CSCI 4830 / 5722
Computer Vision

University of Colorado **Boulder**

CSCI 4830 / 5722
Computer Vision

Dr. Ioana Fleming
Spring 2019
Lecture 9

University of Colorado **Boulder**

Reminders

Submissions:

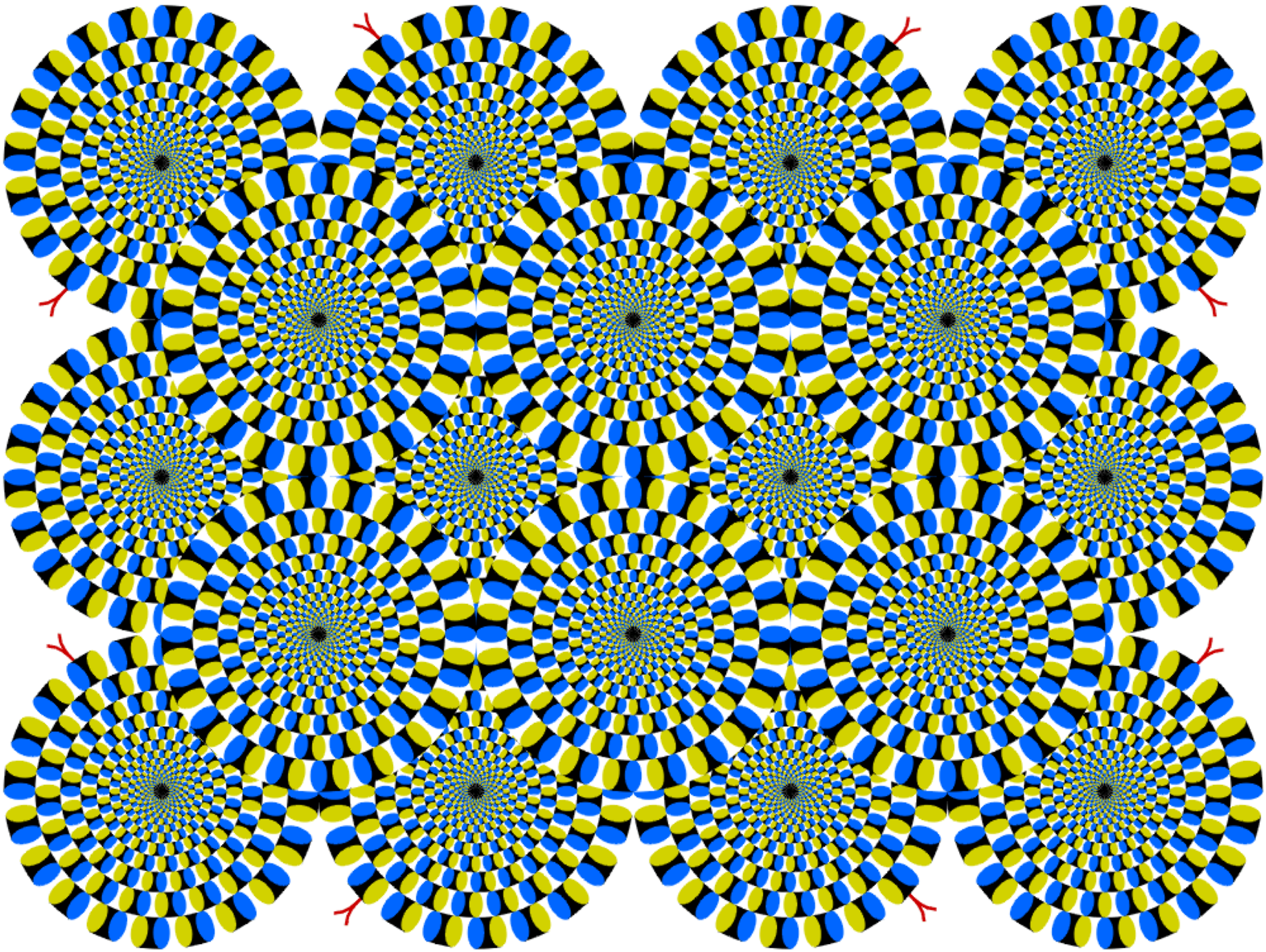
- Homework 2: due Wed 2/13 at 11 pm
- Homework 3: later this week

Readings:

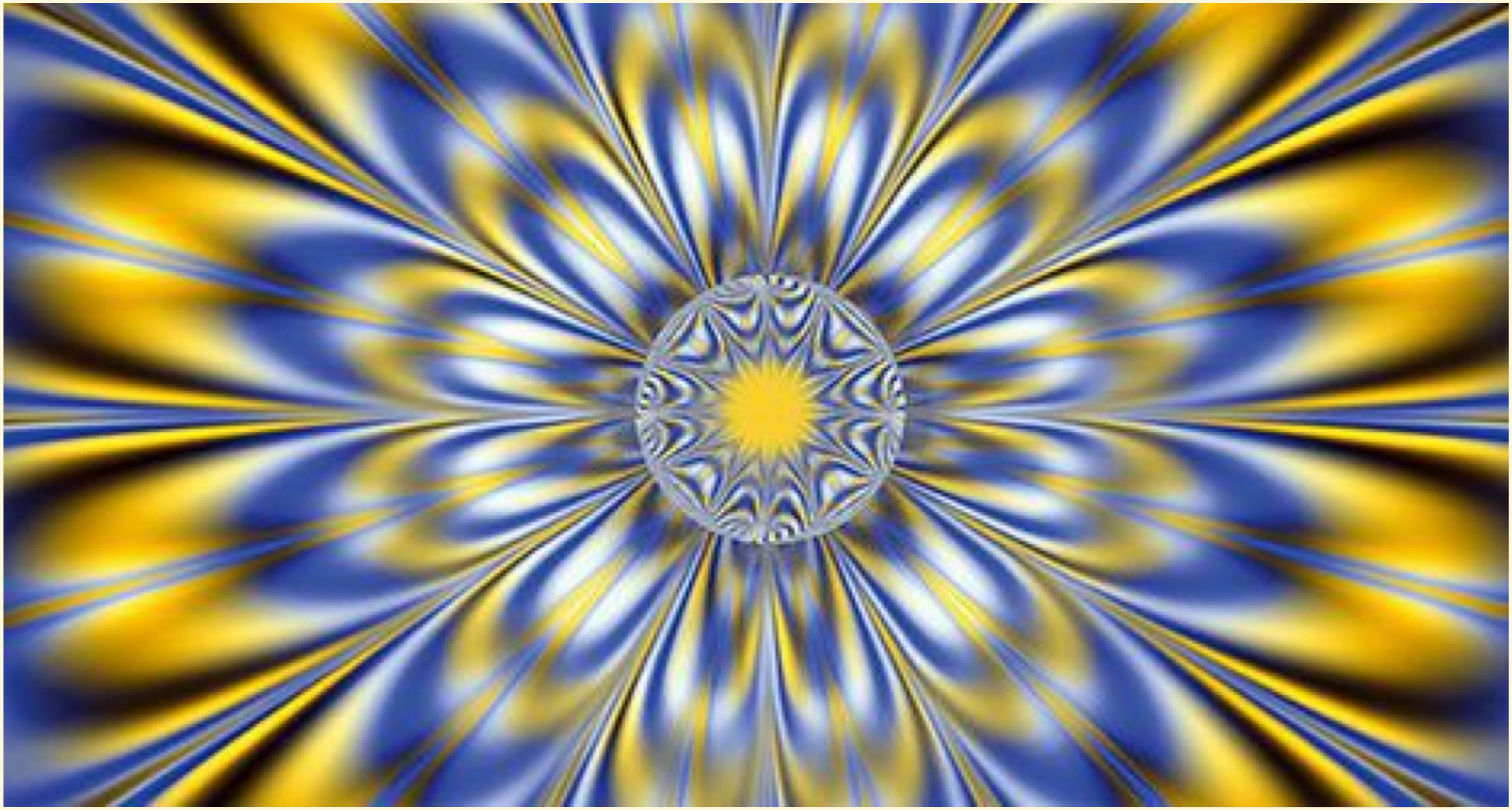
- Szeliski:
    - chapter 3 (filters, changing resolution, Laplacian pyramids, warping)
    - chapter 4.1 (points) and 4.2 (edge detection)
- P&F Ch. 4,5

