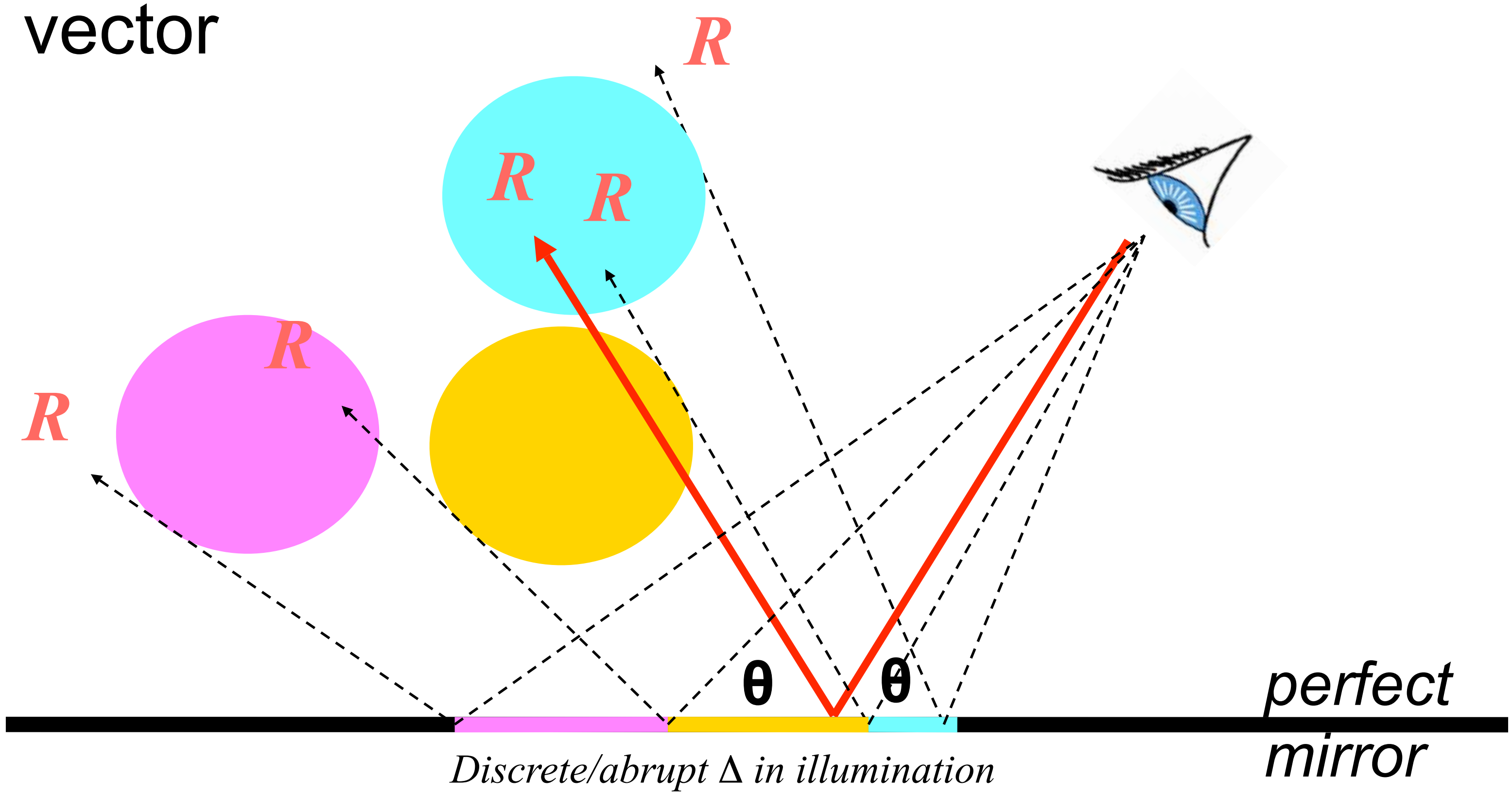




Allen Martin - http://www.cs.wpi.edu/~matt/courses/cs563/talks/dist_ray/dist.html

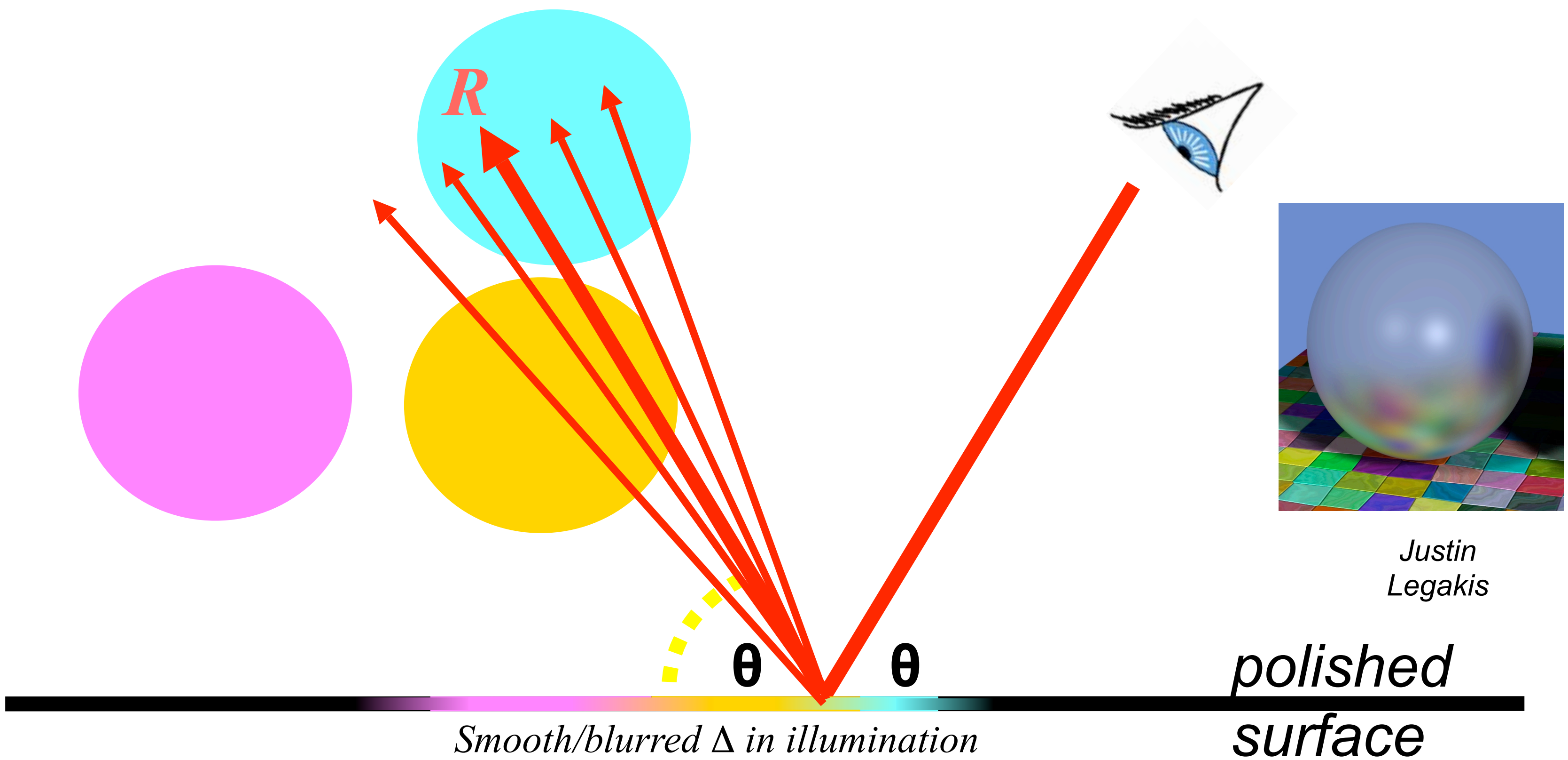
DRT: Glossy Reflections

- *Problem*: Mirror-like reflections
  - Contribution only comes from the reflection vector



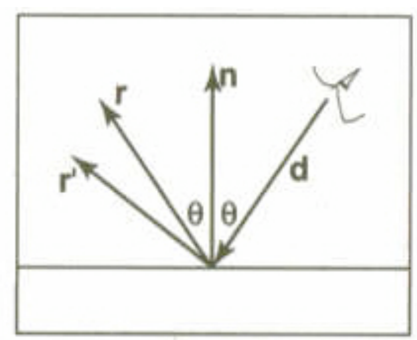
DRT: Glossy Reflections

- *Solution*: Glossy ("blurred") reflections
  - Integrate over additional rays defined about the reflection vector



DRT: Glossy Reflection

- Sampling: define a population of rays about r
  - Define each ray r' as a perturbation from r
  - To do this:
    - create an orthonormal uvw basis with w = r
    - create a random point in the 2D square with side length a centered at the origin
    - create u,v: $u = -a/2 + \varepsilon a$; $v = -a/2 + \varepsilon' a$ with random ε and ε' in [0,1]
    - Then $\mathbf{r}' = \mathbf{r} + u\,\mathbf{u} + v\,\mathbf{v}$



Sampling: A Population of Reflection

- Define a tangent plane to ray *R*
  - Let vectors ***u*** and ***v*** be orthonormal vectors that are perpendicular to ray *R*

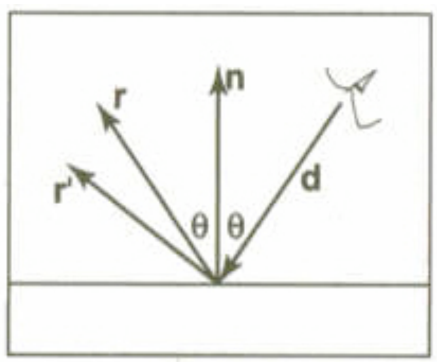


Figure 10.36. The reflection ray is perturbed to a random vector **r**'.

