12. Anti-aliasing and Distribution Ray Tracing

Reading

Required:

- Watt, sections 12.1-12.5.1.

Further reading:

- Watt, chapter 14.
- A. Glassner. An Introduction to Ray Tracing. Academic Press, 1989. [In the lab.]

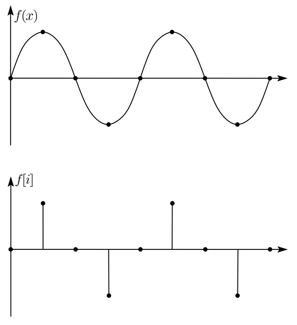
Aliasing

Ray tracing is a form of sampling and can suffer from annoying visual artifacts...

Consider a continuous function f(x). Now sample it at intervals Δ to give $f[i]$ = quantize[$f(i\Delta)$].

Q: How well does $f[i]$ approximate f(x)?

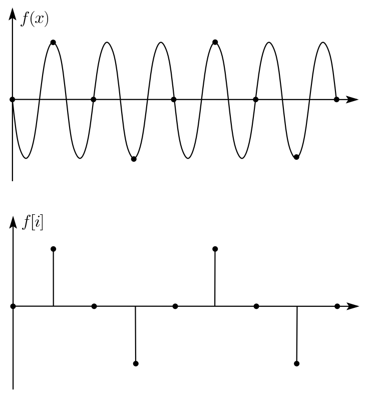
Consider sampling a sinusoid:



In this case, the sinusoid is reasonably well approximated by the samples.

Aliasing (con't)

Now consider sampling a higher frequency sinusoid

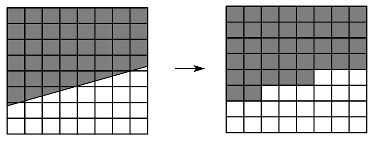


We get the exact same samples, so we seem to be approximating the first lower frequency sinusoid again.

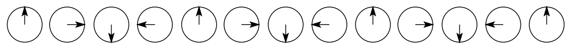
We say that, after sampling, the higher frequency sinusoid has taken on a new "alias", i.e., changed its identity to be a lower frequency sinusoid.

Practical examples of aliasing

Drawing a polygon into the frame buffer:



Temporal aliasing:

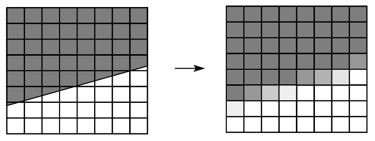


Anti-aliasing

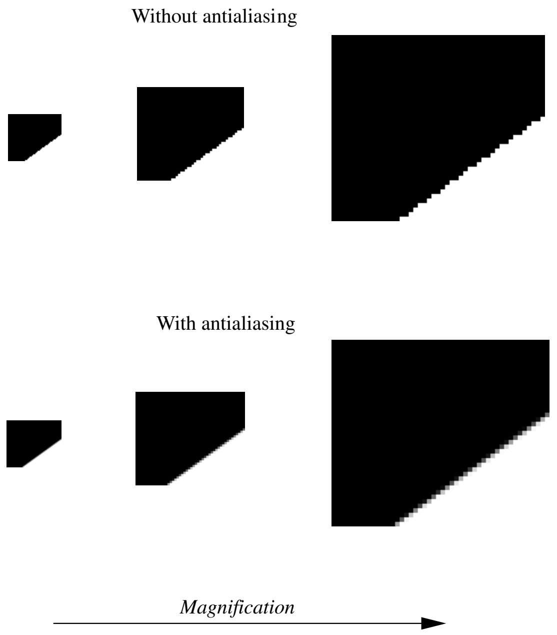
Q: How do we avoid aliasing artifacts?

1. Sampling:
2. Filtering:
3. Combination:

Example - polygon revisited:



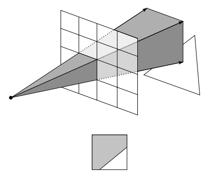
Polygon anti-aliasing

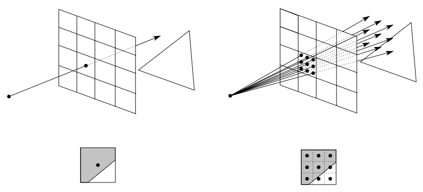


Q:What about temporal aliasing?