Today

- Intensity Surfaces
- Gradients
- Image filters as linear operators



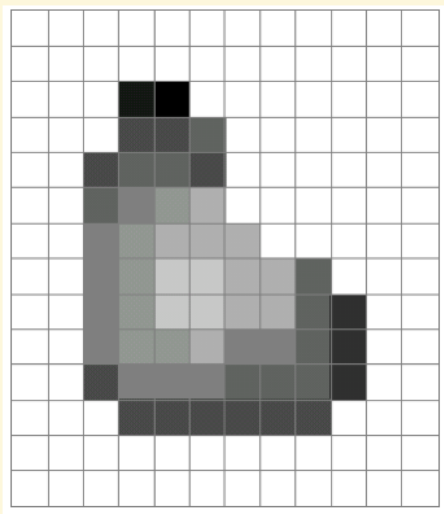
Moiré effect



University of Colorado **Boulder**

What is an image?

- A grid (matrix) of intensity values



=

255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	175	255	255	255	255	255
255	255	127	145	200	200	175	175	95	255	255	255
255	255	127	145	200	200	175	175	95	47	255	255
255	255	127	145	145	175	127	127	95	47	255	255
255	255	74	127	127	127	95	95	95	47	255	255
255	255	255	74	74	74	74	74	74	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

(common to use one byte per value: 0 = black, 255 = white)

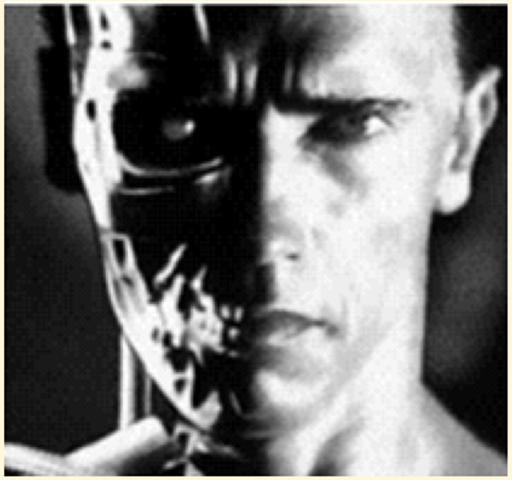
Visualizing Images

Our goal: we want to visualize images at a level high enough to retain human insight, but low enough to allow us to readily translate our insights into mathematical notation and, ultimately, computer algorithms that operate on arrays of numbers.

What is an image?

- We can think of a (grayscale) image as a **function**, *f*, from R^2 to R:
  - $f\,(x,y)$ gives the **intensity** at position (x,y)

A **digital** image is a discrete (**sampled**, **quantized**) version of this function



What is an image?

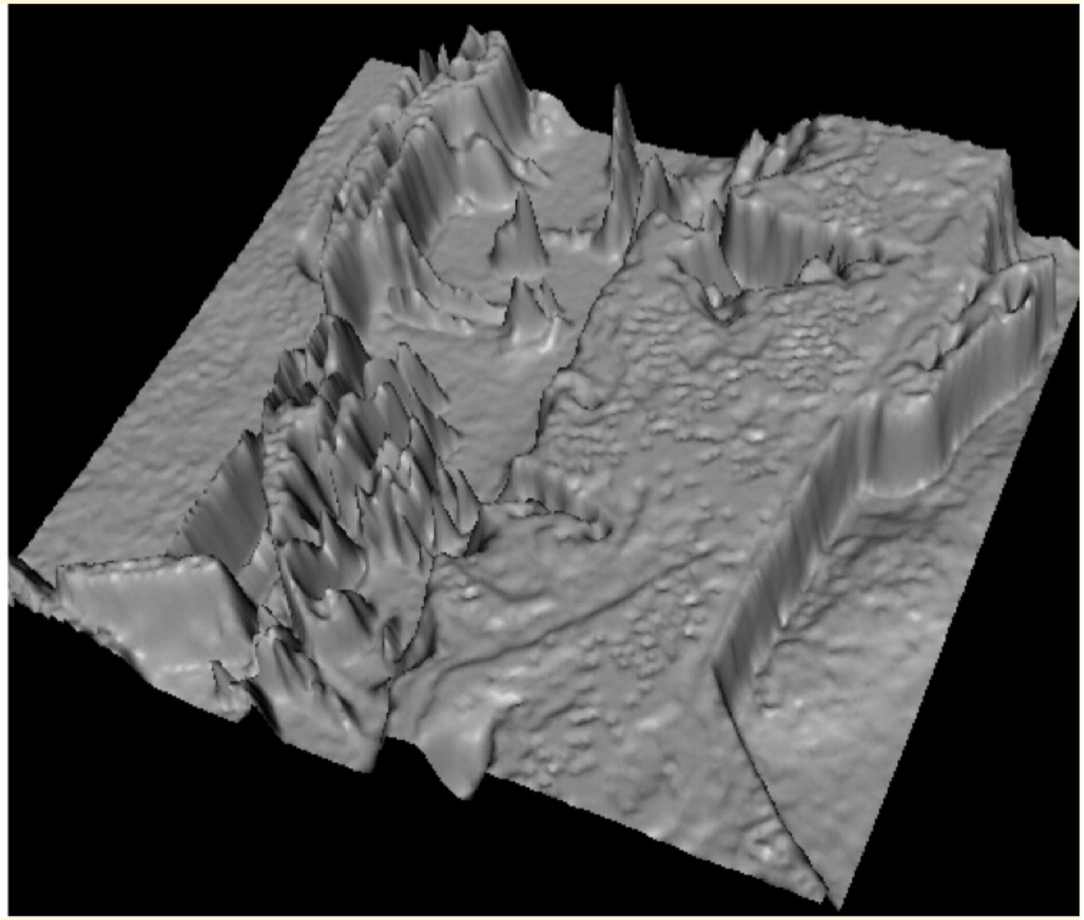
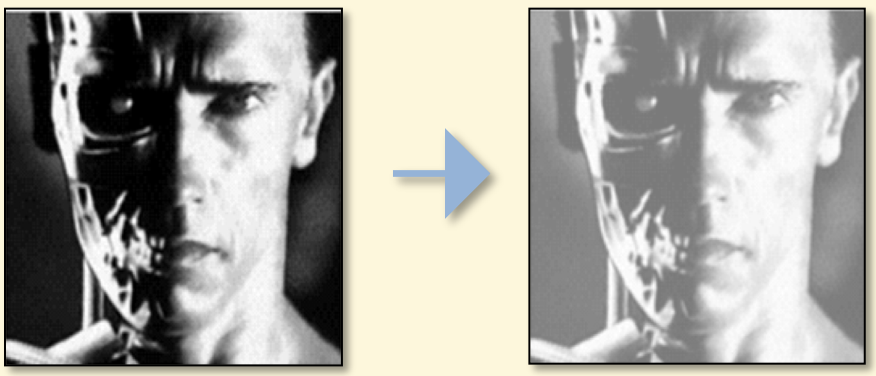
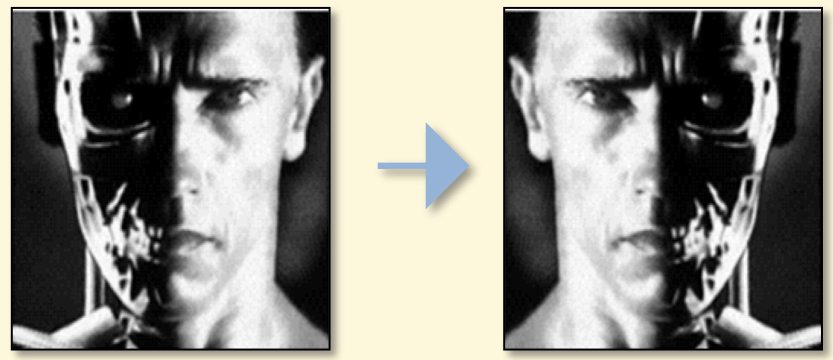