$$\mathbf{r}' = \mathbf{r} + u\mathbf{u} + v\mathbf{v}.$$

$$u = -\frac{a}{2} + \xi a,$$

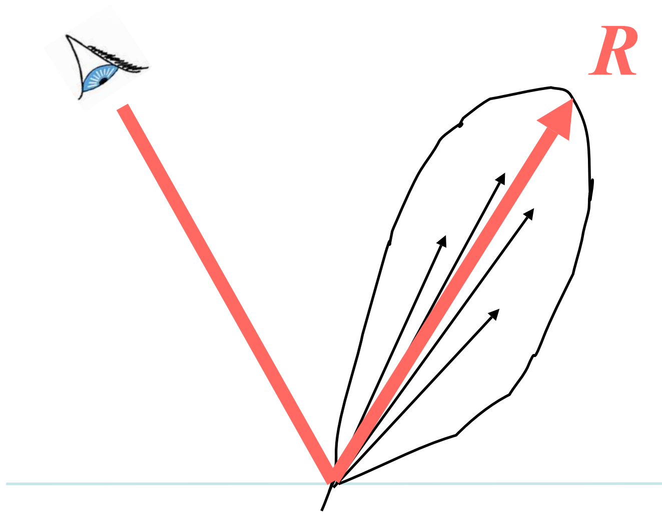
$$v = -\frac{a}{2} + \xi' a.$$

a – blur control

ξ - random value

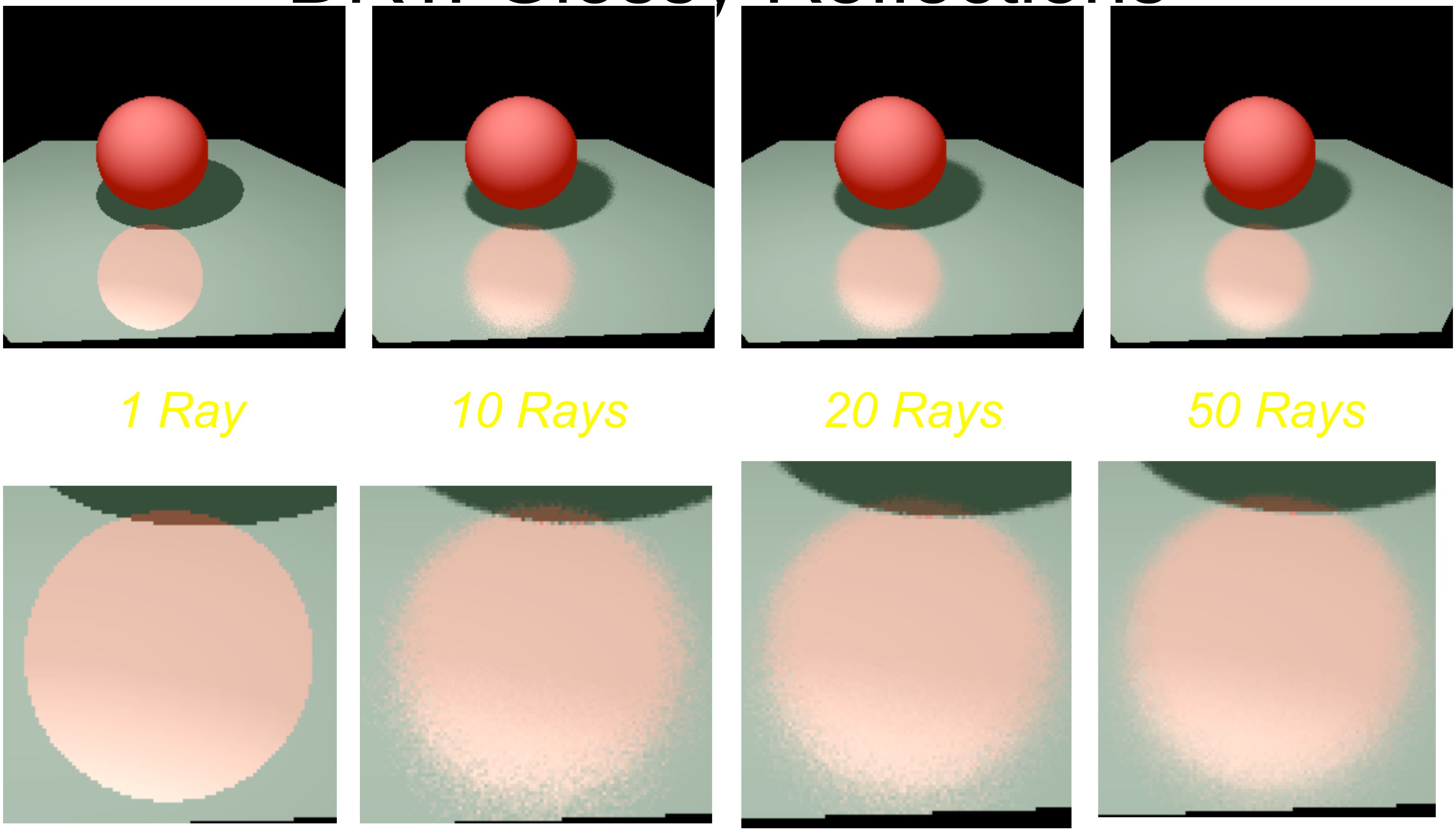
Integrate Over the Population of Reflection

- Let's utilize the same function used when determining specular highlight intensity
- Weight the each ray *R*' according to a lobe, i.e. the cosine of the angle between *R* and *R*'



Glossy Reflection

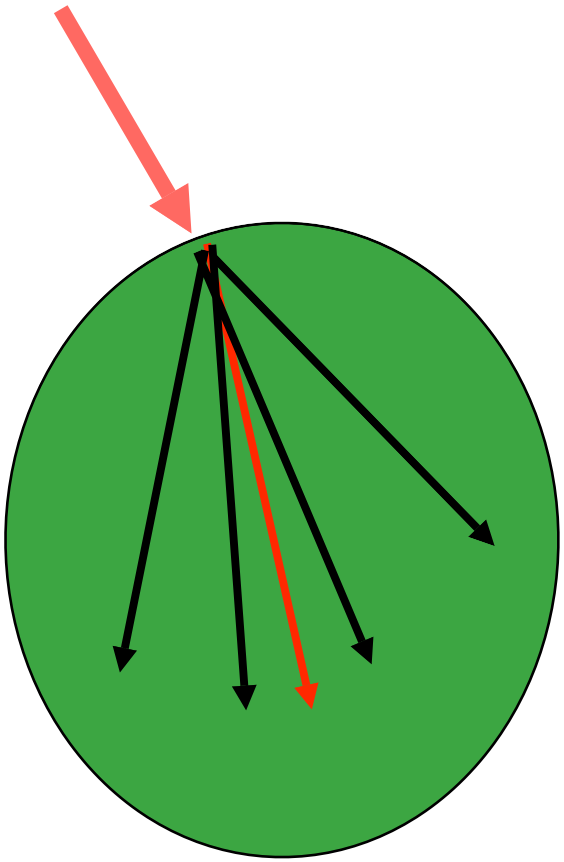
DRT: Glossy Reflections



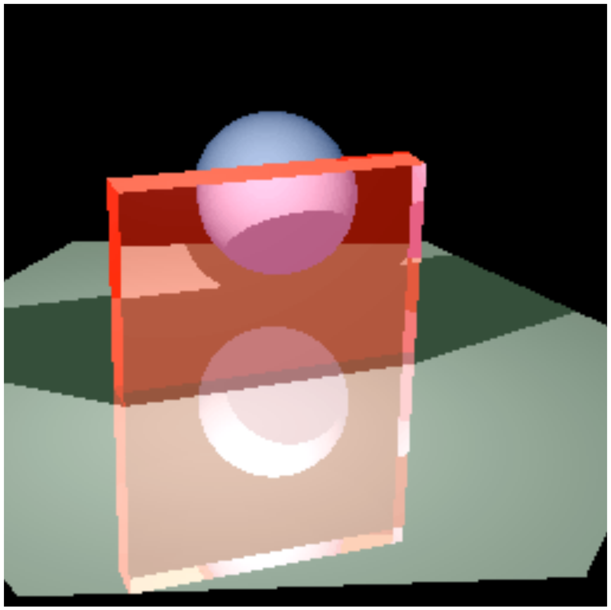
Allen Martin - http://www.cs.wpi.edu/~matt/courses/cs563/talks/dist_ray/dist.html

DRT: Translucency

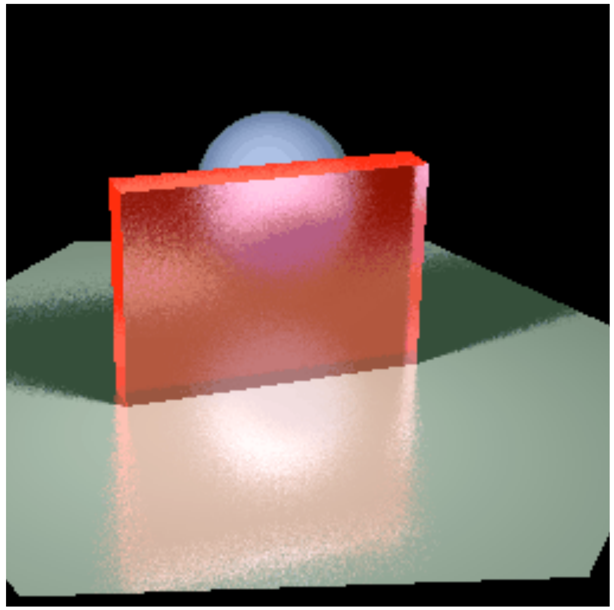
- *Solution*: Same solution as glossy reflection, except use the transmittance vector *T* and integrate over the hemisphere behind the surface



