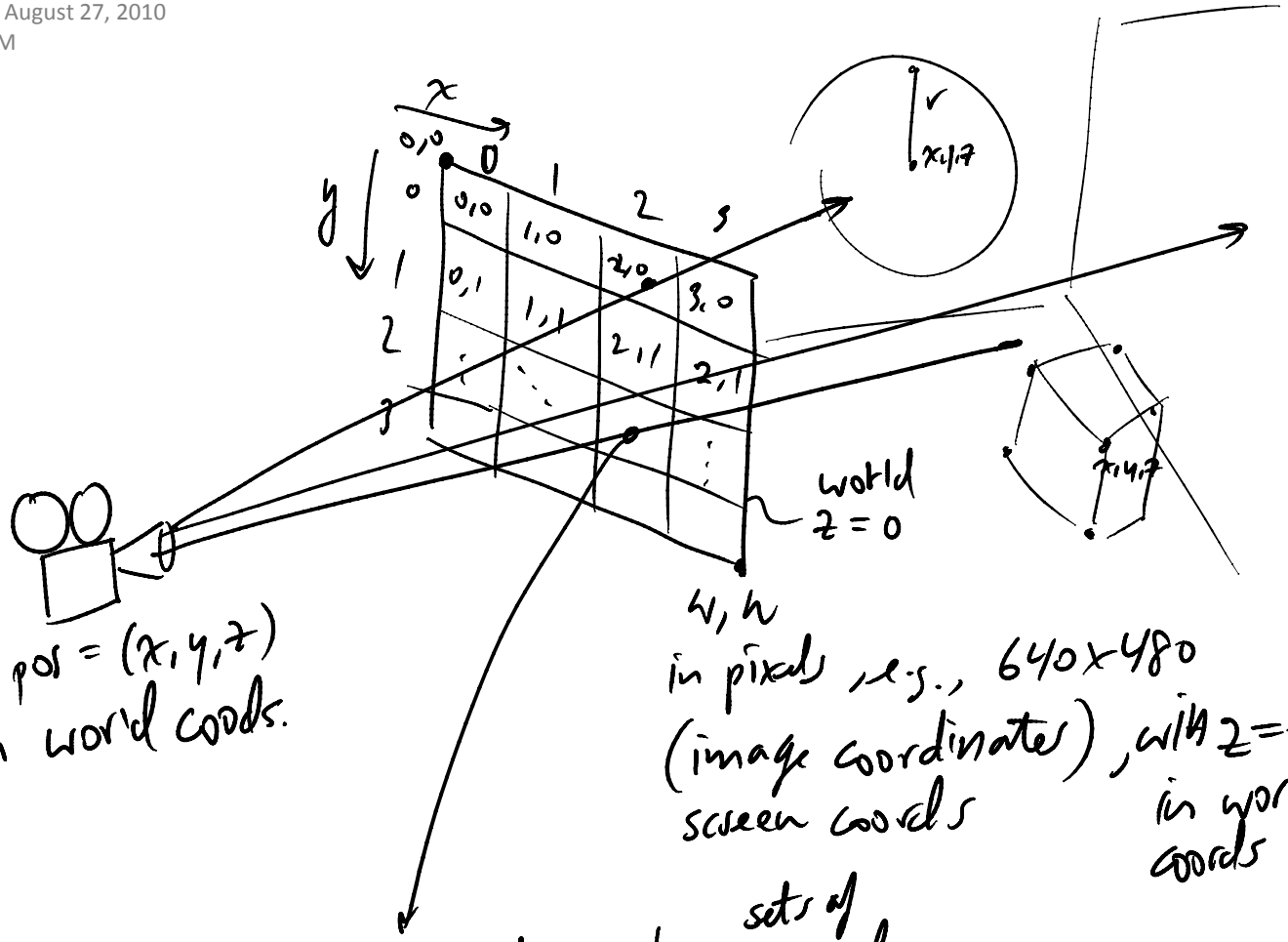
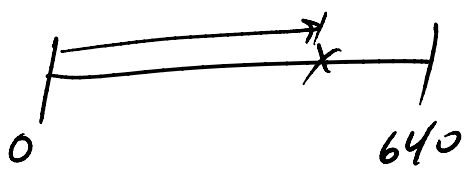
# Ray Tracing Intro

