

קורס גרפיקה ממוחשבת

2009/2010 סמסטר א'

Image Processing II

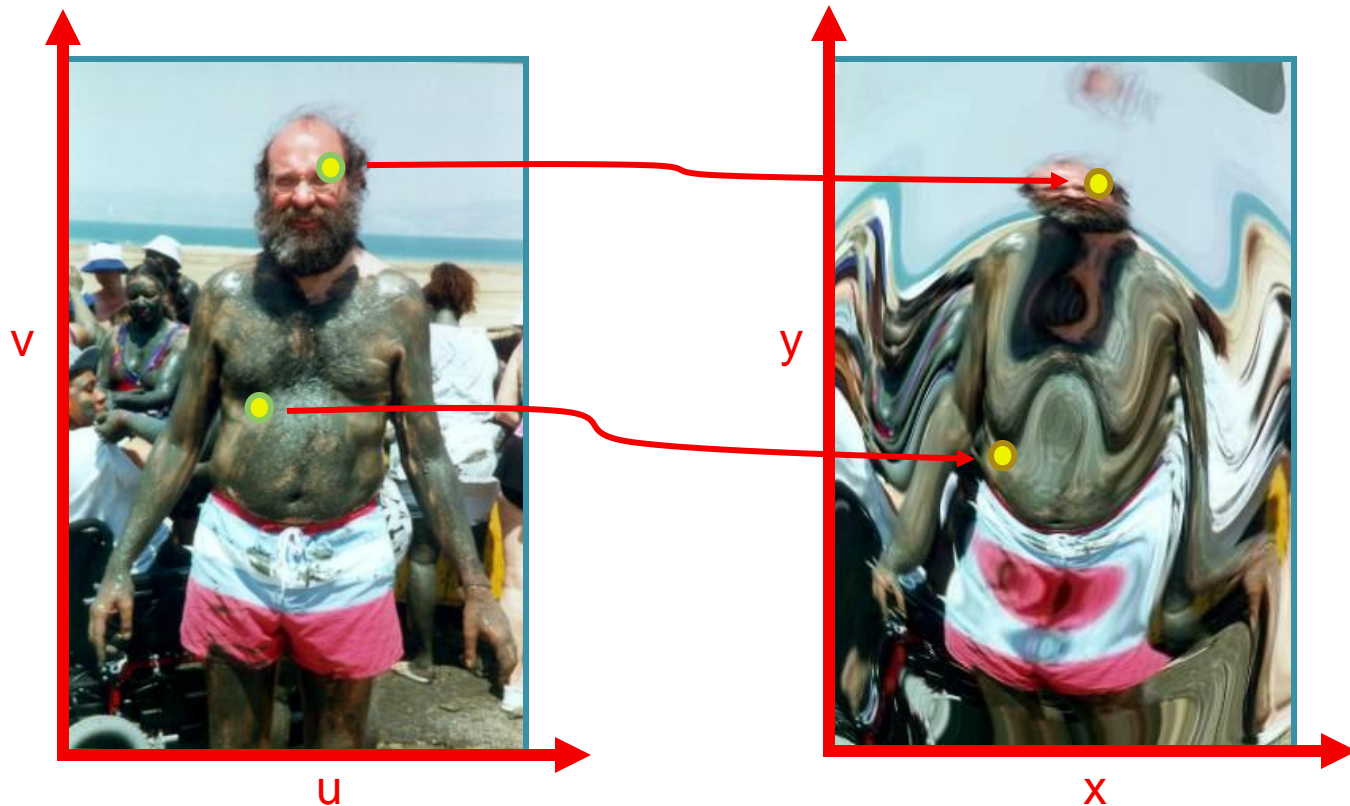


Agenda

- **Warping to Mapping**
- **Quantization**

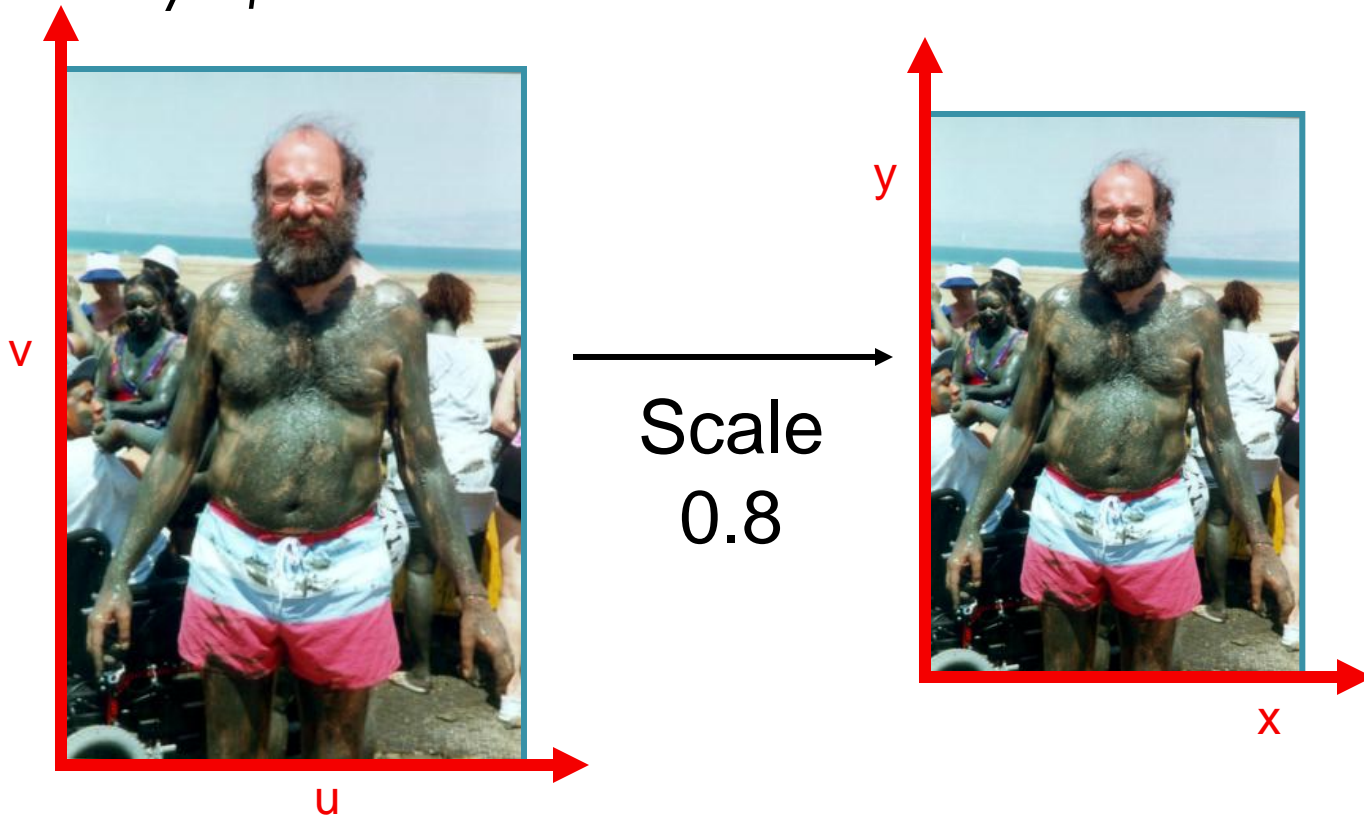
Mapping

- Last time we started to discuss warping and mapping
- In general, we define a transformation
 - Destination (x,y) for every source (u,v)