Antialiasing in a ray tracer

We would like to compute the average intensity in the neighborhood of each pixel.





When casting one ray per pixel, we are likely to have aliasing artifacts.

To improve matters, we can cast more than one ray per pixel and average the result.

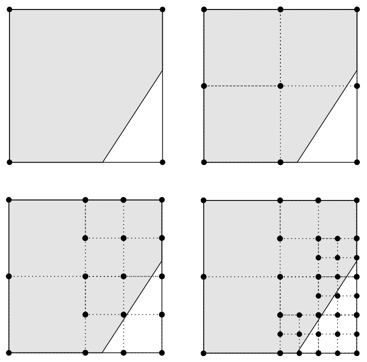
A.k.a., **super-sampling and averaging down**.

Antialiasing by adaptive sampling

Casting many rays per pixel can be unnecessarily costly.

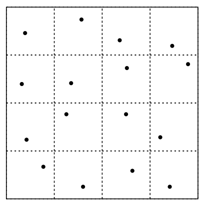
For example, if there are no rapid changes in intensity at the pixel, maybe only a few samples are needed.

Solution: **adaptive sampling**.



Q: When do we decide to cast more rays in a particular area?

Distribution ray tracing

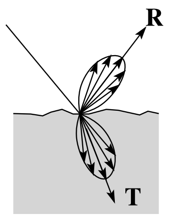


Idea:

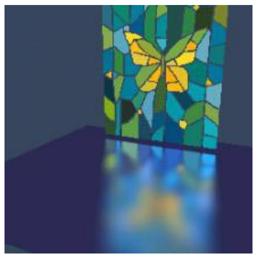
- Use non-uniform (jittered) samples.
- Replaces aliasing artifacts with noise.
- Provides additional effects if we distribute rays in other dimensions:
  - Reflection and refractions
  - Light source area
  - Camera lens area
  - Time

Originally called "distributed ray tracing," but we will call it **distribution ray tracing** so as not to confuse with parallel computing.

DRT to simulate ____________________



Distributing rays over reflection and/or refraction directions gives:



DRT pseudocode

TraceImage() looks basically the same, except now each pixel records the average color of jittered sub-pixel rays.

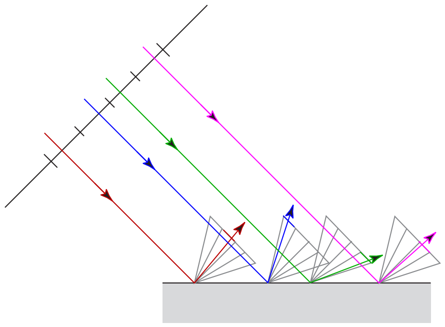
```
function traceImage (scene):
    for each pixel (i, j) in image do
        I(i, j) ← 0
        for each sub-pixel id in (i,j) do
            s ← pixelToWorld(jitter(i, j, id))
            p ← COP
            d ←(s - p).normalize()
            I(i, j) ← I(i, j) + traceRay(scene, p, d, id)
        end for
        I(i, j) ← I(i, j)/numSubPixels
    end for
end function
```

A typical choice is numSubPixels = 4*4.

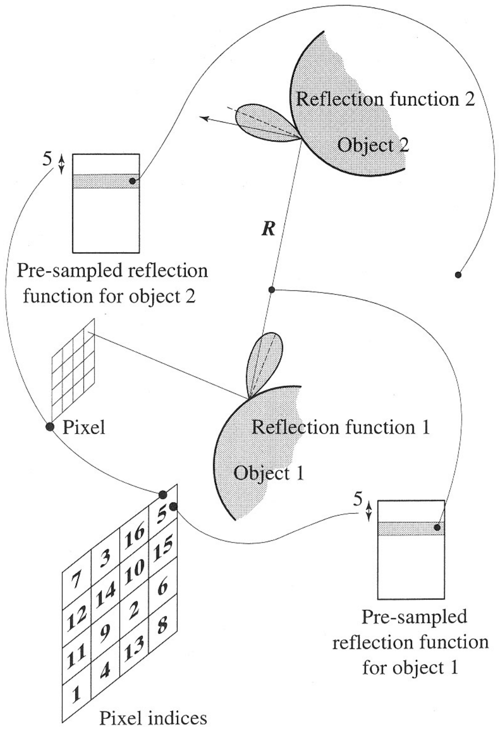
DRT pseudocode (cont'd)