Image transformations

- As with any function, we can apply operators to an image

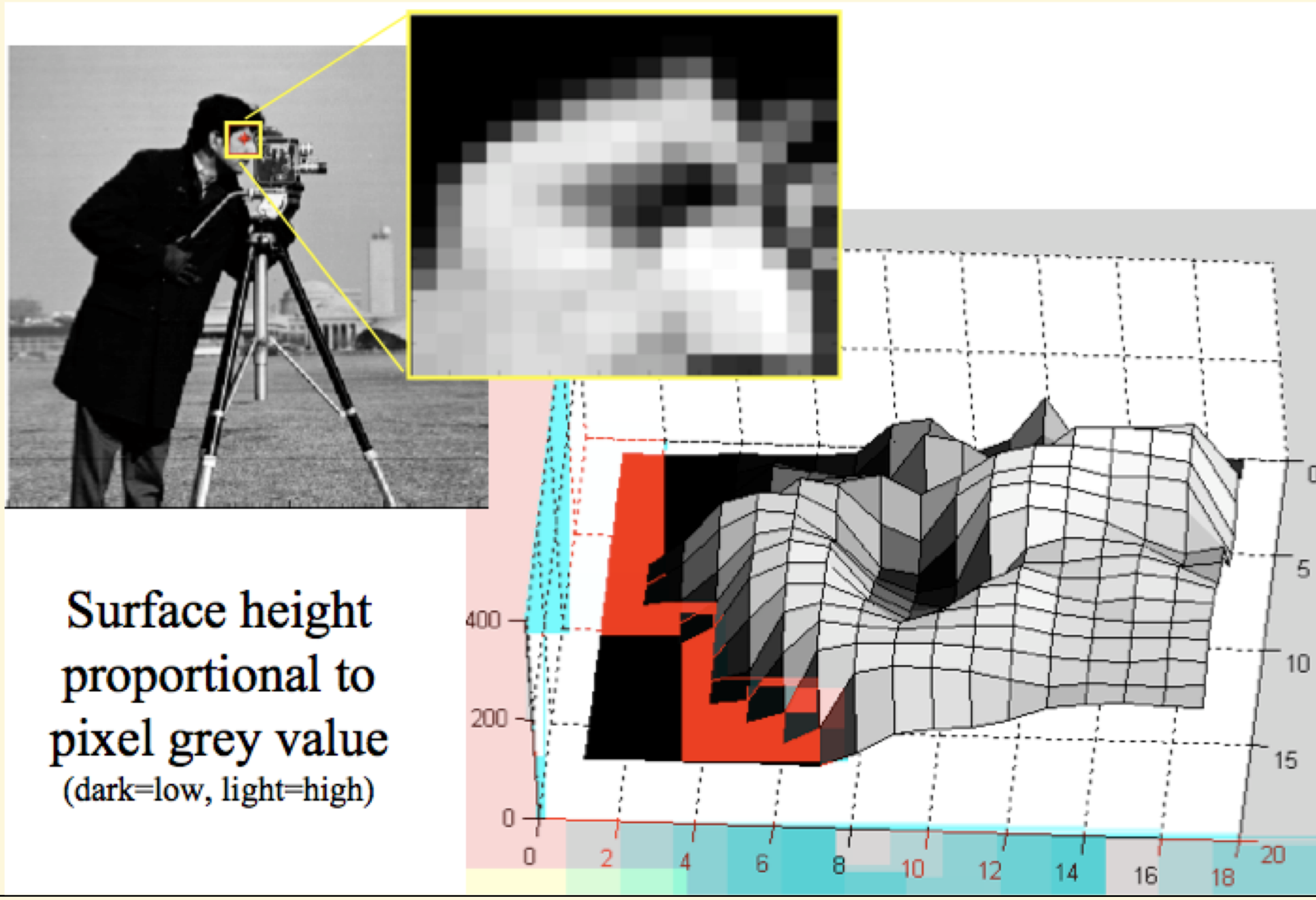


$$g(x,y) = f(x,y) + 20$$



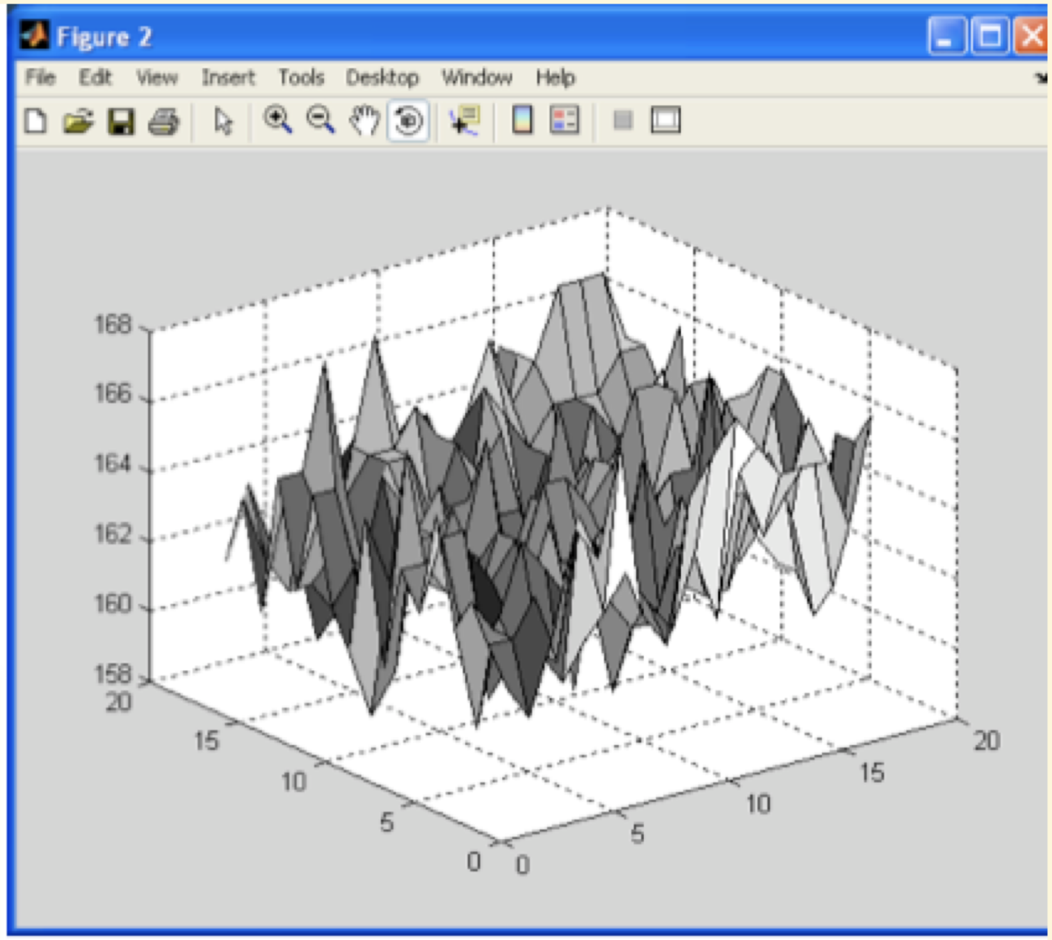