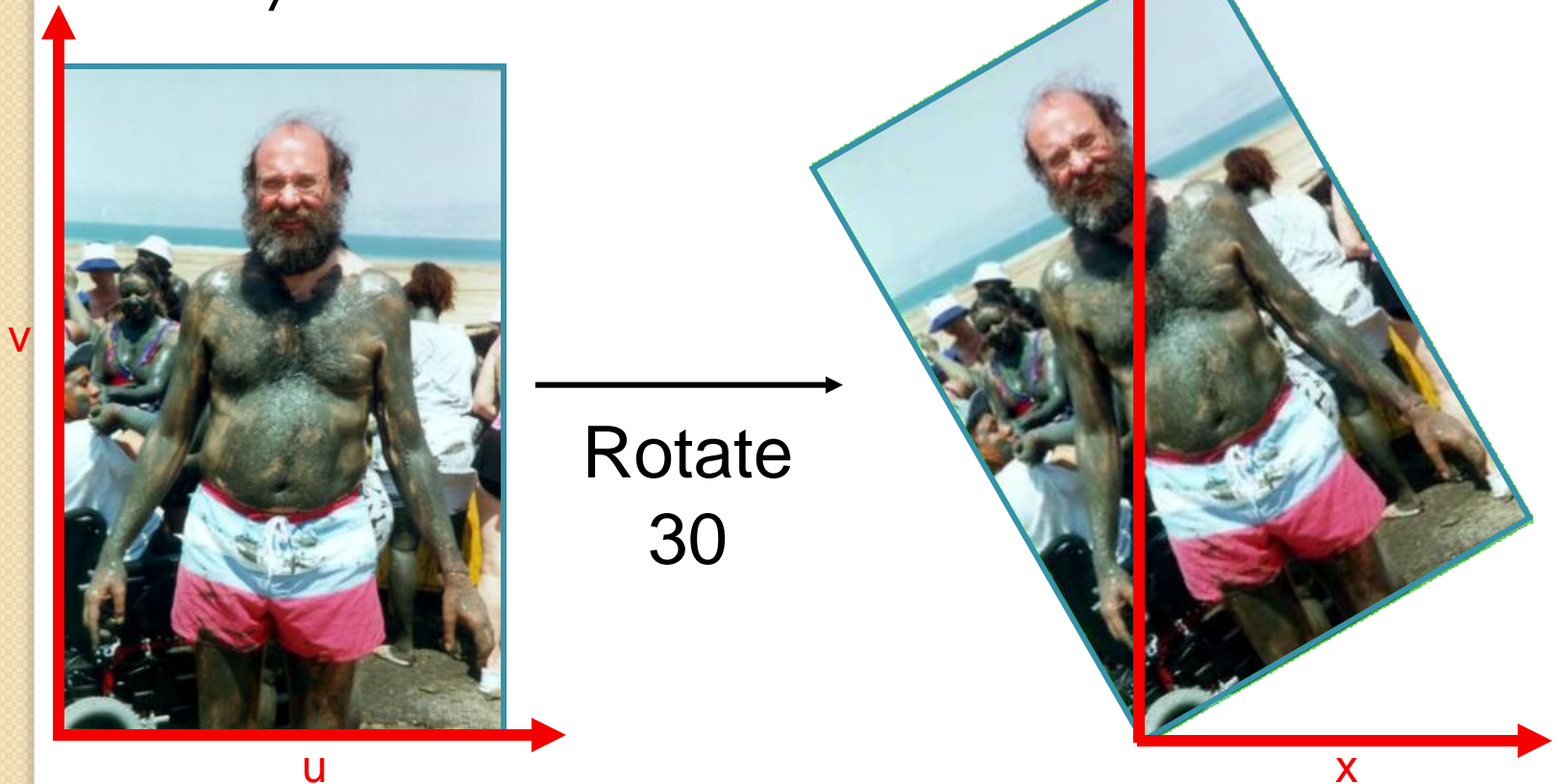
Example Mappings

- Scale by *factor*:
 - $x = \text{factor} * u$
 - $y = \text{factor} * v$



Example Mappings

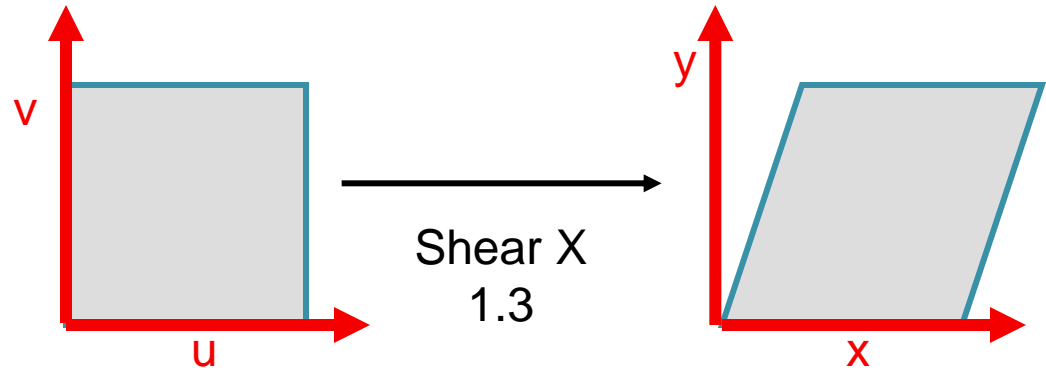
- Rotate by Θ degrees:
 - $x = u \cos \Theta - v \sin \Theta$
 - $y = u \sin \Theta + v \cos \Theta$



Example Mappings

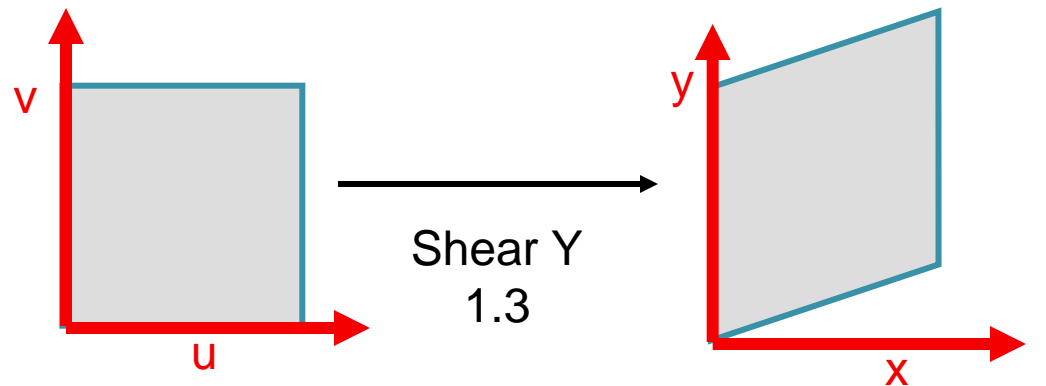
- Shear in X by factor:

- $x = u + \text{factor} * v$
- $y = v$



- Shear in Y by factor:

