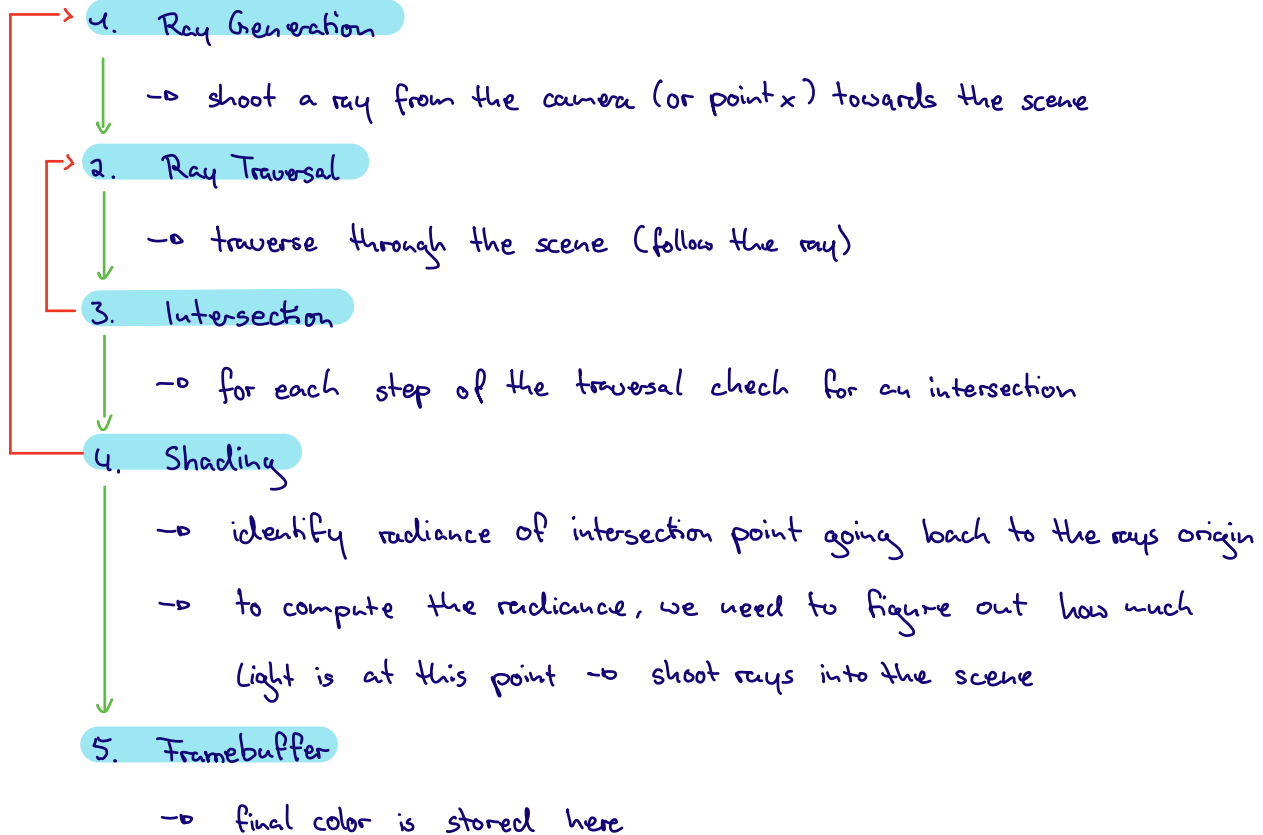


# Basic Ray-Tracing (Algorithm)

## Ray Tracing Pipeline



## Ray Types

- primary ray: ray from camera towards scene
- secondary ray: reflected or refracted ray
- shadow ray: ray directly towards the lightsource
  - check if point is occluded or illuminated

⇒ common raytracing uses next-event-estimate (shadow rays)

## Ray Tracing pseudo code

→ simplest implementation (Lambertian Shading)

→ image plane has dimension  $resX \cdot resY$

## function raytracer

MAX\_DEPTH = 3

for  $y : 0 \rightarrow resY$ :  
for  $x : 0 \rightarrow resX$ :  
} iterate over image plane

image[x,y] = 0

for  $s : 0 \rightarrow \#samplesperpixel$  → Samples per pixel

ray = generateCameraRay(x,y)

depth = 0

radiance = trace(ray, depth) / MAX\_DEPTH

image[x,y] += radiance

image[x,y] /= #samplesperpixel

## function trace(ray, depth)

radiance = 0

if depth != MAX\_DEPTH:

hitpoint = intersect(ray)

direction = SampleRandomCosine(hitpoint)

random\_ray = Ray(hitpoint, direction)

radiance += emit(hitpoint) + brdf(hitpoint) · trace(random\_ray, depth + 1)

return radiance