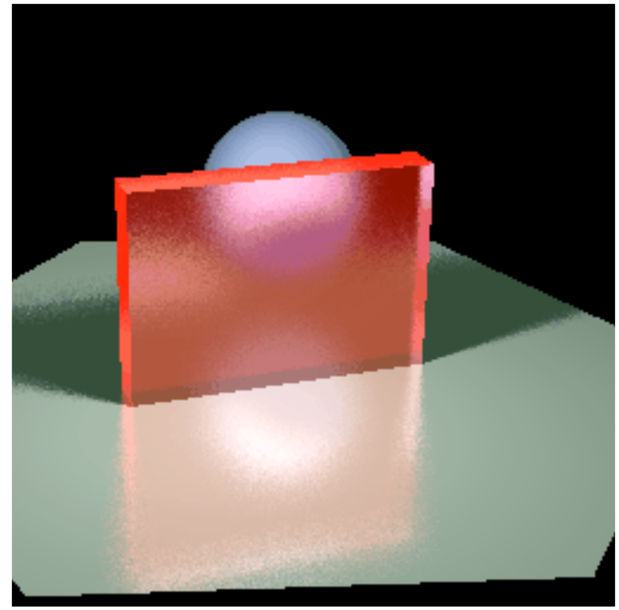
DRT: Translucency



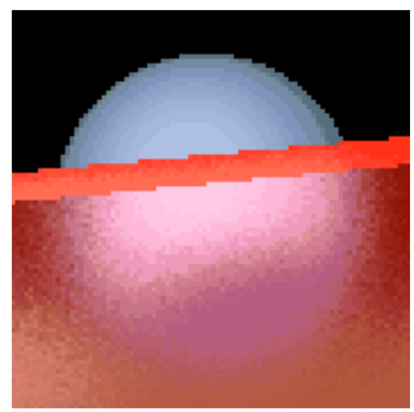
1 Ray



10 Rays



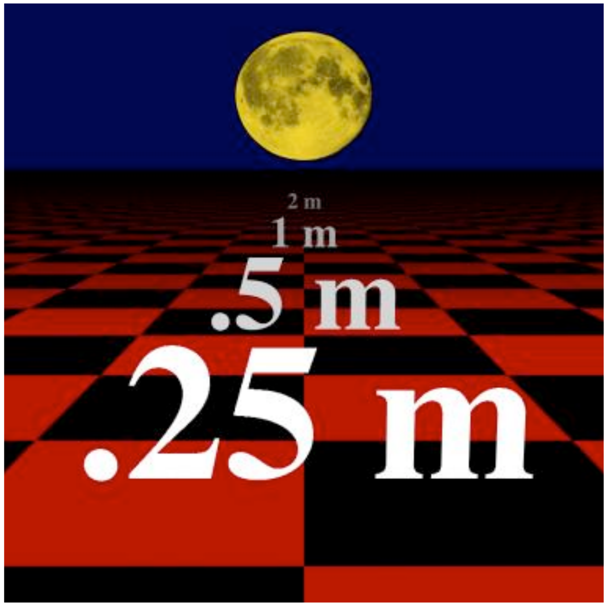
20 Rays



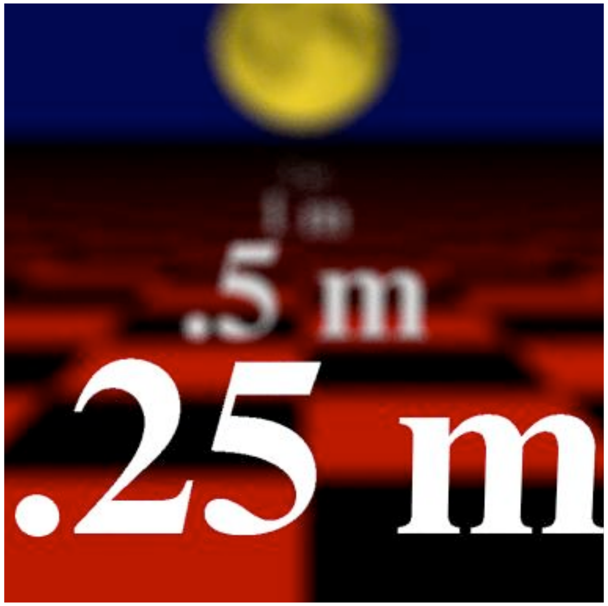
Allen Martin - http://www.cs.wpi.edu/~matt/courses/cs563/talks/dist_ray/dist.html

DRT: Depth Of Field

- *Problem*: Pinhole camera model keeps the entire scene in focus



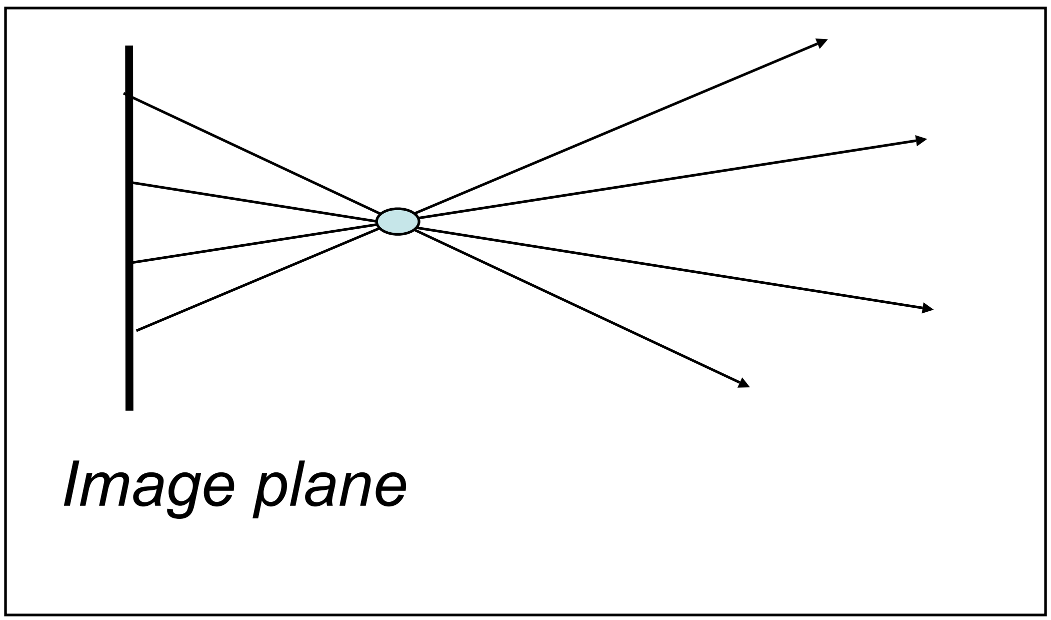
Pinhole camera

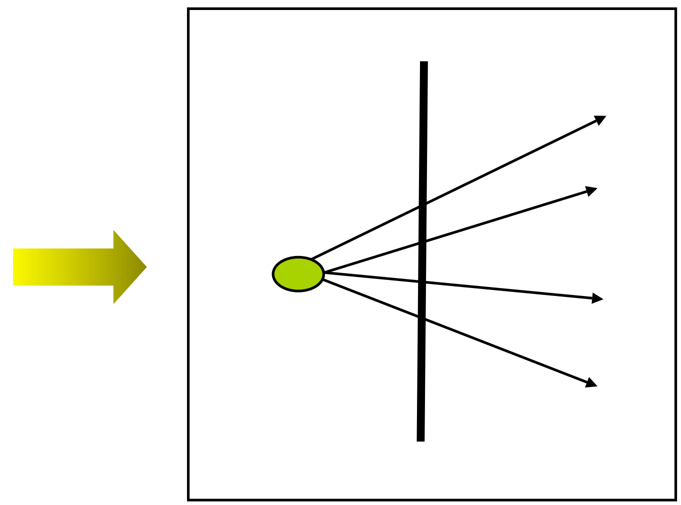


Depth of Field

Pinhole Camera

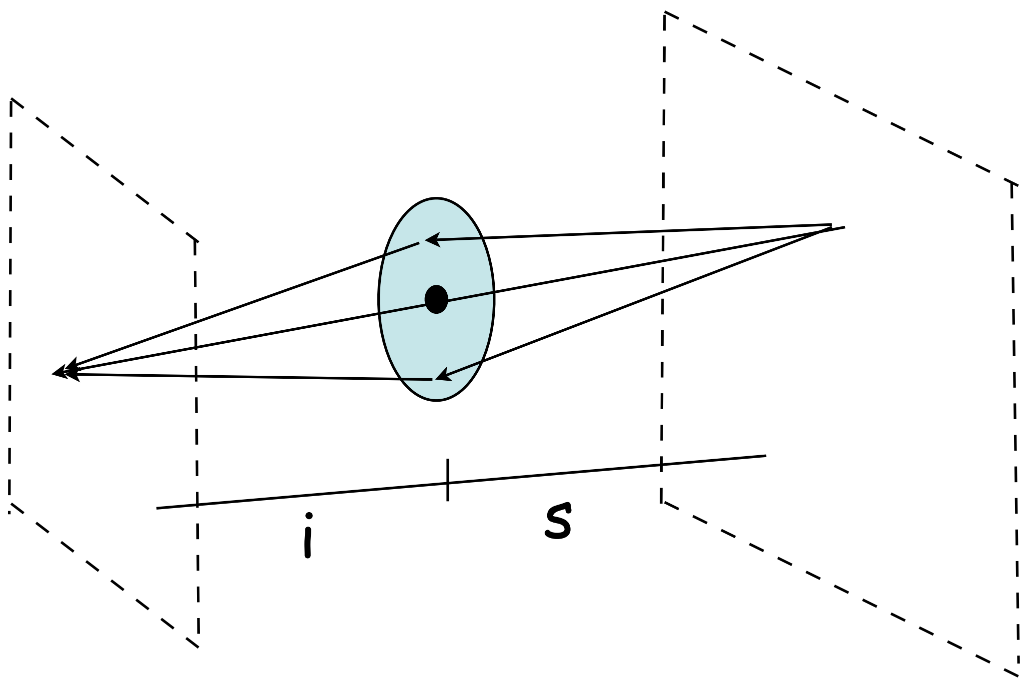
- When using pinhole camera, the “lens” is just a point to project light from the scene onto the image plane