- $x = u$
- $y = v + \text{factor} * u$



Other Mappings

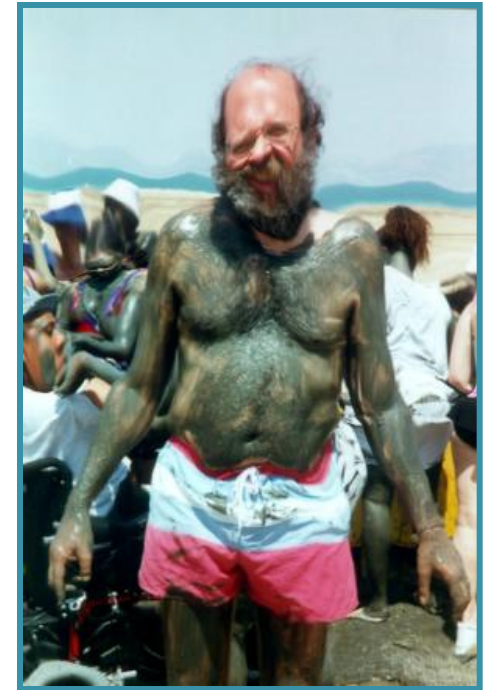
- Any function of u and v :
 - $x = f_x(u,v)$
 - $y = f_y(u,v)$



Fish-eye



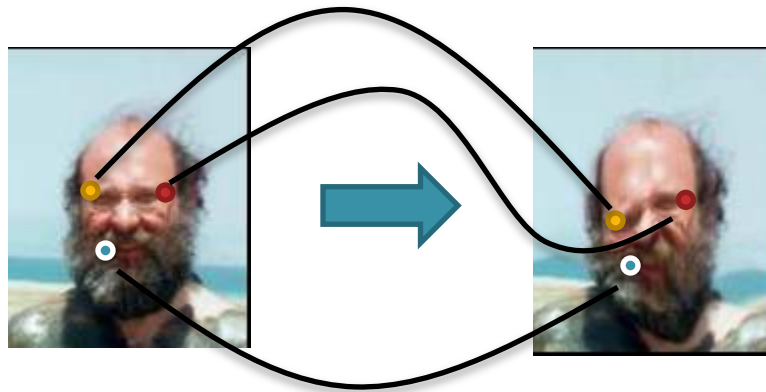
“Swirl”



“Rain”

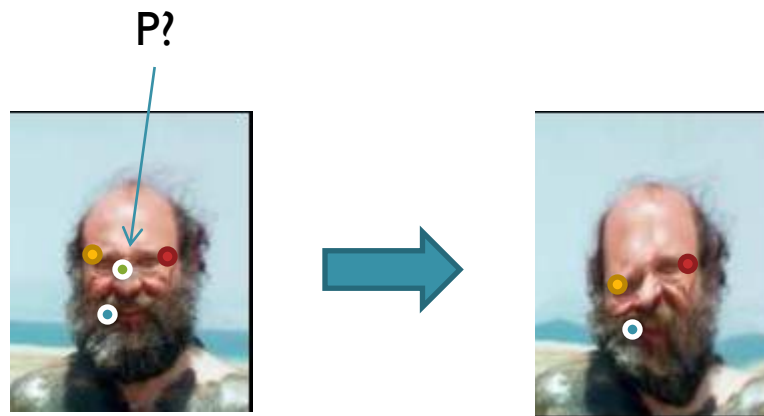
Point Correspondence

- Another way to define mapping is by correspondences
 - $A \leftrightarrow A'$
 - $B \leftrightarrow B'$
 - $C \leftrightarrow C'$