$$g(x,y) = f(-x,y)$$

Images as surfaces



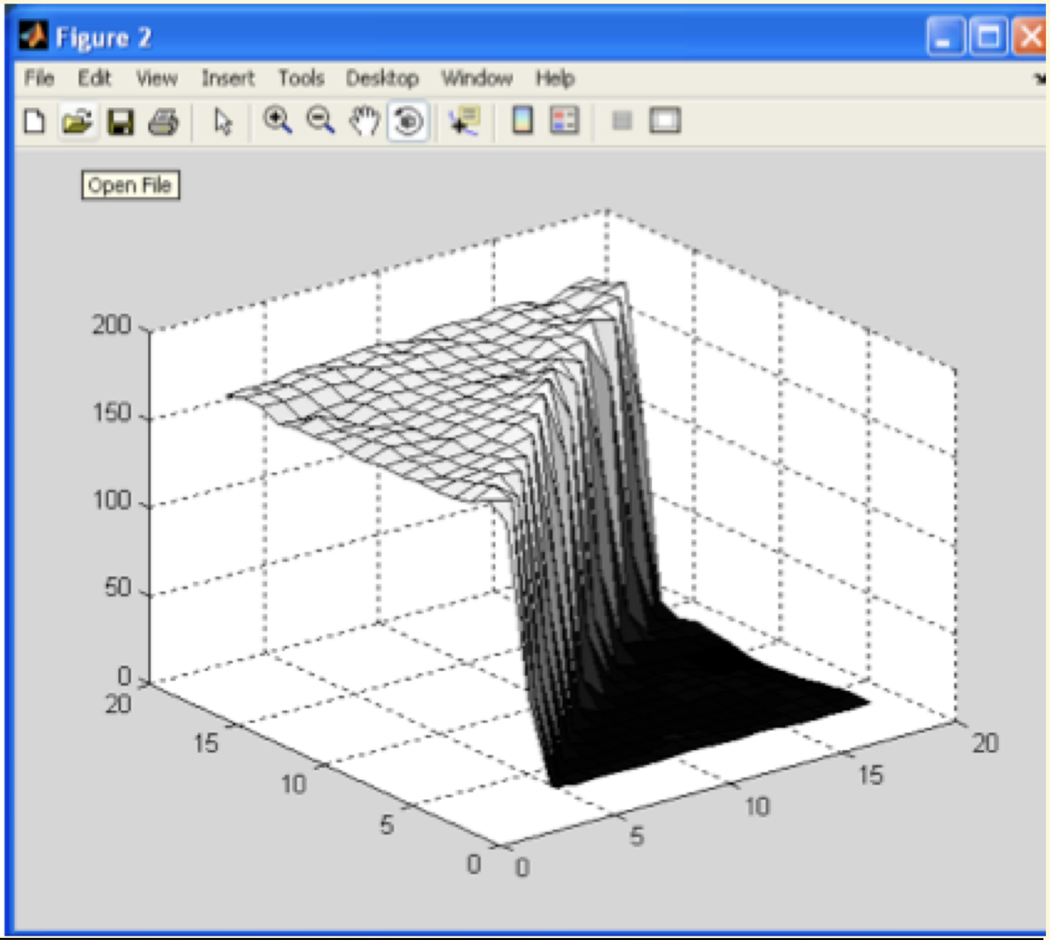
Surface height proportional to pixel grey value
(dark=low, light=high)