Thin-lens Camera

- Depth-of-field can be simulated using a thin-lens camera
- A thin-lens camera replace the pinhole by a disk-shaped thin-lens

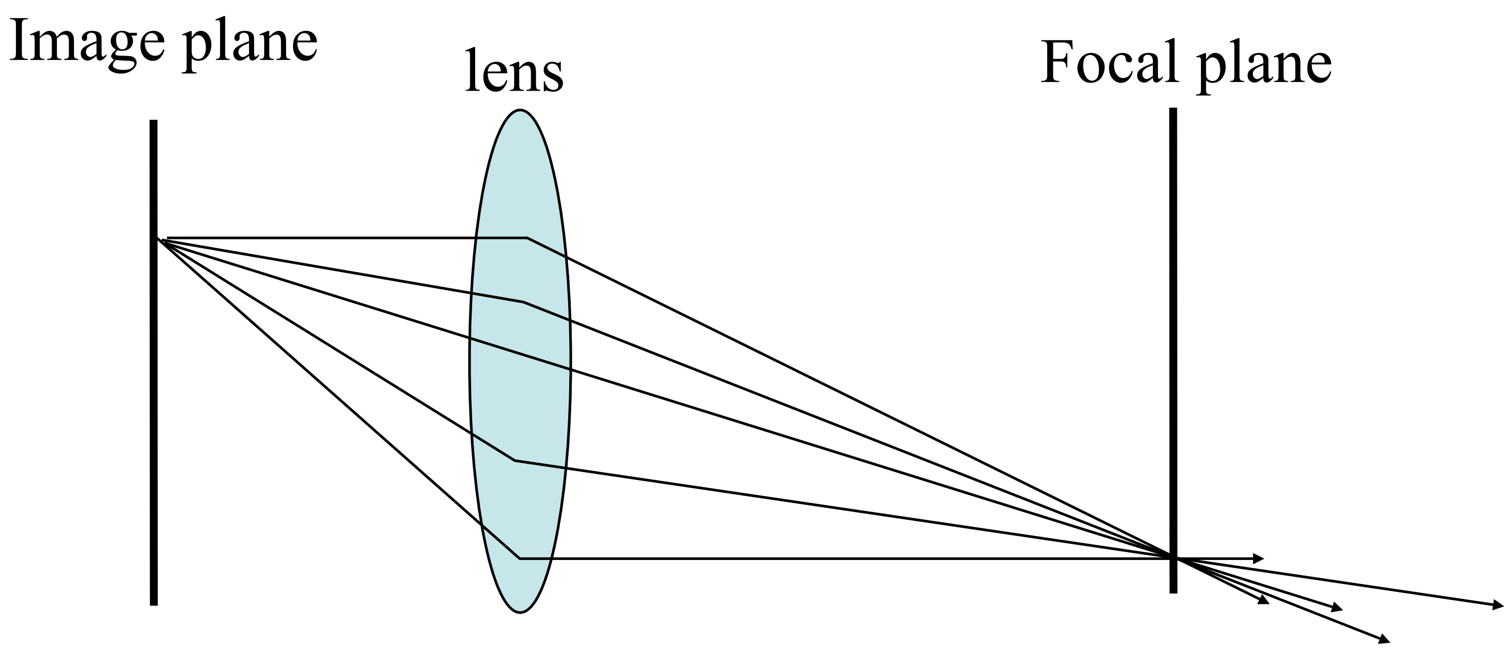


1/s+1/i = 1/f
f: focal length

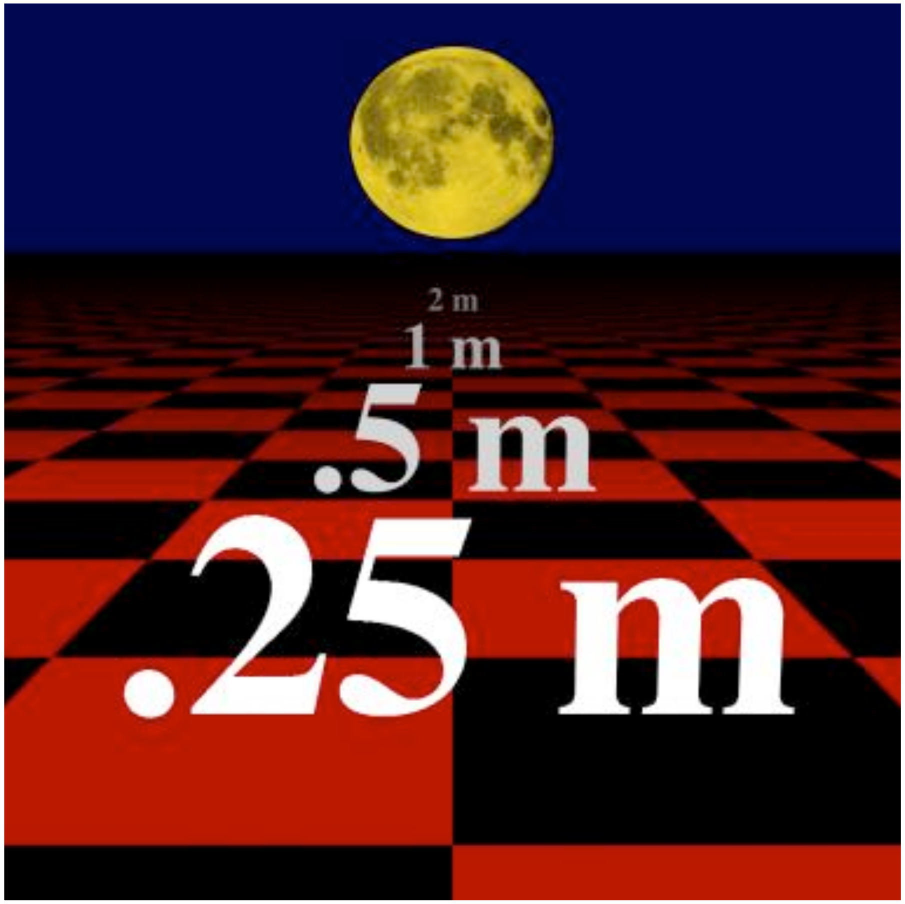
when s = infinity
f = i

Lens Model

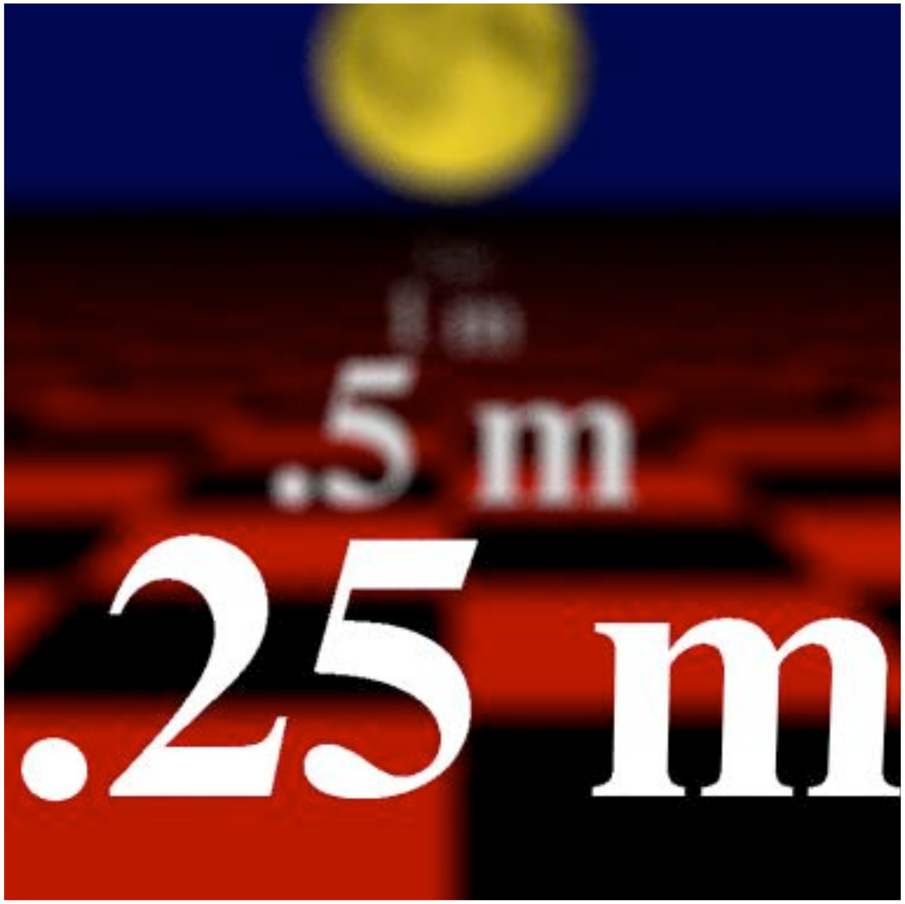
- A lens lets in more light into the camera