Point Correspondence

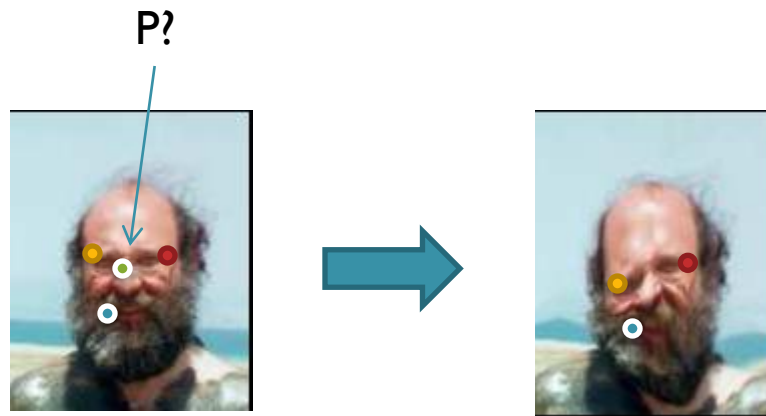
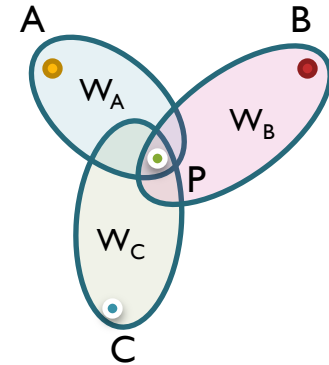
- Another way to define mapping is by correspondences
 - $A \leftrightarrow A'$
 - $B \leftrightarrow B'$
 - $C \leftrightarrow C'$



Point Correspondence

- How to compute P'

$$P' = w_A A + w_B B + w_C C$$



Point Correspondence

- Barycentric Coordinates

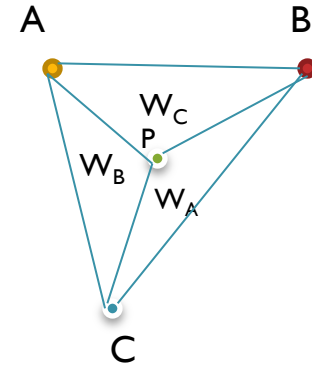
$$p = \lambda_1 A + \lambda_2 B + \lambda_3 C$$

$$\begin{cases} p_x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 \\ p_y = \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3 \\ \lambda_1 + \lambda_2 + \lambda_3 = 1 \end{cases}$$



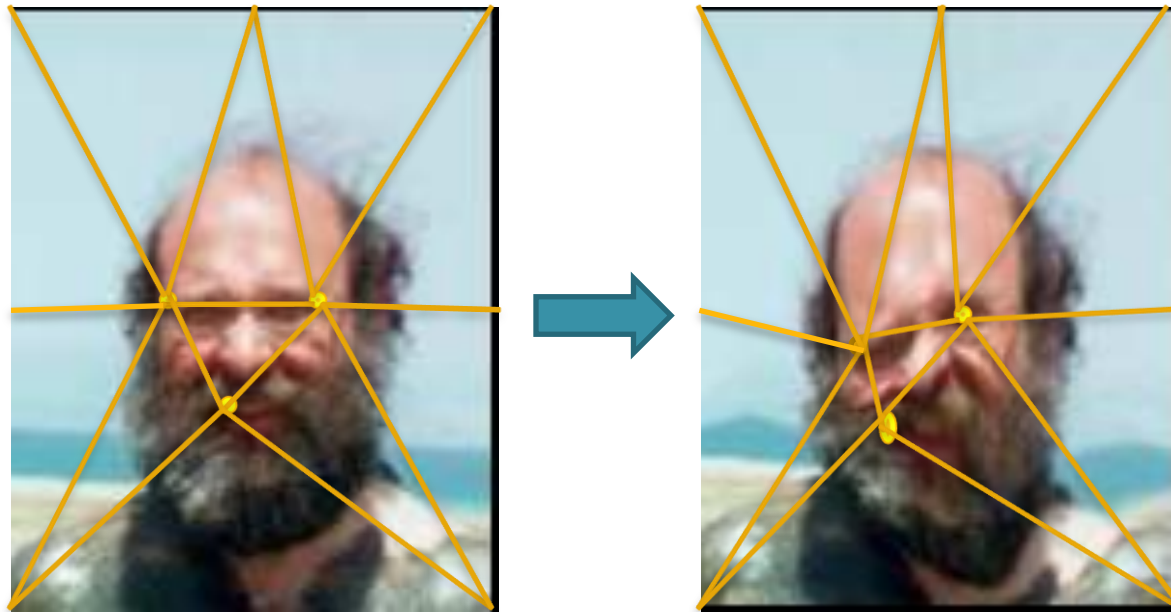
$$\lambda_1(x_1 - x_3) + \lambda_2(x_2 - x_3) + x_3 - p_x = 0$$

$$\lambda_1(y_1 - y_3) + \lambda_2(y_2 - y_3) + y_3 - p_y = 0$$



Point Correspondence

1. Triangulate image
2. Transform mesh
3. Map each pixel using Barycentric coordinates



Possible application: Morphing

- User specifies corresponding points
- Blend while warping



Image Processing

- Pixel operations
 - Add random noise
 - Add luminance
 - Add contrast
 - Add saturation
- Filtering
 - Blur
 - Detect edges
 - Sharpen
 - Emboss
 - Median
- Quantization
 - Uniform Quantization
 - Floyd-Steinberg dither
- Warping
 - Scale
 - Rotate
 - Warps
- Combining
 - Composite
 - Morph