Friday, August 27, 2010
9:01 AM

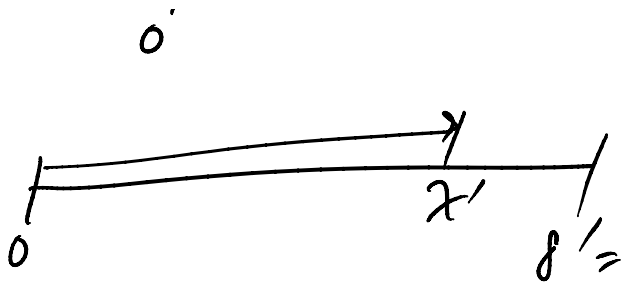



this pixel has two sets of coords, one in each reference frame (screen, world)

- how to go between the two reference frames?
- support image in coords, world, is 8 feet x 6 feet
- use simple linear mapping (linear interpolation)




$\frac{x}{w}$ normalizes x to [0,1] range




range

$\frac{x}{w} * w' = x'$

e.g. pixel at (100, 40)

$\frac{100}{640} * 8 = 1.25$

$\frac{40}{480} * 6 = 0.5$

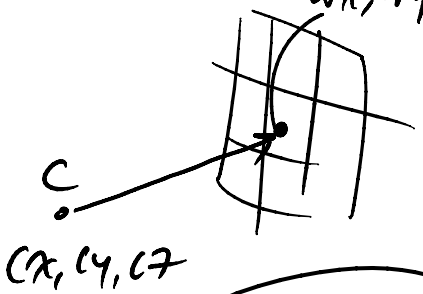
} floating point coords — can't be used for image indexing

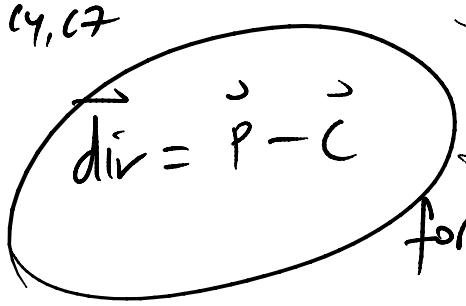
⇒ have to bank & forth between ref. frames

Friday, August 27, 2010
9:25 AM

Goal: Compute ray direction from camera to each pixel

```
int x,y;  int w = 640, h = 480;
for (y = 0; y<h; y++) {
   for (x = 0; x< w; x++) { // screen coords
      float ww = 8.0, wh = 6.0;
      float wx = (float)x / (float)w * ww;
      float wy = (float)y / (float)h * wh;
      float wz = 0.0;
      vec_t pixel(wx,wy,wz);
```



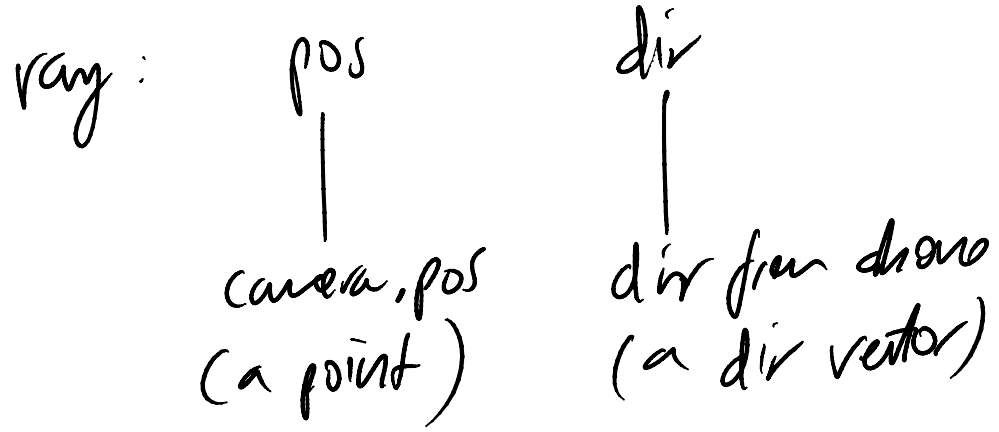




~~for( int i=0; i<3; i++) { div[i] = pixel[i] - camera.pos[i]; }~~

```
      vec_t div = vec_sub(pixel,
                          camera.pos);

      vec_unit(div, div); // make div
                          // more to come here
   } // end for
} // end for
```

We now have a ray:




where does this camera come from?