Changing the Focal Length



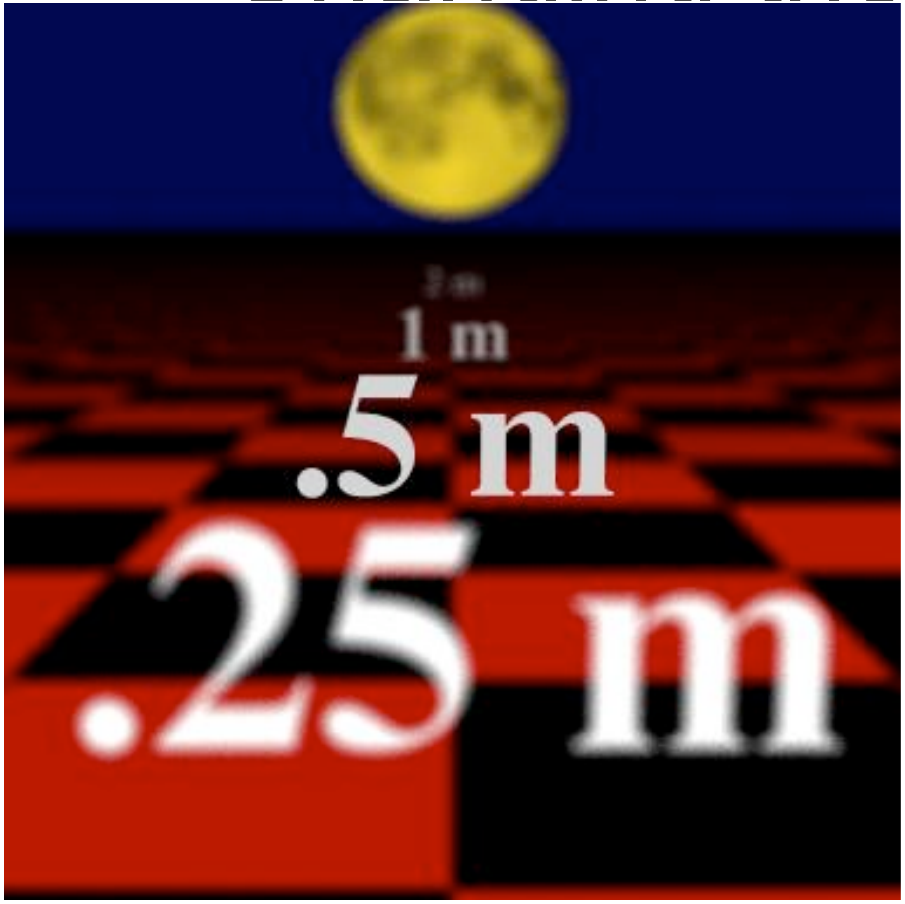
Pinhole Camera



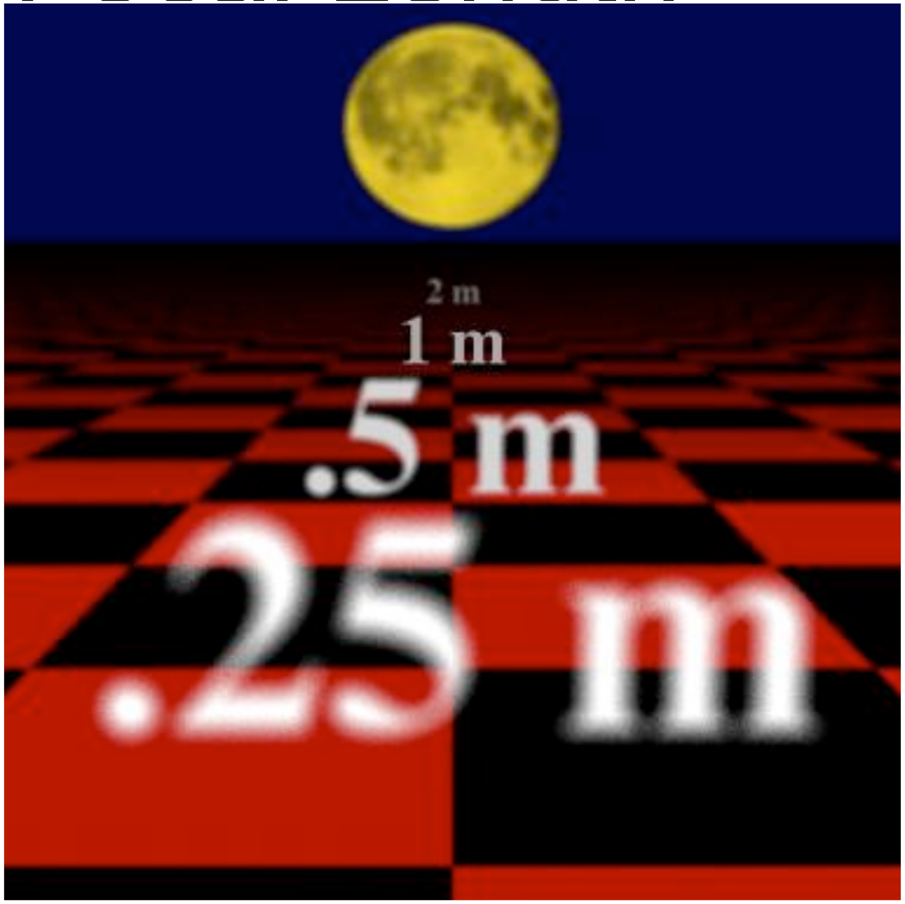
.25 m Focal Length

Mike Stark - http://www.cs.utah.edu/~shirley/classes/cs684_98/students/mstark/hw4/hw4.html