Quantization

- Reduce intensity resolution
 - Frame buffers have limited number of bits per pixel
 - Physical devices have limited dynamic range

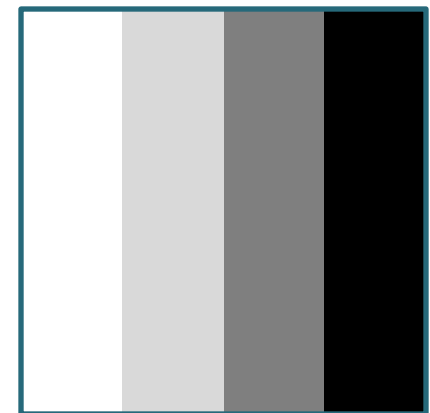
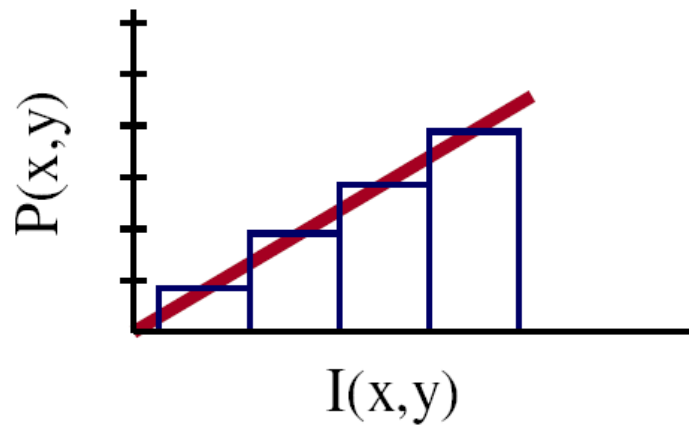


Uniform Quantization

- $P(x,y) = \text{round}(I(x,y))$



$I(x,y)$



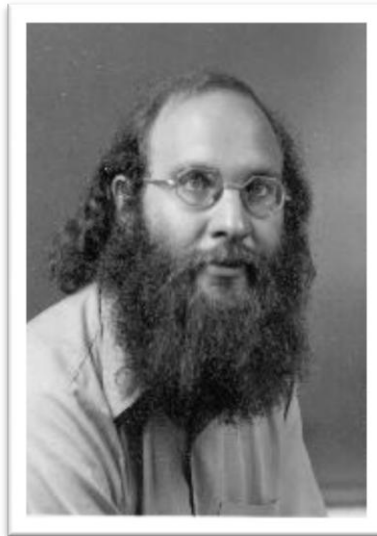
$P(x,y)$ – 2 bits per pixel

Uniform Quantization

- Images with decreasing bits per pixel:



8 bits



4 bits



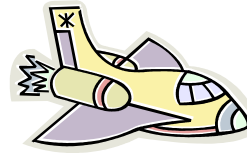
2 bits



1 bit

Reducing effects of Quantization

- Dithering
 - Random dither
 - Ordered dither
 - Error diffusion dither
- Halftoning
 - Classical halftoning



Dithering

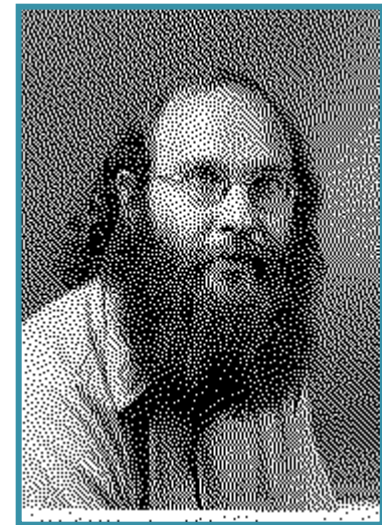
- Distribute errors among pixels
 - Exploit spatial integration in our eye
 - Display greater range of perceptible intensities



Original
(8 bits)



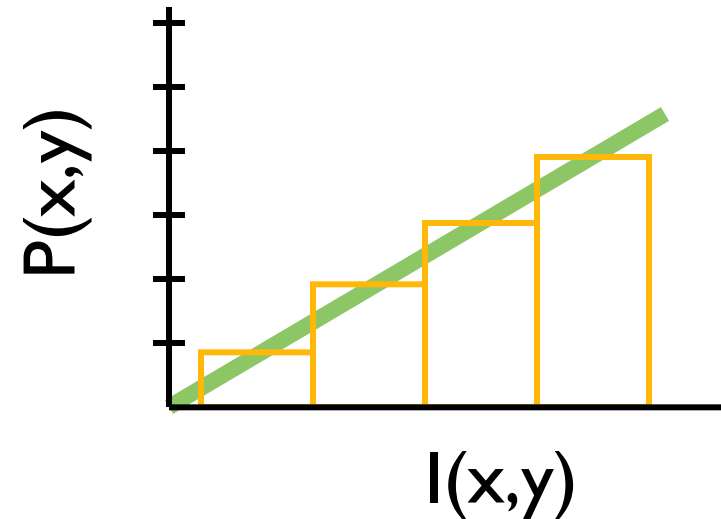
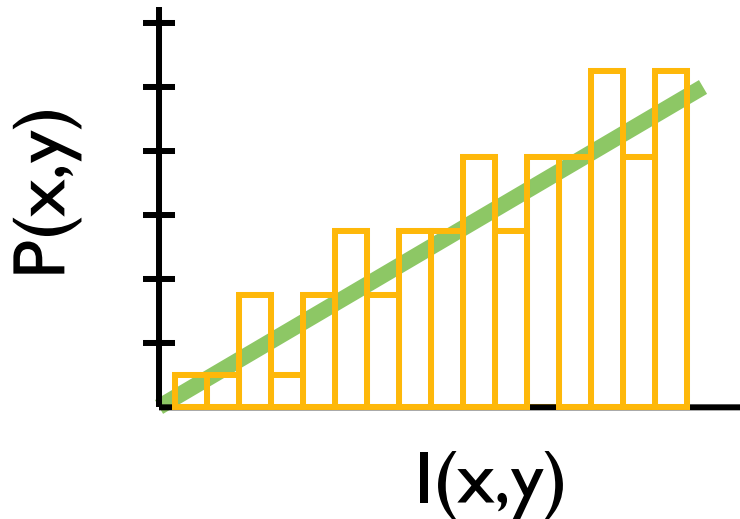
Uniform
Quantization
(1 bit)



