EECS 332 Digital Image Analysis

Image Formation

Ying Wu

Dept. Electrical Engineering and Computer Science.
Northwestern University
Evanston, IL 60208

http://www.ece.northwestern.edu/~yingwu
yingwu@ece.northwestern.edu

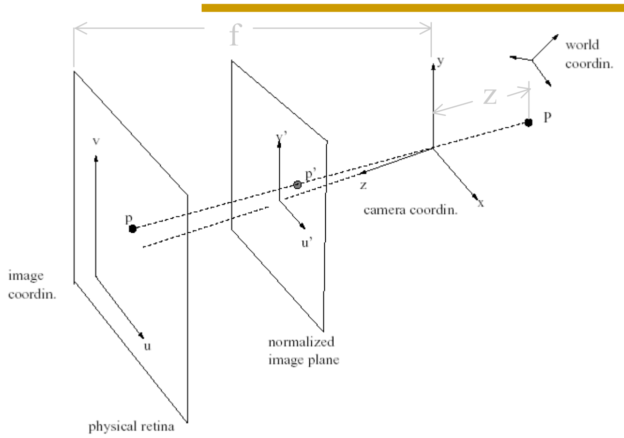
Questions for today

- How is an image formed by a camera?
- Why digital images?
- How can we draw a line in an image?

Outline

- Image Geometry
- Coordinate Transformation
- Digital Image
- Digital Thinking

Image Geometry



f --- focal length

z --- depth

[x,y,z] --- 3D point

[u',v'] --- normalized image coordinates

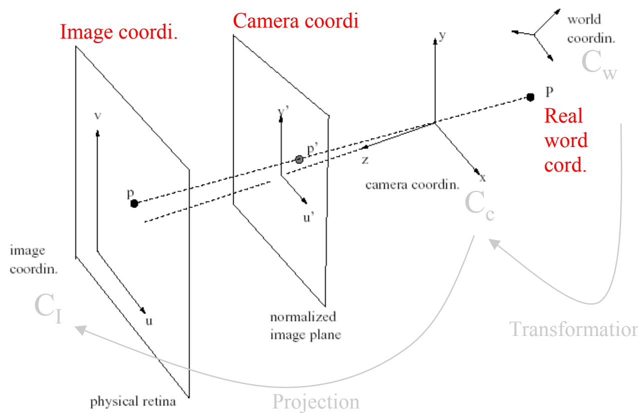
[u,v] --- image coordinates

$$u = \frac{f}{z} x$$

$$v = \frac{f}{z} y$$

Assuming the camera coordinate frame is the same as the world frame

Coordinate Transformation



$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = {}^c_w R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + {}^c_w t$$

Craig Notation

- ^{F}P means the coordinate of P in frame F
- Translation

$$^B\mathrm{P} = {}^A\mathrm{P} + {}^B\mathrm{O}_A$$

- Rotation

$$^B\mathrm{P} = {}^B_A\mathrm{R}\,{}^A\mathrm{P}$$

- Rigid transformation

$$^B\mathrm{P} = {}^B_A\mathrm{R}\,{}^A\mathrm{P} + {}^B\mathrm{O}_A$$

Digital Images

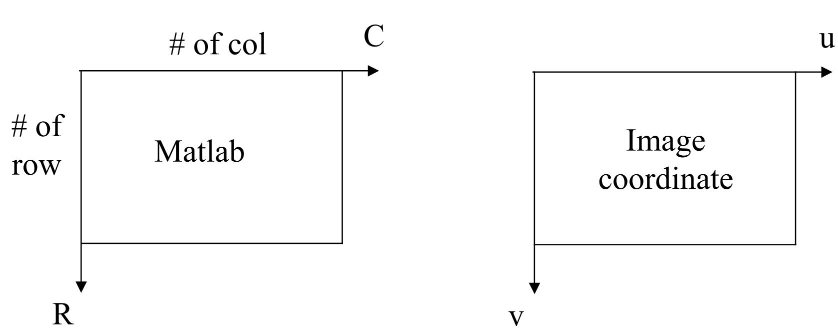
- Sampling & Quantization

  Sampling → resolution; Quantization → intensity level
- Digital image: an array of integers
  - Pixel: "gray-value + location" I (u, v) = E[0, 255]+2^8 ==> (dark, light)
    - ✓ a sample of image intensity quantized to an integer value
  - Color pixel
    - ✓ RGB vs. gray scale
  - Using matlab

```
>> img = imread('test.bmp','bmp');
>> [R,C] = size(img);
```