Changing the Focal Length

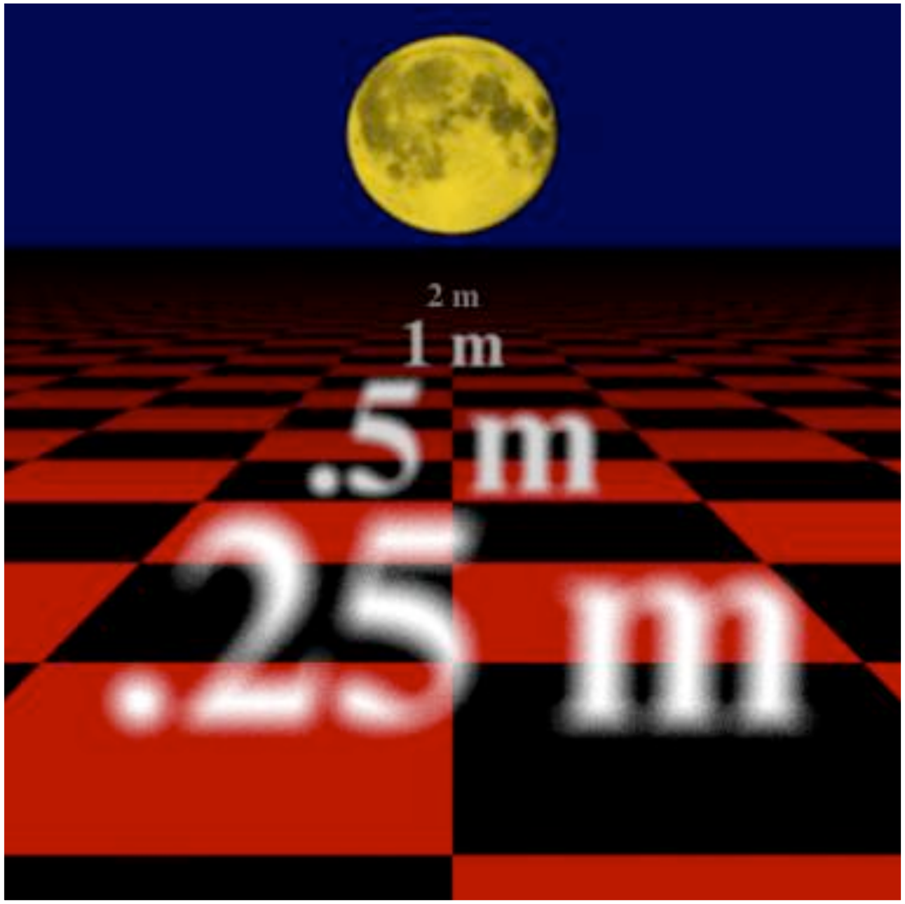


0.5 m Focal Length

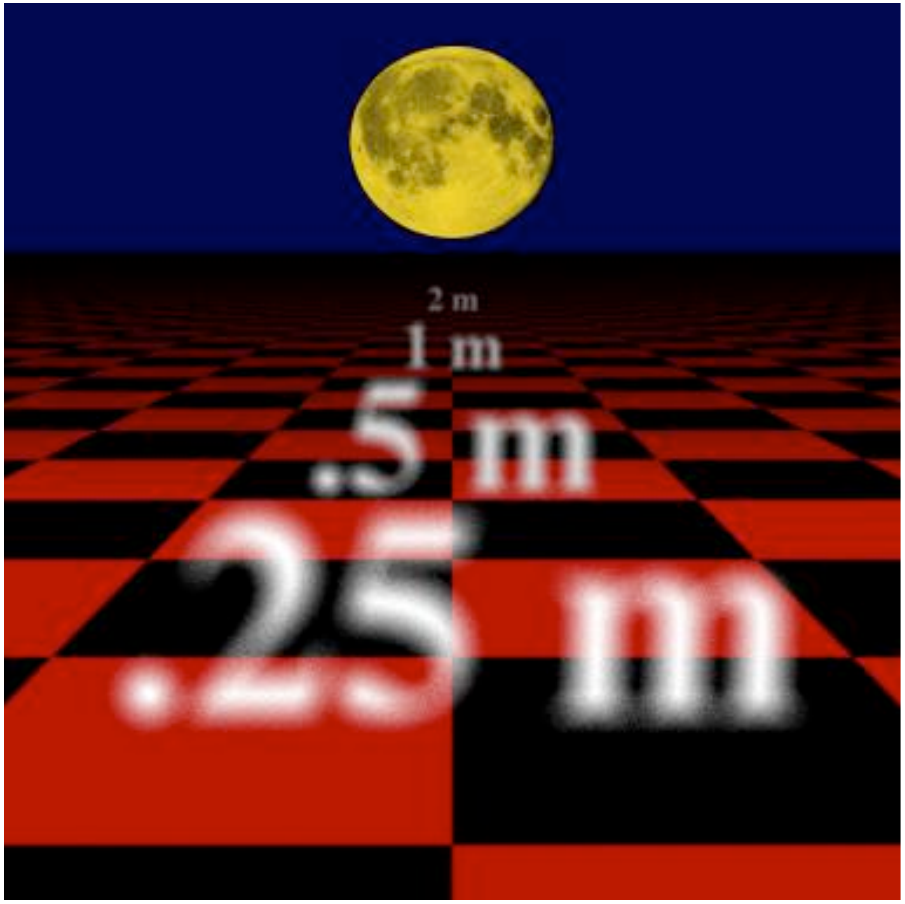


1 m Focal Length

Changing the Focal Length



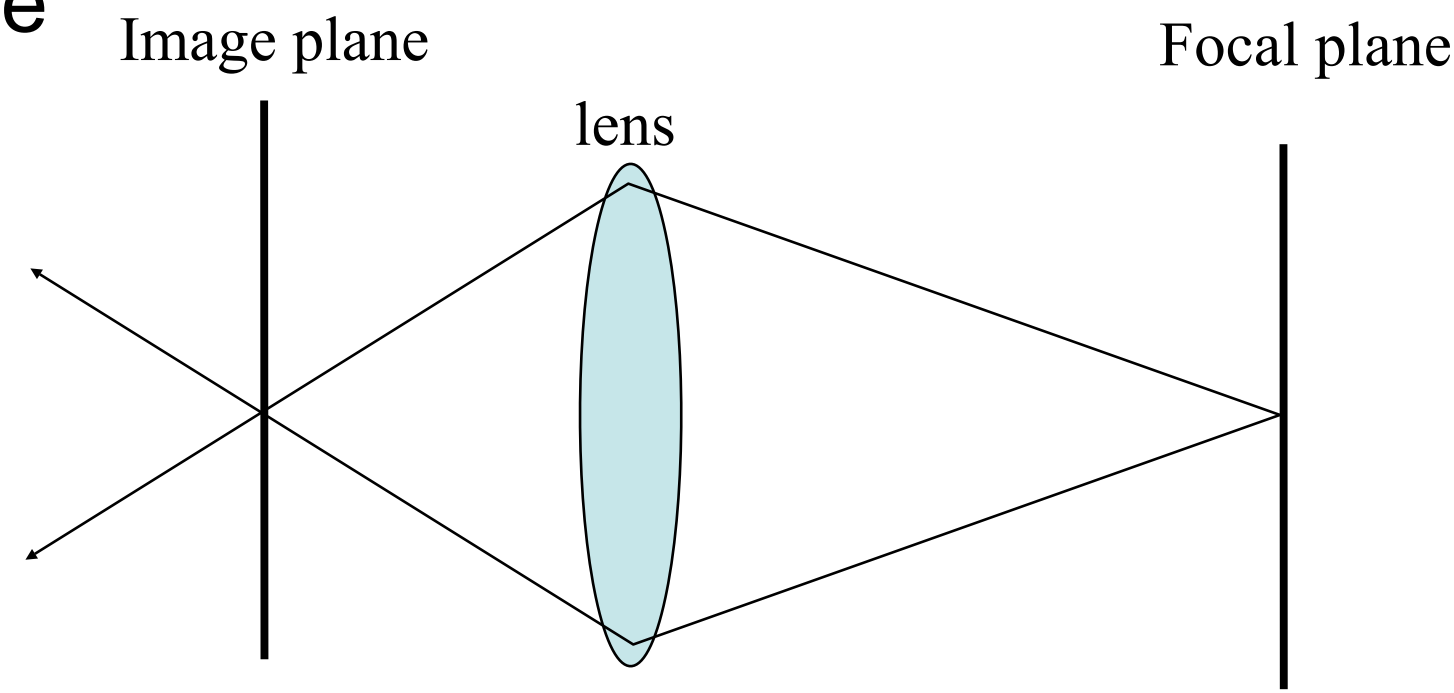
2 m Focal Length



Infinite Focal Length

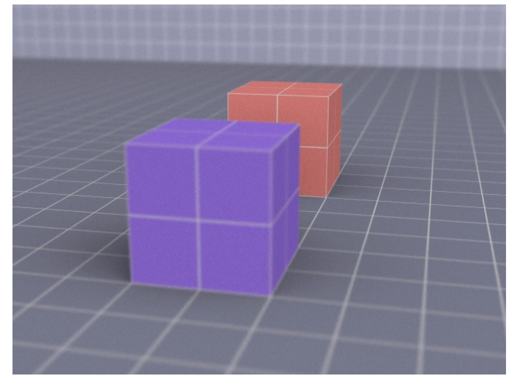
Circle Of Confusion

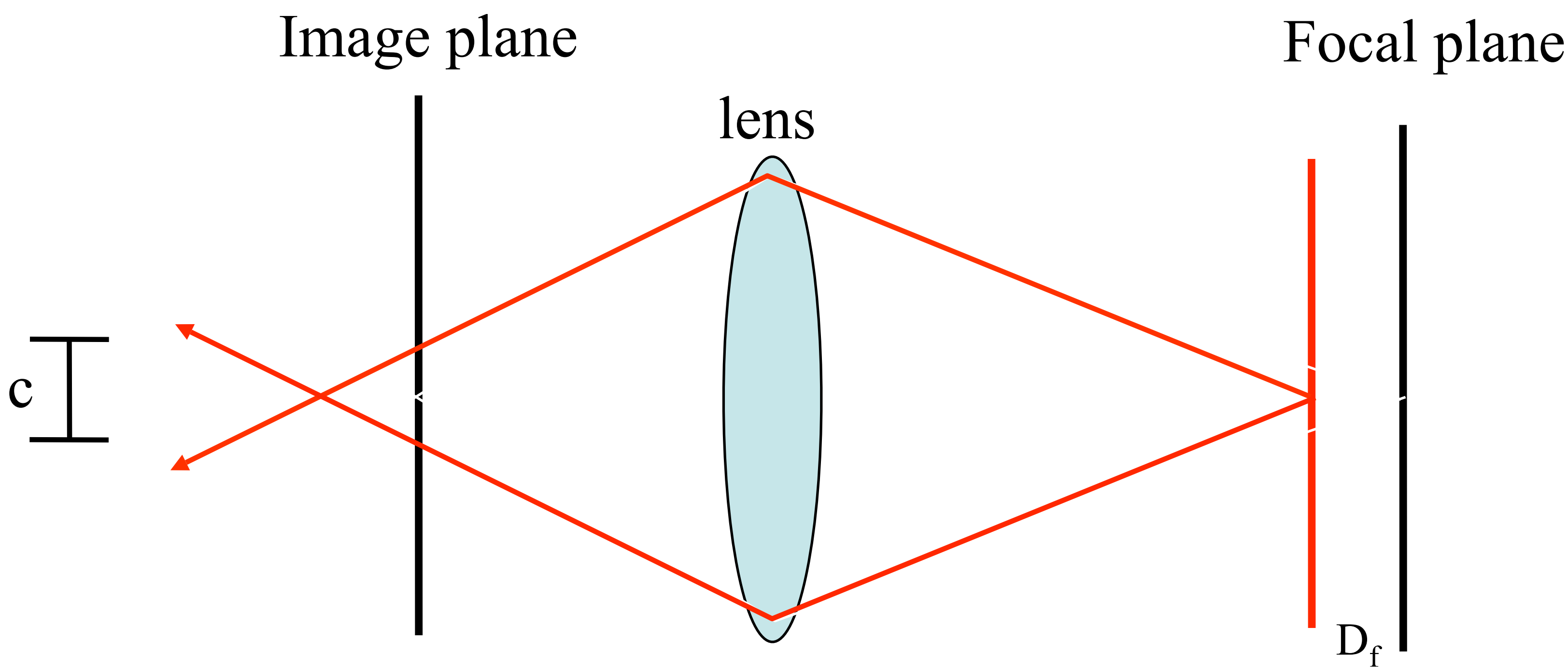
- The circle of confusion determines a scene point's contribution to the image plane