Floyd-Steinberg
Dither
(1 bit)

Random Dither

- Randomize quantization errors
 - Errors appear as noise



$$P(x, y) = \text{trunc}(I(x, y) + \text{noise}(x, y) + 0.5)$$

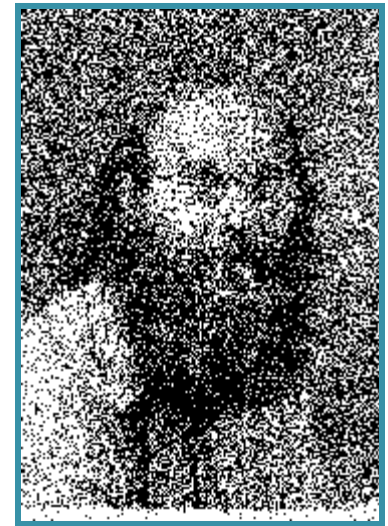
Random Dither



Original
(8 bits)



Uniform
Quantization
(1 bit)



Random
Dither
(1 bit)

Ordered Dither

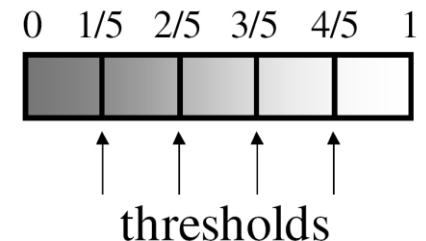
- Pseudo-random quantization errors
 - Matrix stores pattern of thresholds

$$D_2 = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$$

For each pixel (x,y)

oldpixel = I(x,y) + D(x mod n, y mod n)

P(x,y) = find_closest_color(oldpixel)



Ordered Dither

- Bayer's ordered dither matrices

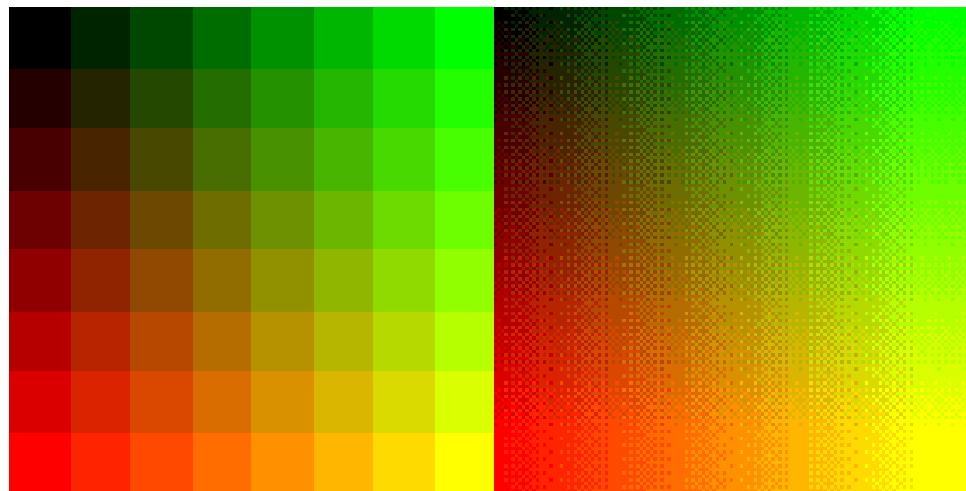
$$D_n = \begin{bmatrix} 4D_{n/2} + D_2(1,1)U_{n/2} & 4D_{n/2} + D_2(1,2)U_{n/2} \\ 4D_{n/2} + D_2(2,1)U_{n/2} & 4D_{n/2} + D_2(2,2)U_{n/2} \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \quad D_4 = \begin{bmatrix} 15 & 7 & 13 & 5 \\ 3 & 11 & 1 & 9 \\ 12 & 4 & 14 & 6 \\ 0 & 8 & 2 & 10 \end{bmatrix}$$

Basic idea: organize successive integers such that the average distance between two successive numbers in the map is as large as possible

Ordered Dither

- An example
 - Palette consists of 8 red tones, 8 green tones and their combinations (64 colors)
 - Original image had 19600 colors



Undithered

Dithered

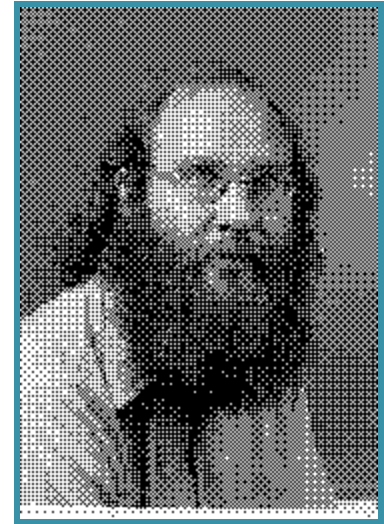
Ordered Dither



Original
(8 bits)



Random
Dither
(1 bit)



Ordered
Dither
(1 bit)