Images as surfaces

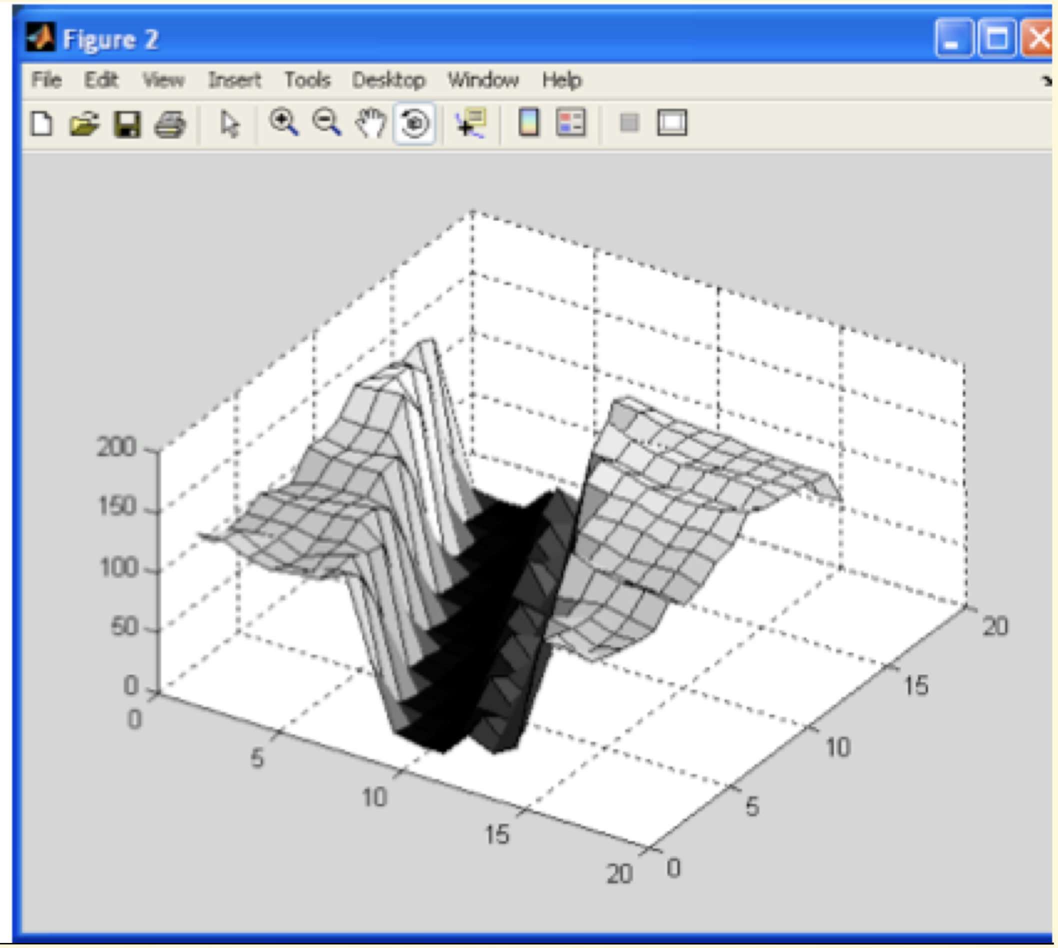


Mean = 164 Std = 1.8

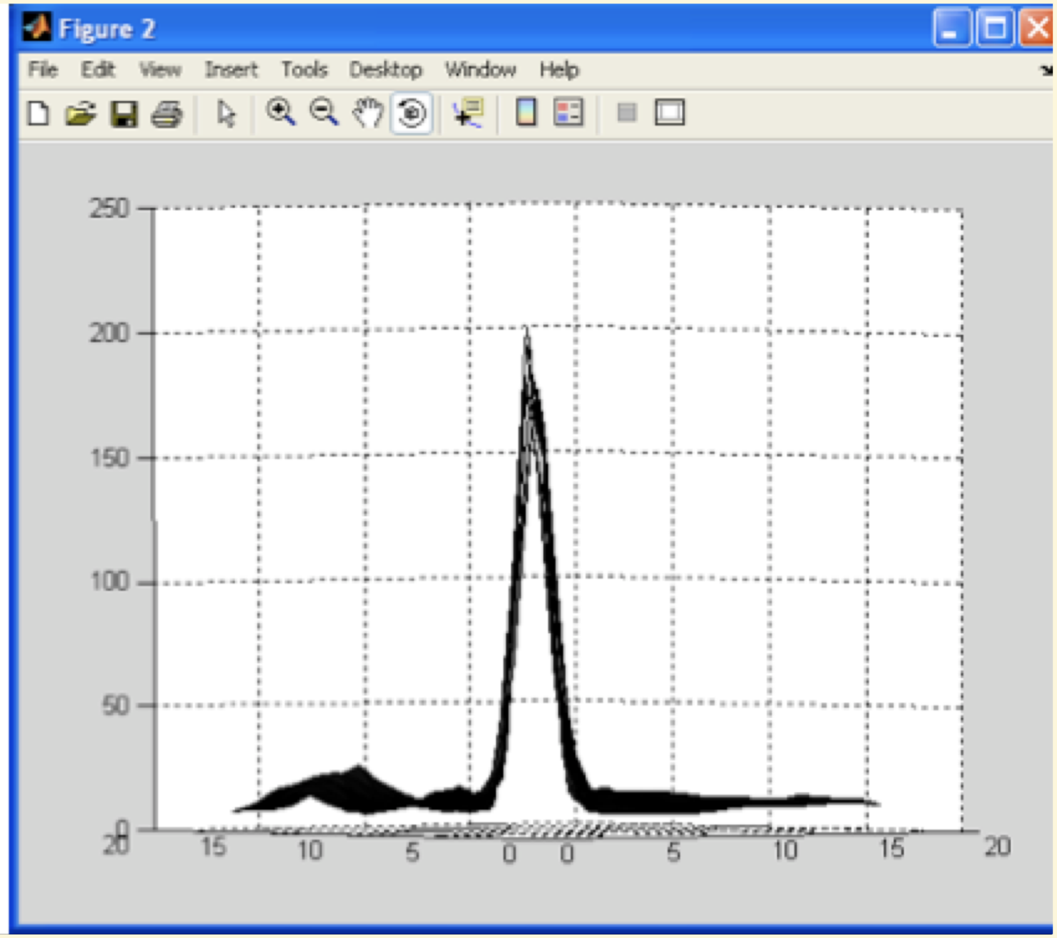
Images as surfaces



Images as surfaces

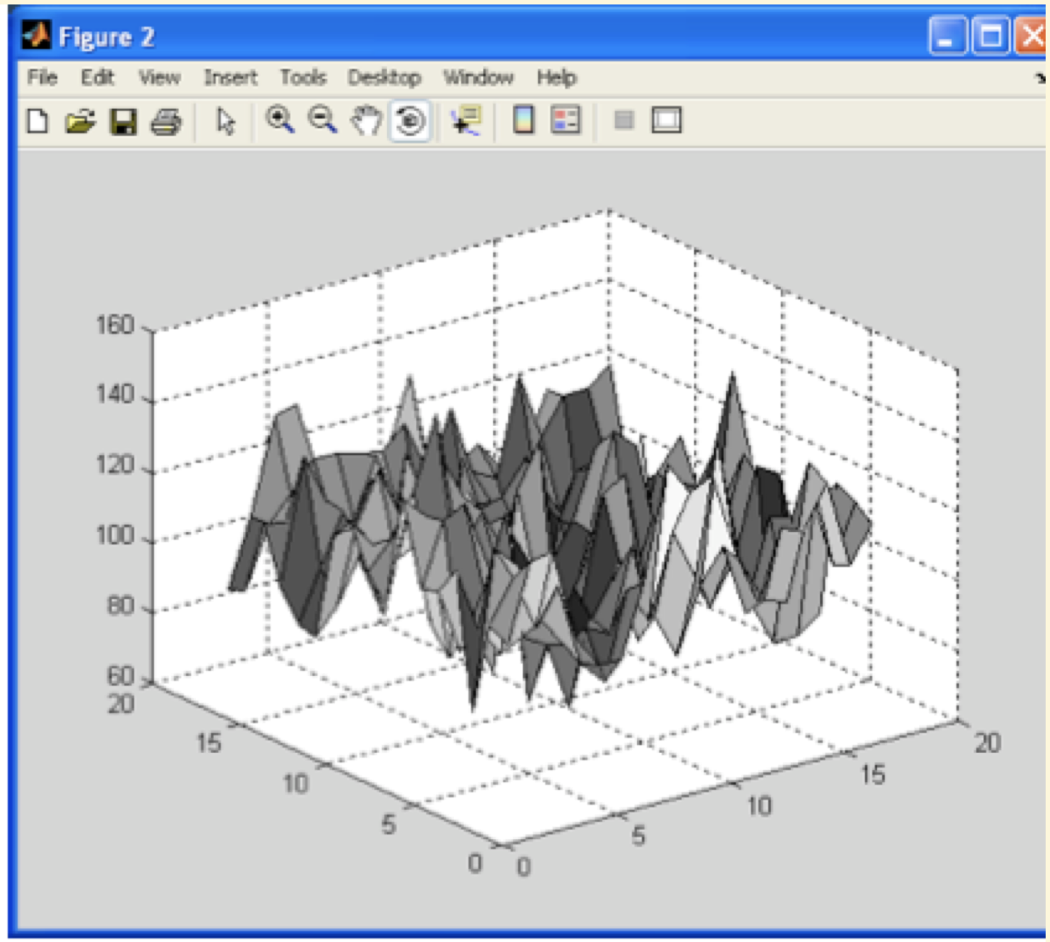