Friday, August 27, 2010
9:38 AM

(a model of the world)

- camera comes from scene description, stored in a text file

- Ray Tracing Alg:

step1. Load (read) the world model
  - a list of objects (we don't know how many) (need a linked list)
    (planes, spheres, etc.)
  - a list of materials (surface properties)
    (green, yellow, ...)
    - diffuse
    - ambient
    - specular
  - a camera
    {
      image size (w, h)
      world size (ww, wh)
      position (x, y, z)
    }

step 2. for each pixel in the image,

{

init color of pixel to (0.0, 0.0, 0.0) (R G B)

spawn ray thru pixel

find closest object to ray (hit point)

make a copy of the ambient color at the hit point

* scale color by $\frac{1.0}{\text{distance to hitpoint}}$

add color to pixel

}

step 3. - convert colors (floats) (R G B) ([0,1], [0,1], [0,1])

to unsigned characters [0, 255]

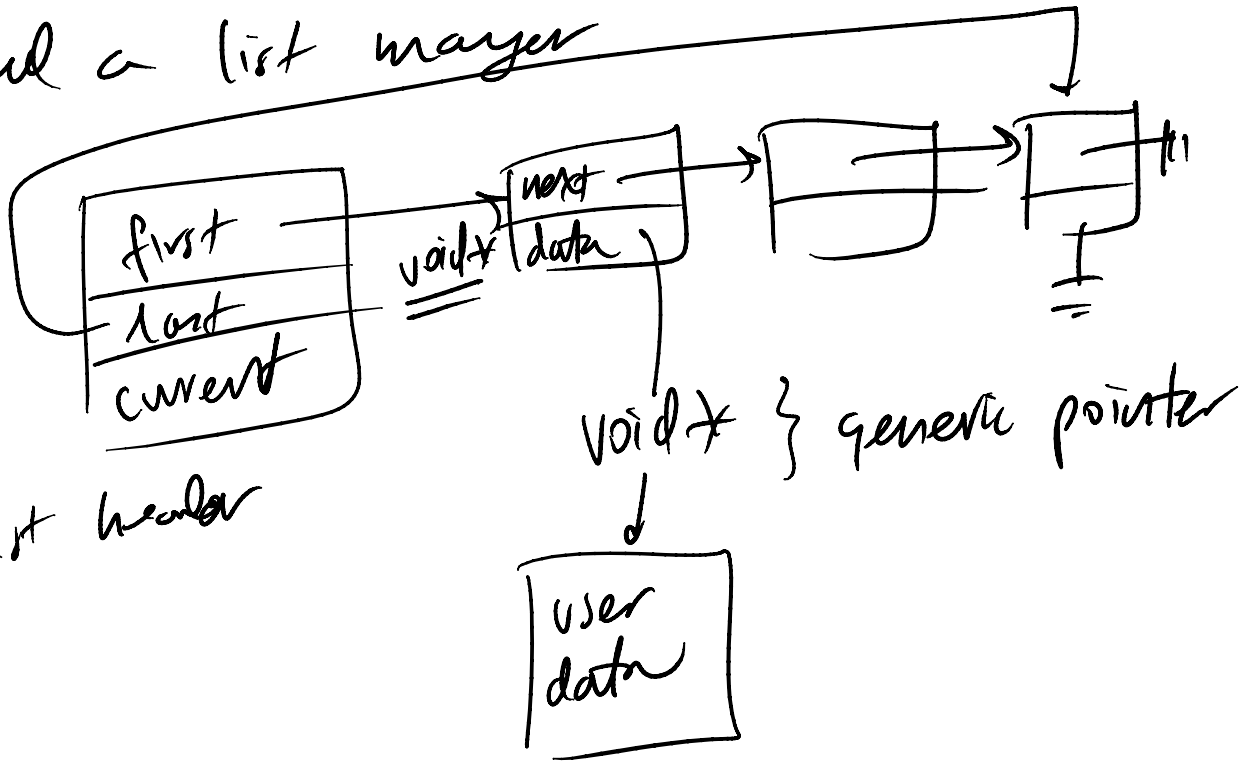
- output ppm image

Friday, August 27, 2010
9:49 AM

example input file:

```
camera cam1
{
    pixel dim   640   480
    world dim     8     6
    viewpoint     4     2     6
}
:
```

- need a list manager




list header

void* } generic pointer

from list's point of view, it holds generic items

generic items