CS130 : Computer Graphics
Lecture 20: Ray Tracing

Tamar Shinar
Computer Science & Engineering
UC Riverside
Ray Tracing
shallow depth of field, area light sources, diffuse interreflection
up to 16 reflections per ray

Greg L., Wikimedia Commons
for each pixel

1. **cast view ray**: compute view ray from camera through pixel into scene
2. **intersect**: find intersection of ray with closest object
3. **shade**: compute the color of the intersection point
Ray Tracing Program

for each pixel do
    compute viewing ray
    if ( ray hits an object with t in [0, inf] ) then
        compute n
        evaluate shading model and set pixel to that color
    else
        set pixel color to the background color
Recursive ray tracing

\[ \text{ray} = \text{ray}(e, d, t_0, t_{\text{max}}) \]

**function** \( \text{ray\_color}(\text{ray}) \)

\[ \text{if} \ (\text{Intersection}(\text{ray})) \ \text{then} \]
\[ \quad \text{point} = \text{ray\_Point}(\text{ray\_t\_max}) \]
\[ \quad \text{color} \ c = \text{color\_ambient} \]
\[ \quad \text{if} \ (! \ \text{Intersection}(\text{ray(point, l, eps, inf)})) \]
\[ \quad \quad \ h = \text{halfway\_vector} \]
\[ \quad \quad \ c = c + \text{color\_diffuse} + \text{color\_specular} \]
\[ \quad \quad \ c = c + k_m \ * \ \text{ray\_color}(\text{ray(point, r, eps, inf)}) \]
\[ \text{else} \]
\[ \quad \text{color} \ c = \text{background\ color} \]
class Surface
{
    public:
        void Intersection(RAY& ray)=0;
        Box Bounding_Box()=0;
    }

Other objects: Ray, Light, Material, Camera, Film, World