Images as surfaces



University of Colorado **Boulder**

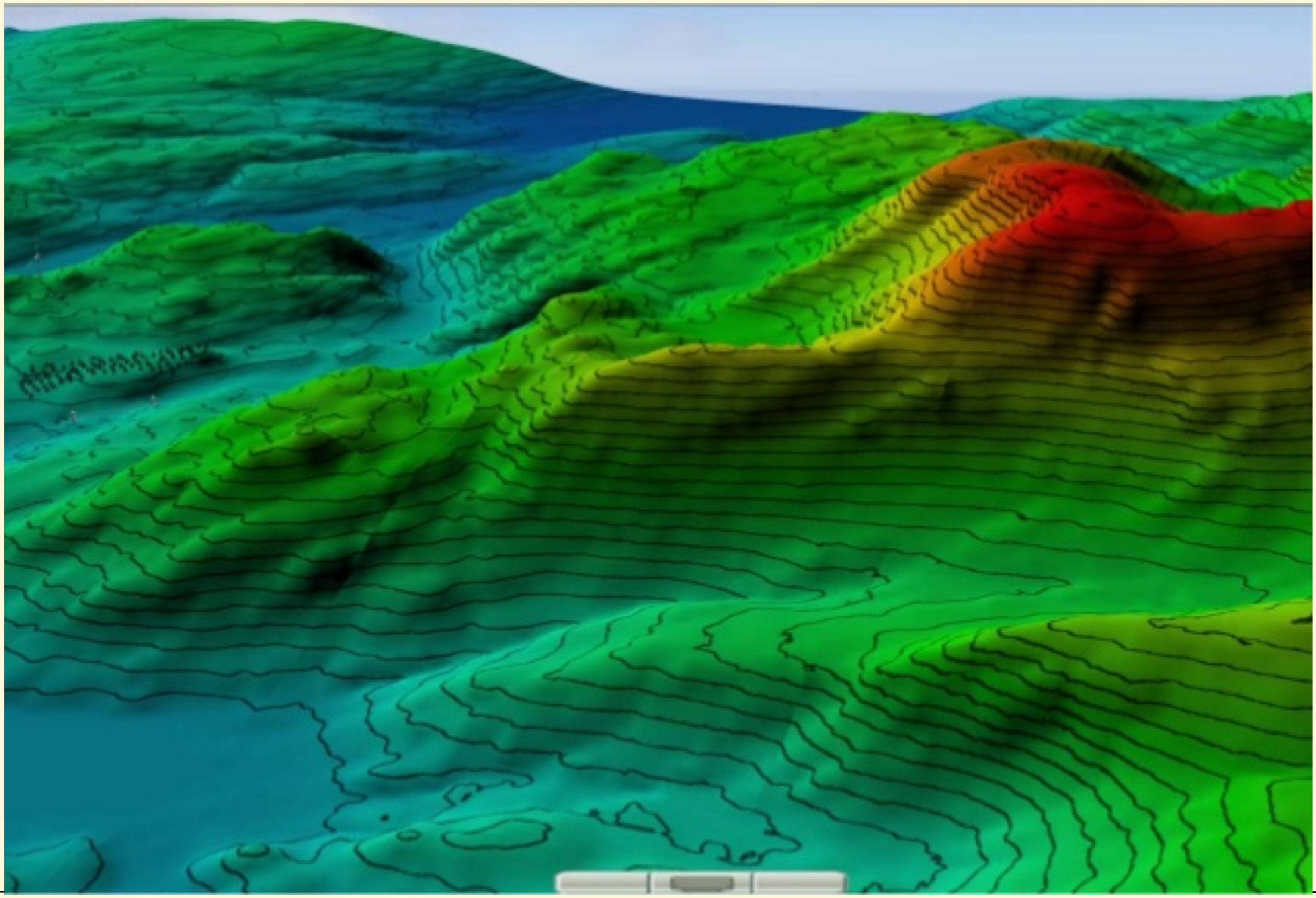
Images as surfaces



Mean = 111 Std = 15.4

How does this visualization help us?