Now consider *traceRay*(), modified to handle (only) opaque glossy surfaces:

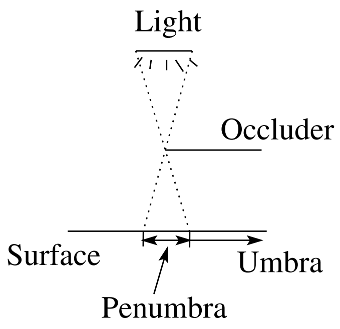
```
function traceRay(scene, p, d):
    (q, N, material) ← intersect (scene, p, d)
    I ← shade(…)
    R ← jitteredReflectDirection(N, -d, id)
    I ← I + material.k_r * traceRay(scene, q, R)
    return I
end function
```



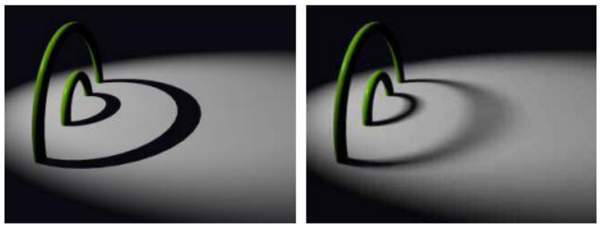
Pre-sampling glossy reflections



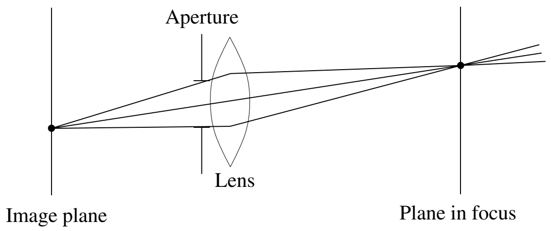
DRT to simulate ___________________



Distributing rays over light source area gives:



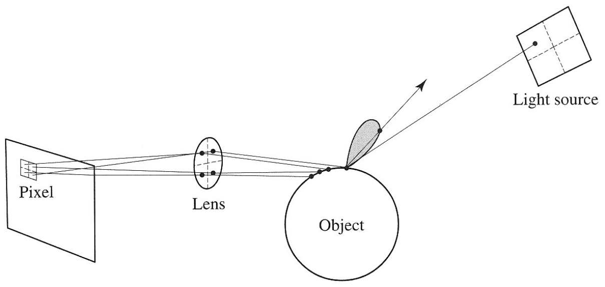
DRT to simulate ___________________



Distributing rays over a finite aperture gives:

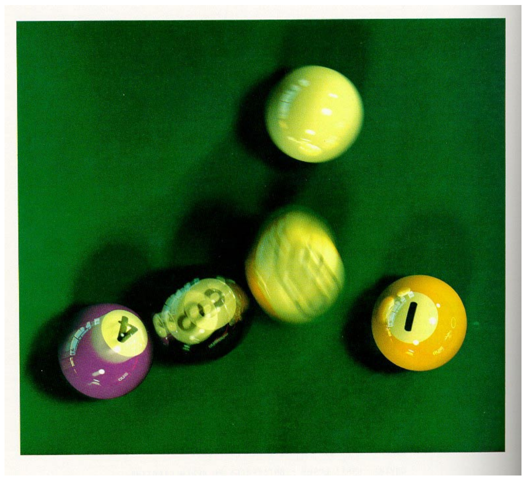
Chaining the ray id's

In general, you can trace rays through a scene and keep track of their id's to handle *all* of these effects:



DRT to simulate ___________________

Distributing rays over time gives:



Summary

What to take home from this lecture:

1. The meanings of all the boldfaced terms.
2. An intuition for what aliasing is.
3. How to reduce aliasing artifacts in a ray tracer
4. How distribution ray tracing works and what effects it can simulate.