For color image, you can have `img(:,:,:)`

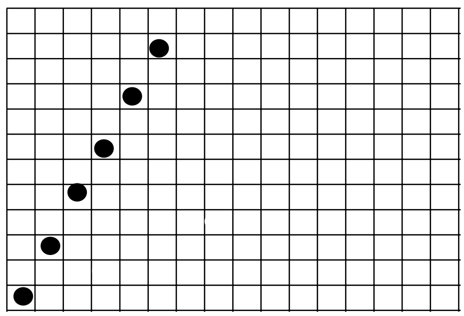
Conventions



`img(r,c)` ≠ `img[u,v]`

Digital Thinking

- Q: how can we draw a line in a digital image?
  (or how to find those pixels on that line?) dot -> dots> to line
- Example: y=2x vs. y = 0.5x



	y=2x	y=0.5x
x=0	y=0	y=0
x=1	y=2	y=[0.5]=0
x=2	y=4	y=1
x=3	y=6	y=[1.5]=1

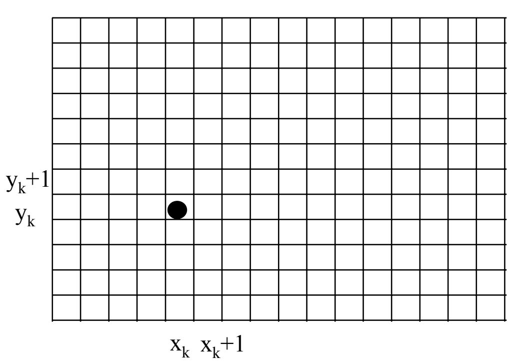
x++ algo → not good for slop>1

y++ algo → not good for slop <1

So, can we combine them?

Idea

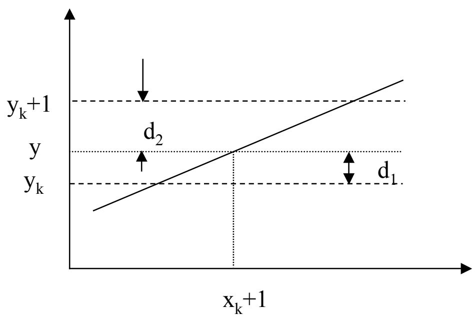
- Consider the case: slop <1, x++
  - i.e the Line: y=mx+b, where m<1



For a given (x_k,y_k), we have to make a choice between (x_k+1,y_k), or (x_k+1,y_k+1)

Then, how can we make the decision?

Decision making



True value: $y=m(x_k+1)+b$

$d_1 = y-y_k = m(x_k+1)+b-y_k$

$d_2 = y_k+1-y = y_k+1-m(x_k+1)-b$

Then

$d_1-d_2 = 2m(x_k+1)-2y_k+2b-1$

d1-d2 <0	select (x_k+1, y_k)
d1-d2>0	select (x_k+1, y_k+1)

Is it good enough?

Motivation

- Two observations
  - It is expensive to use floating point operations
  - It is time-consuming to calculate d1-d2 from scratch every time
- Solutions?
  - Floating point → integer
  - Everything from scratch → recursive algorithm

Bresenham's algo

- Given the two end points (x_1,y_1), (x_2,y_2)
- Trick 1: Introducing a decision parameter p_x
  - Define $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$
  - It follows: $m = \Delta y / \Delta x$
  - Define: $p_k = \Delta x(d_1 - d_2)$

    $= 2\,\Delta y x_k - 2\,\Delta x y_k + [2\,\Delta y + \Delta x(2b-1)]$
  - p_k doesn't change the sign of $(d_1 - d_2)$

Bresenham's algo

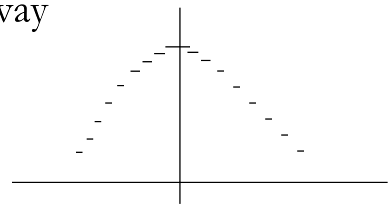
- Trick 2:
  - Don't calculate p_k every time
  - Since $p_{k+1} - p_k = 2\,\Delta y(x_{k+1} - x_k) - 2\,\Delta x(y_{k+1} - y_k)$

    $= 2\,\Delta y - 2\,\Delta x(y_{k+1} - y_k)$ (WHY?)
  - Thus
    - ✓ if $p_k < 0$, $p_{k+1} = p_k + 2\,\Delta y$
    - ✓ if $p_k > 0$, $p_{k+1} = p_k + 2\,\Delta y - 2\,\Delta x$
- Please prove

  $p_0 = 2\,\Delta y - \Delta x$

Generalization

- Draw an ellipse $r_x x^2+r_y y^2=r_x^2 r_y^2$
- A bad way



- Any idea?