Terrain Concepts



Terrain Concepts

Basic notions:

- Uphill / downhill
- Contour lines (curves of constant elevation)
- Steepness of slope
- Peaks/Valleys (local extrema)

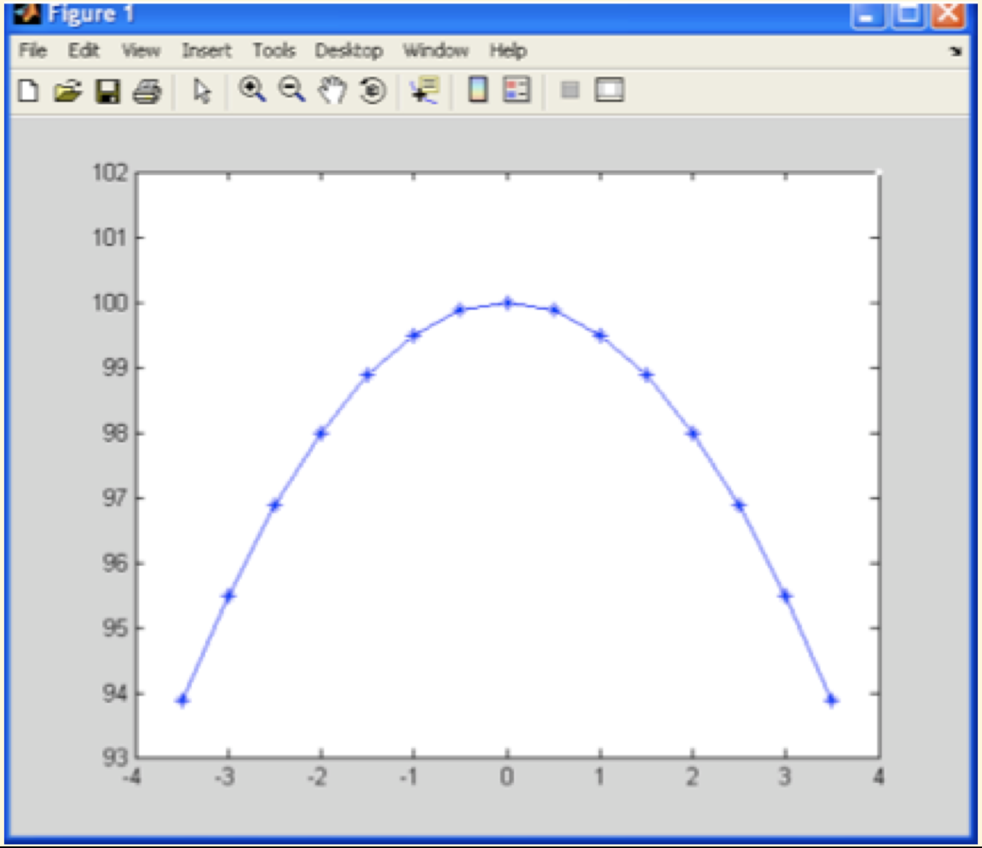
More mathematical notions:

- Tangent Plane
- Normal vectors
- Curvature

Gradient vectors (vectors of partial derivatives) will help us define/compute all of these.

1 D Gradient

Consider function $f(x) = 100 - 0.5 * x^2$

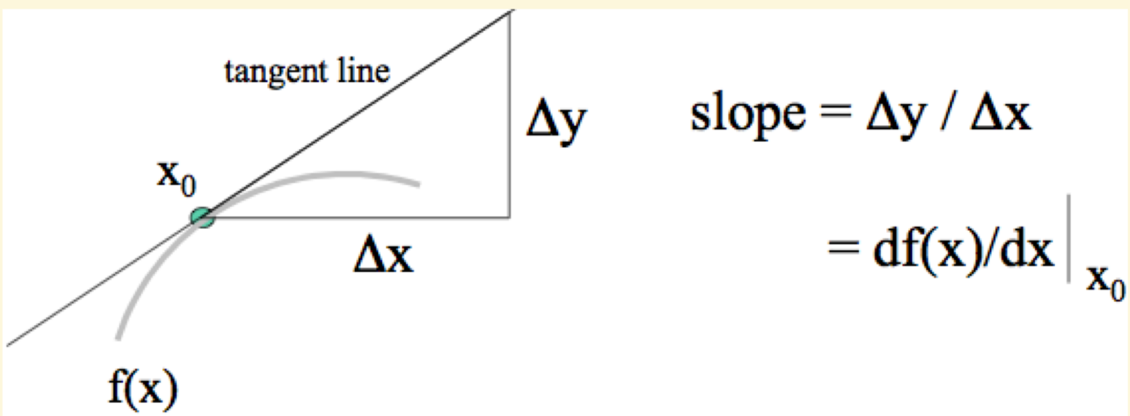


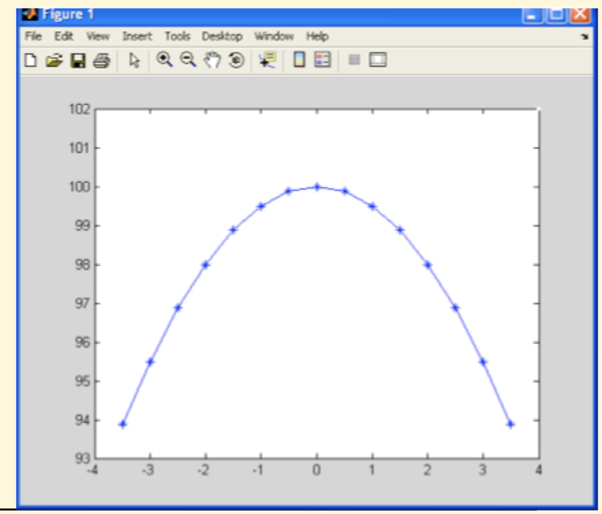
1 D Gradient

Consider function f(x) = 100 - 0.5 * x^2

Gradient is df(x)/dx = -2 * 0.5 * x = -x

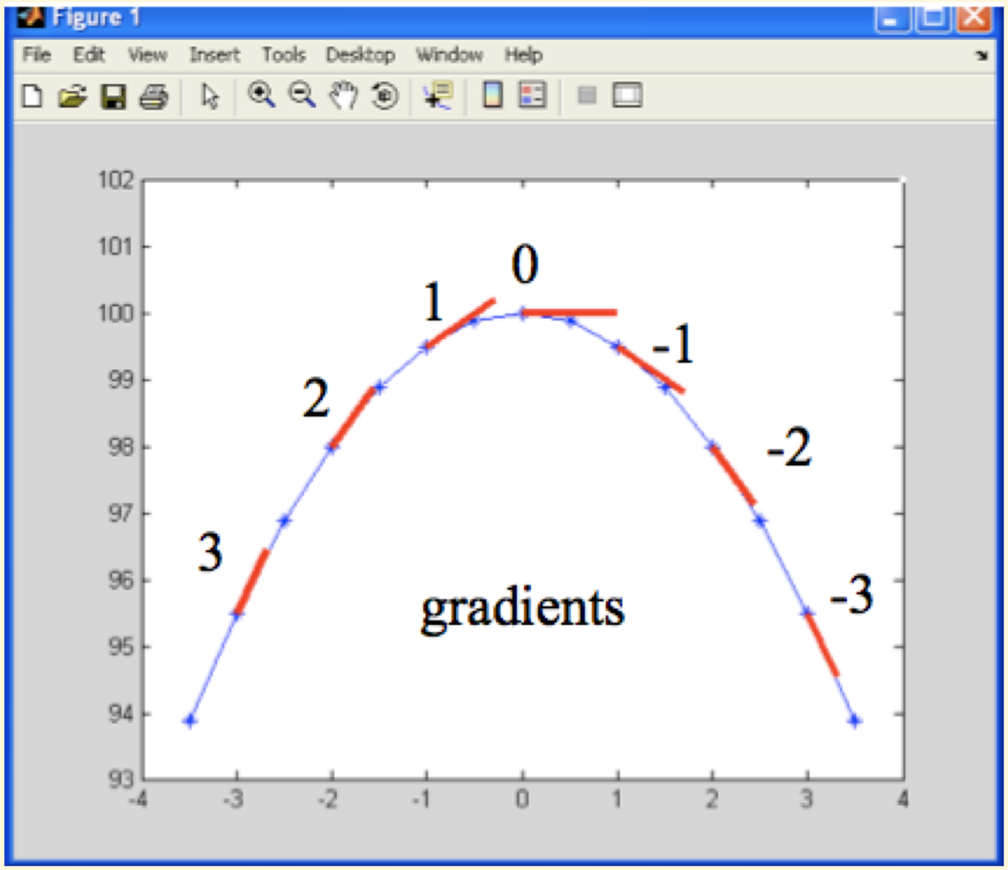
Geometric interpretation:
gradient at x_0 is slope of tangent line to curve at point x_0