Circle of Confusion: Out-of-focus

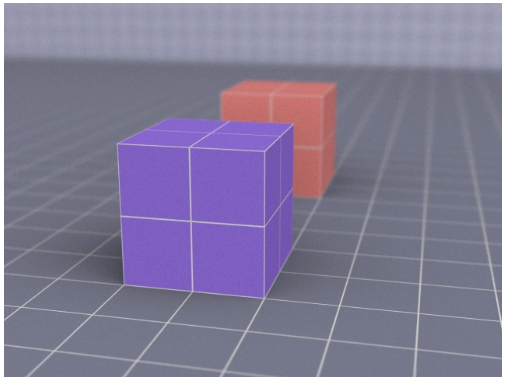
- Closer object

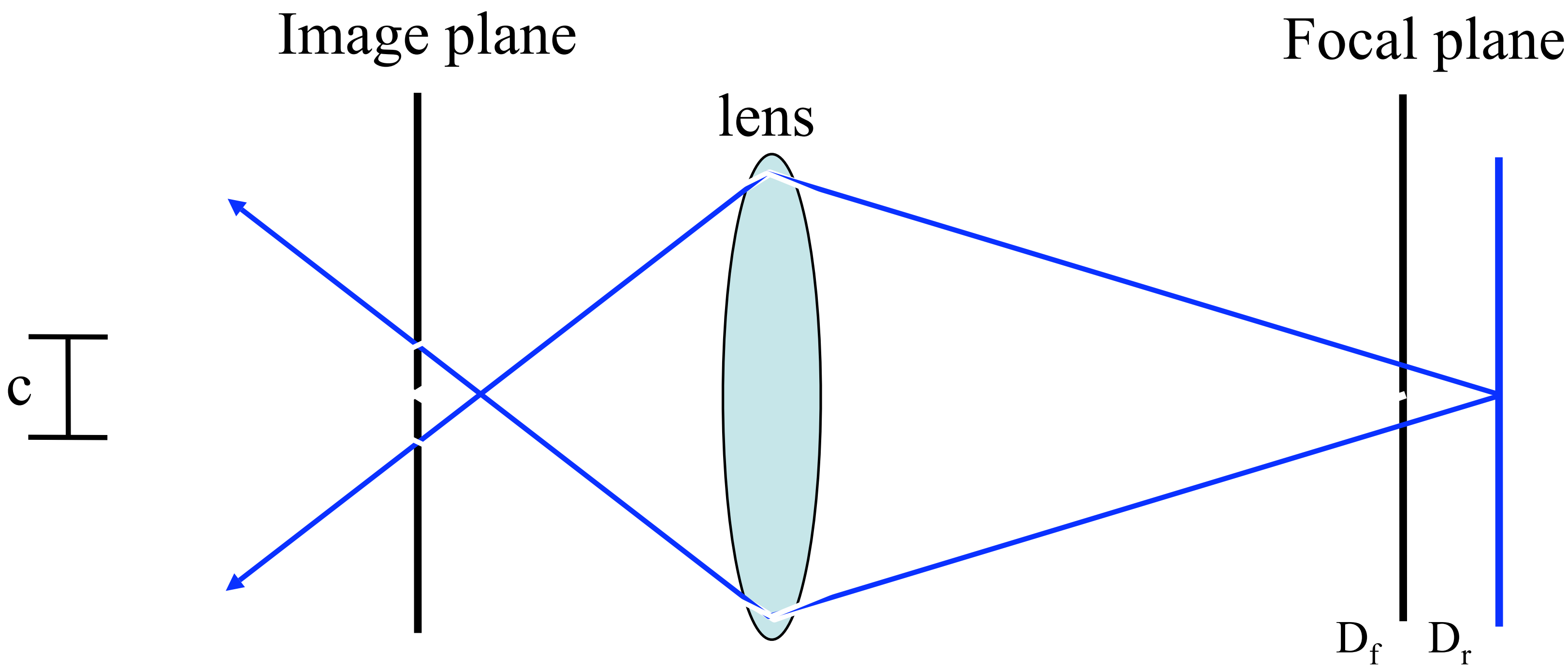




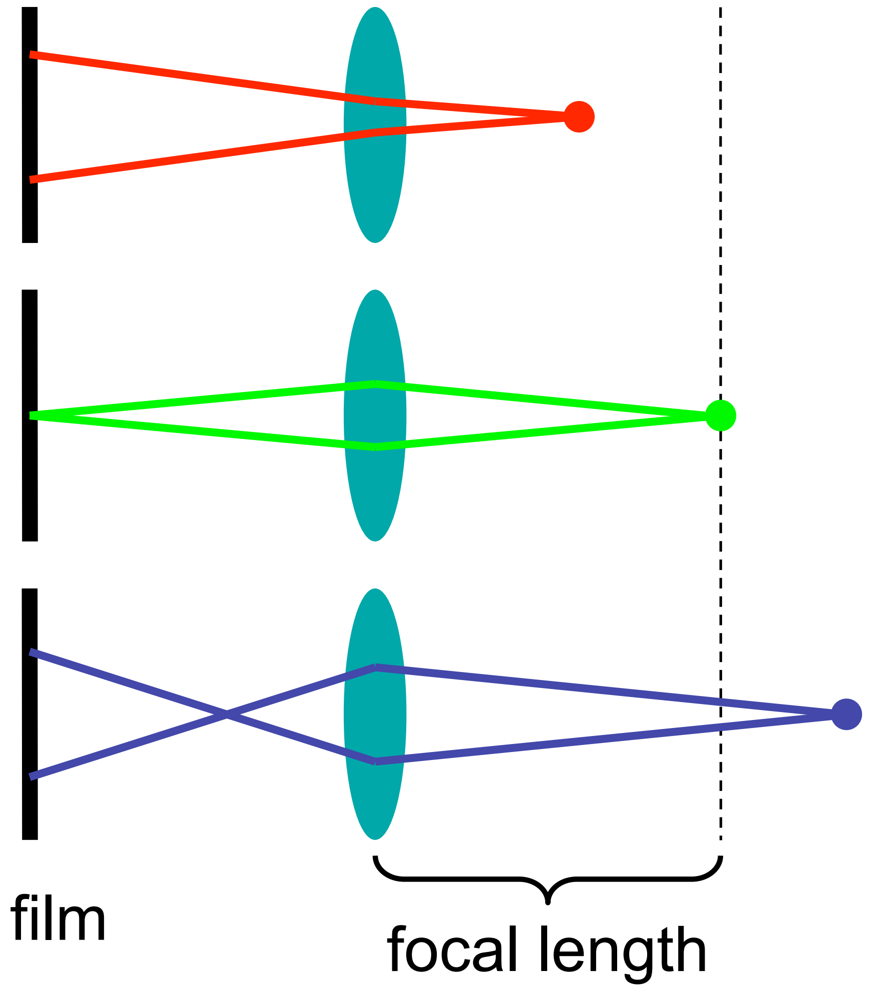
Circle of Confusion: Out-of-focus

- Further object



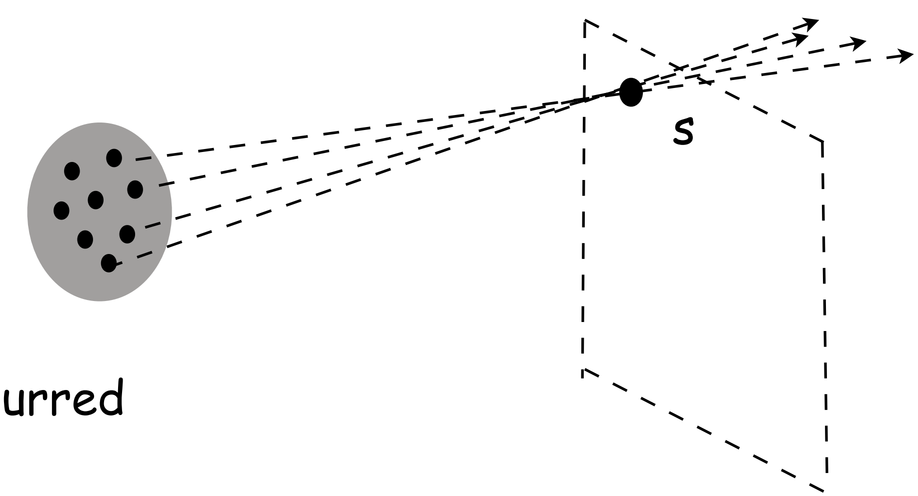


Summary



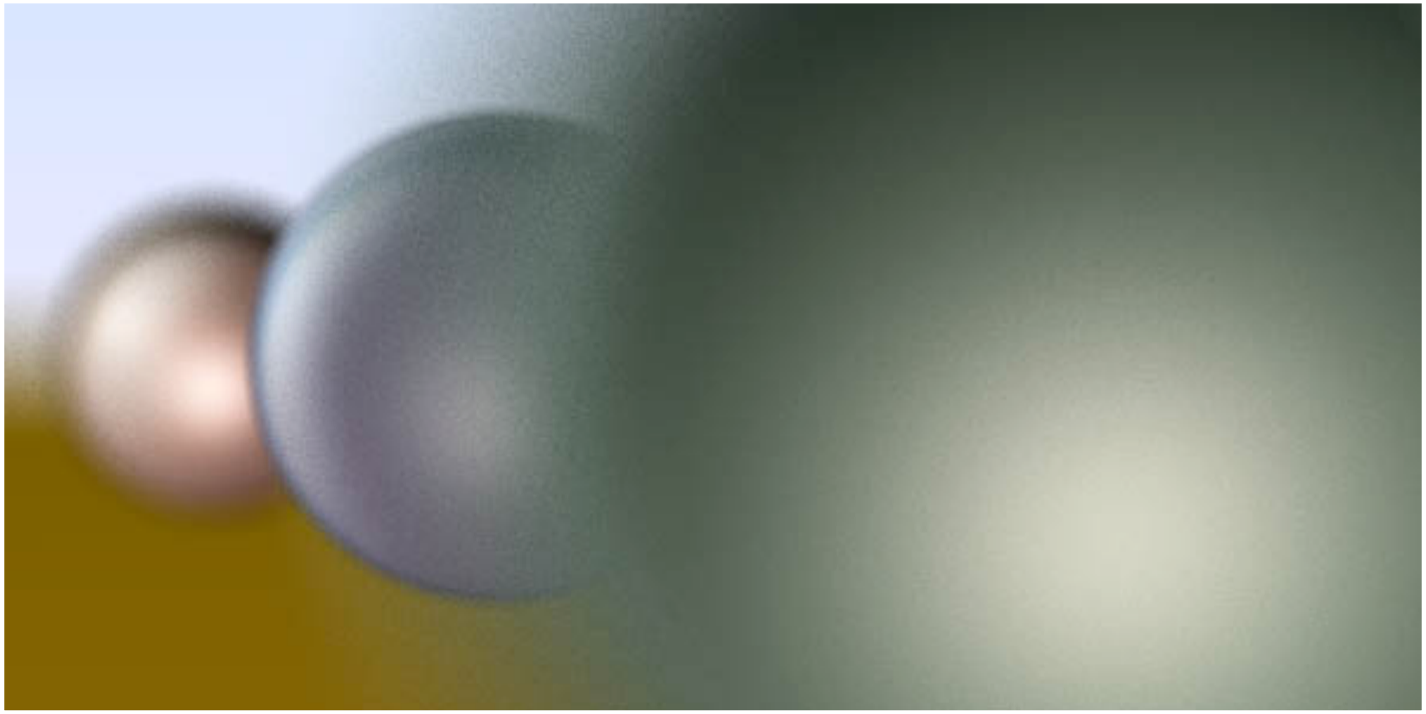
Implementation

- Place your image *S* distance away, where you have the complete focus
- Assume the radius of the lens is R, for each pixel, randomly select N points within a disk around the camera (the disk is perpendicular to the camera view direction). Use those N points as your camera position and shoot rays
- Average the N colors from the rays and assign it to the pixel