Floyd-Steinberg Algorithm

```
for (x = 0; x < width; x++) {  
    for (y = 0; y < height; y++) {  
        P(x,y) = trunc(l(x,y) + 0.5)  
        e = l(x,y) - P(x,y)  
        l(x,y+1) +=  $\alpha$ *e;  
        l(x+1,y-1) +=  $\beta$ *e;  
        l(x+1,y) +=  $\gamma$ *e;  
        l(x+1,y+1) +=  $\delta$ *e;  
    }  
}
```

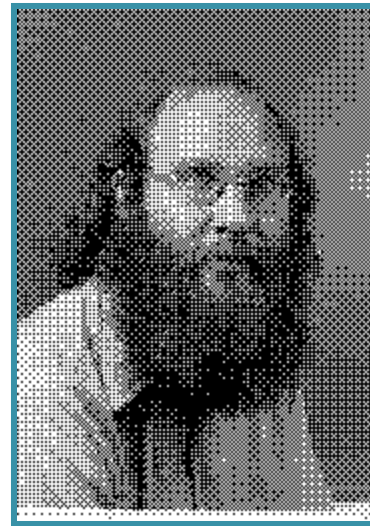
Error Diffusion Dither



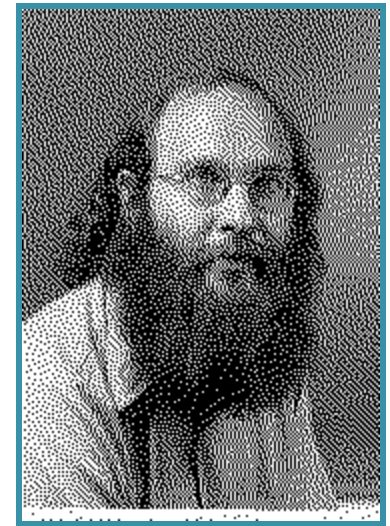
Original
(8 bits)



Random
Dither
(1 bit)



Ordered
Dither
(1 bit)



Floyd-Steinberg
Dither
(1 bit)

More examples

Original



Threshold



Random



Bayer



Floyd-Steinberg



Jarvice, Judice & Ninke



Stucki



Burkes

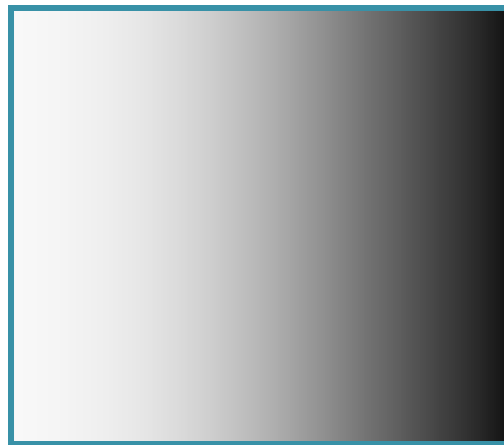


Reducing effects of Quantization

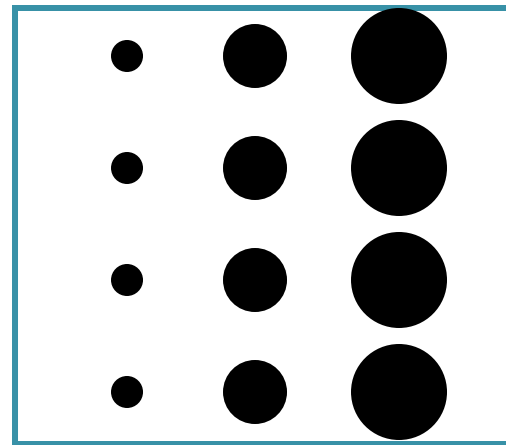
- Dithering
 - Random dither
 - Ordered dither
 - Error diffusion dither
- Halftoning
 - Classical halftoning

Classical Halftoning

- Use dots of varying size to represent intensities
 - Area of dots proportional to intensity in image