For objects at the focal plane, the jittered camera positions have no effect. Other objects will become blurred

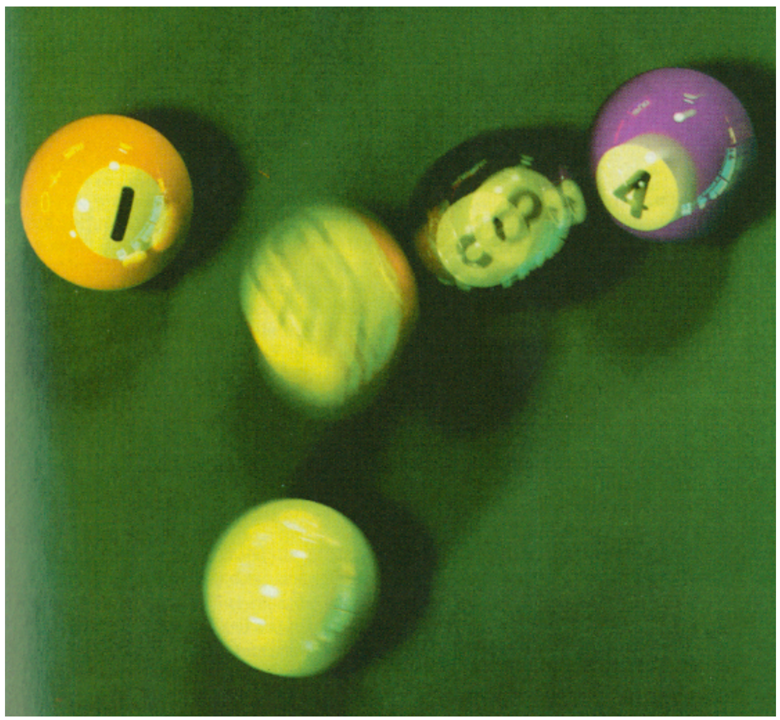
Depth Of Field Example



Vince Scheib - http://www.cs.unc.edu/~scheib/school/238/imoire/index.html

DRT: Motion Blur

- *Problem*: Object (or camera) motion requires an exposure (samples over time or shutter speed) rather than a single sample in time



DRT: Motion Blur

- *Solutions*

- Quick fix: Post-process blurring (i.e. render and blur in 2D)

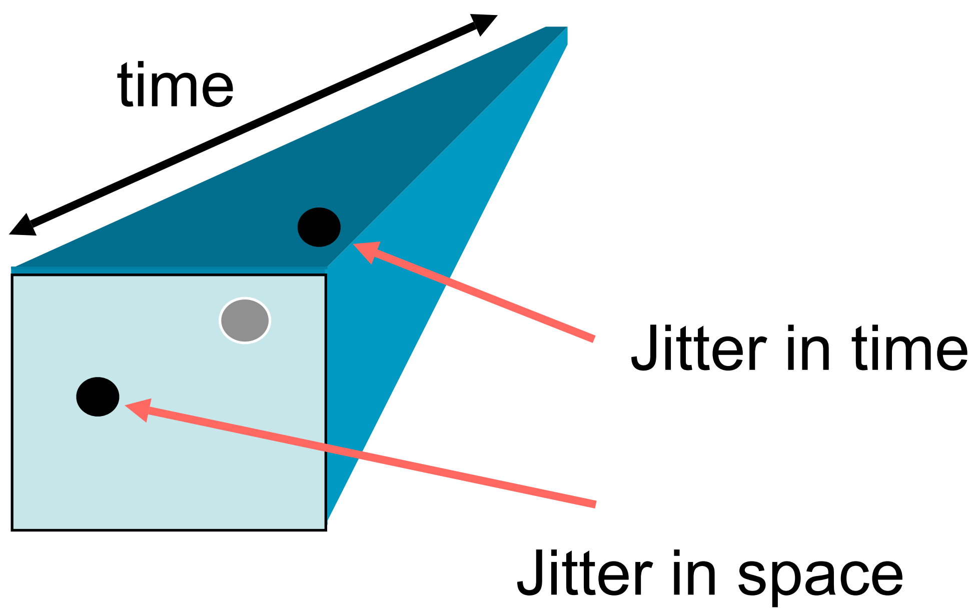
  - Two objects moving so that one always obscures the other
    - Can’t render and blur objects separately

  - A spinning top with texture blurred but highlights sharp
    - Don’t want to blur the highlight

  - The blades of a fan creating a blurred shadow
    - Must consider the movement of other objects

DRT: Motion Blur

- *Solutions* … contd.
- Sample objects temporally
  - Distribute rays over time
  - $T = T_0 + \xi(T_1 - T_0)$

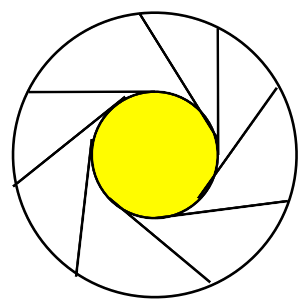


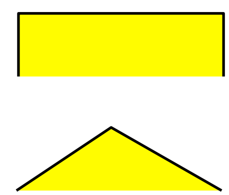
Shutter Functions

- Supersample on each pixel
- Sample the scene at different time samples
- What Reconstruction Filter should we use?
  - The reconstruction filter controls the shutter speed length
  - Box filter – fast shutter
  - Triangle filter – slow shutter

$$I(x, y, t) = r(t) * f(x, y, t) s(t)$$

shutter function | animated continuous image function | temporal samples





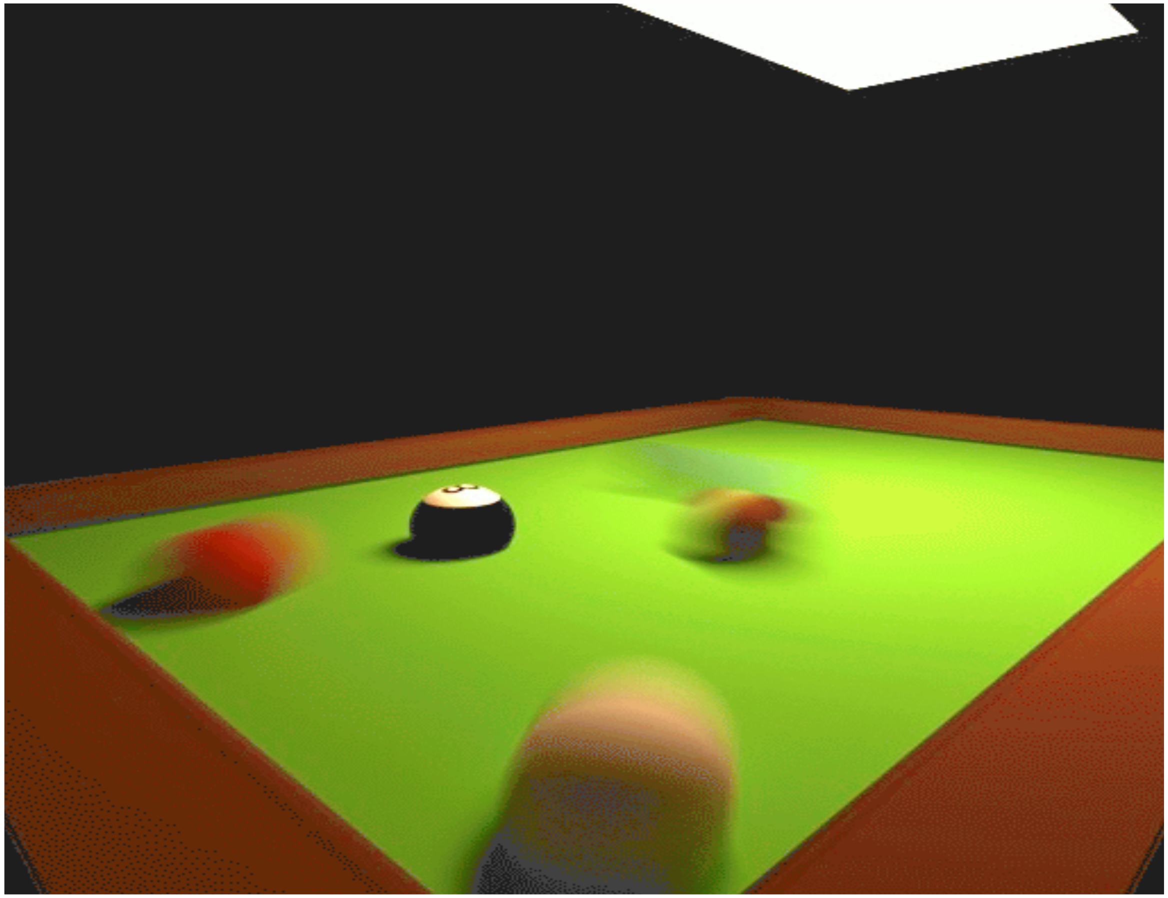
Temporal Jittered Sampling

- Stochastically sample in the time domain as well as in the spatial domain

7	11	3	14
4	15	13	9
16	1	8	12
6	10	5	2

Another Example

400
samples
per pixel



Greg Coombe - http://www.cs.unc.edu/~coombe/cs6620/2.5/prog10.html