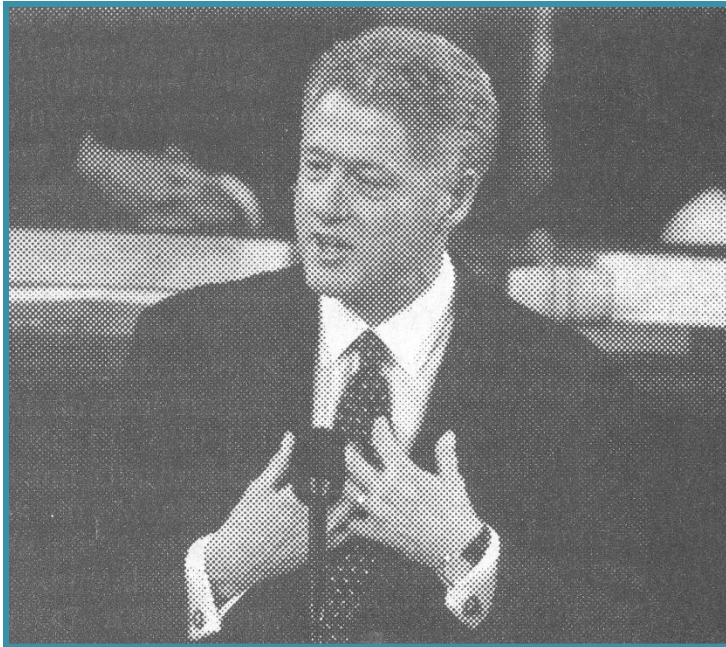


$I(x,y)$

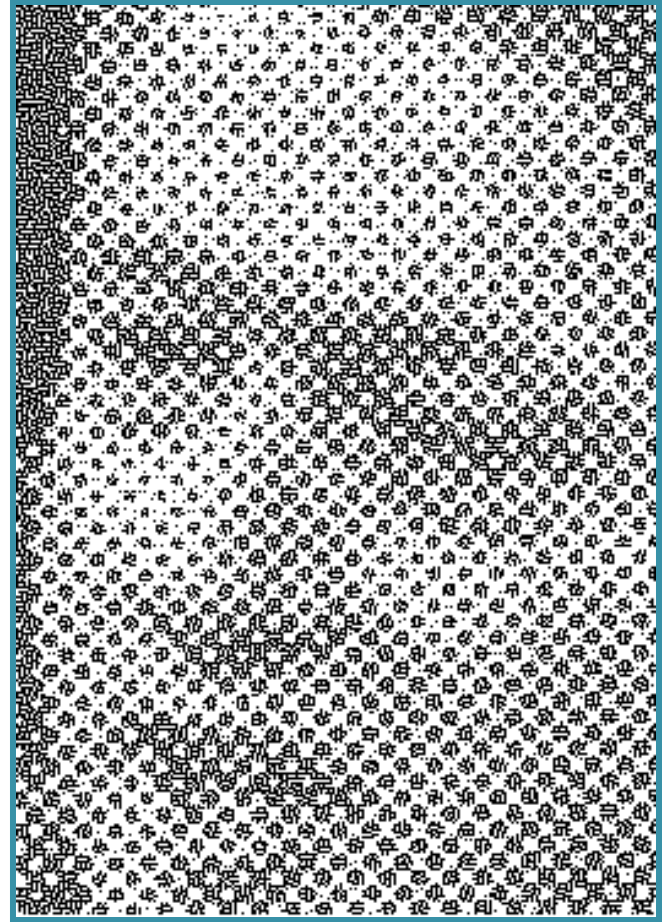


$P(x,y)$

Classical Halftoning



Newspaper Image



From New York Times, 9/21/99

Halftone patterns

- Use cluster of pixels to represent intensity
 - Trade spatial resolution for intensity resolution

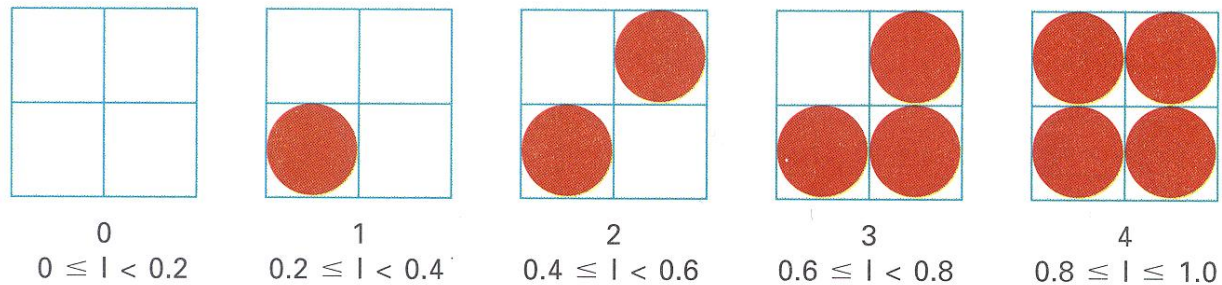


Figure 14.37 from H&B

Halftone patterns

- How many intensities in a $n \times n$ cluster?

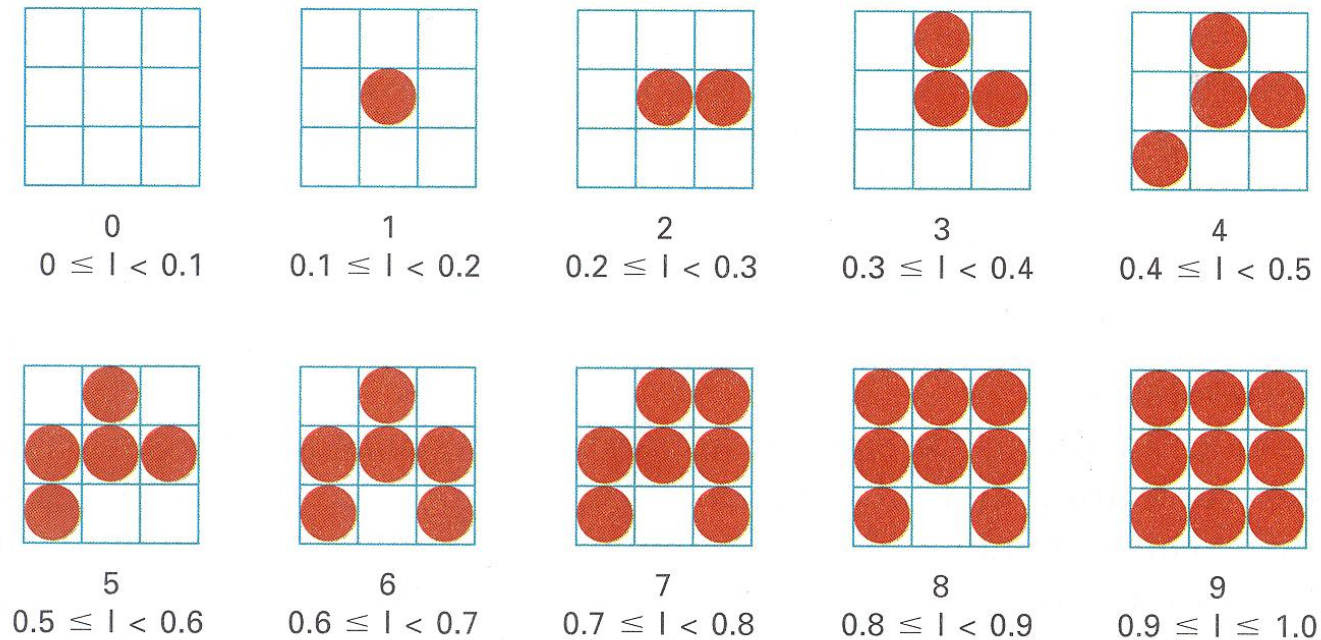


Figure 14.37 from H&B