1 D Gradient

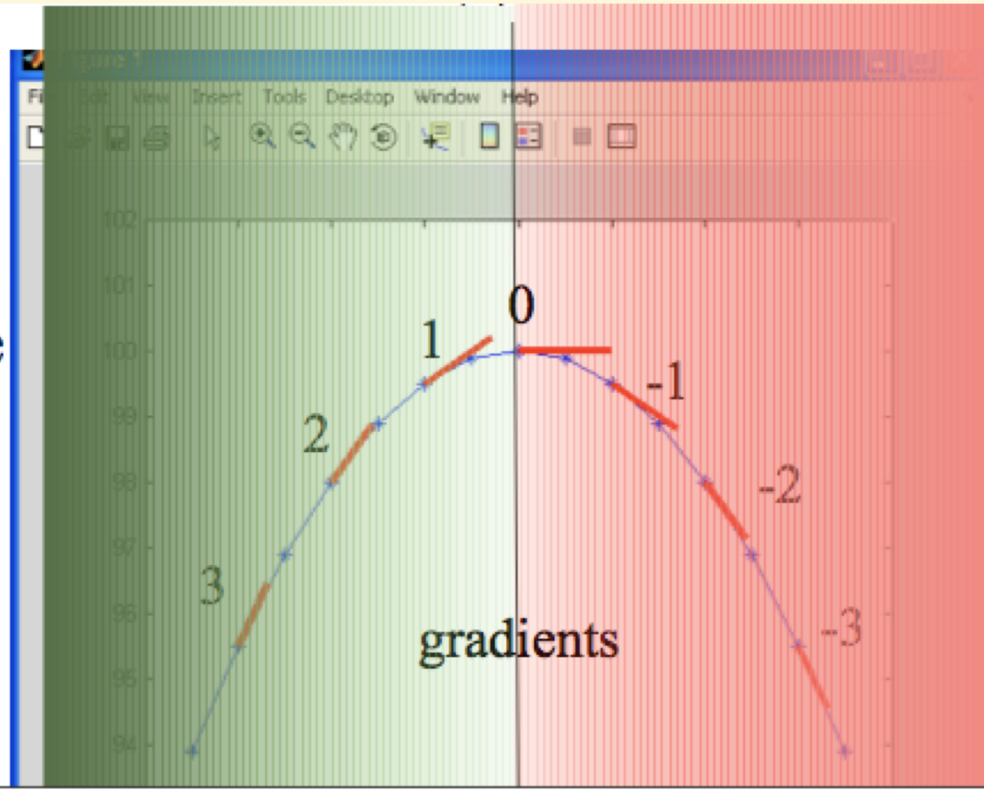
f(x) = 100 - 0.5 * x^2 df(x)/dx = -x



1 D Gradient

$f(x) = 100 - 0.5 * x^2$ $\quad\quad$ $df(x)/dx = -x$

Gradients on this side of peak are positive



Gradients on this side of peak are negative

Note: Sign of gradient at point tells you what direction to go to travel "uphill"

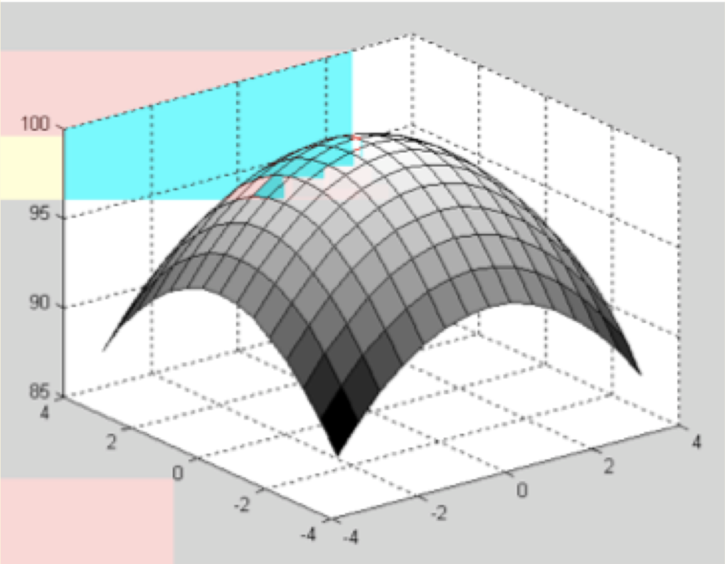
2 D Gradient

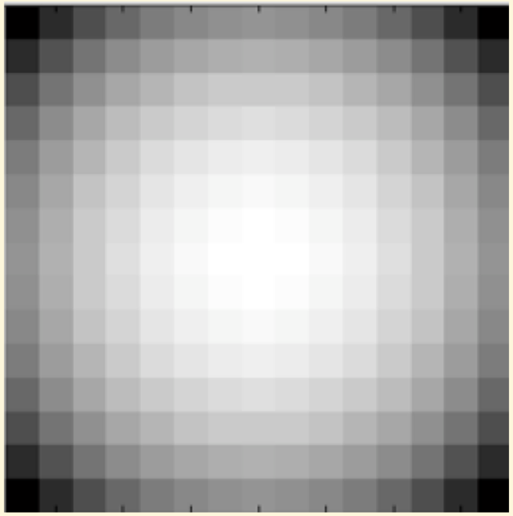
f(x,y) = 100 - 0.5 * x^2 - 0.5 * y^2

df(x,y)/dx = -x

df(x,y)/dy = -y

Gradient = [df(x,y)/dx, df(x,y)/dy] = [-x , -y]



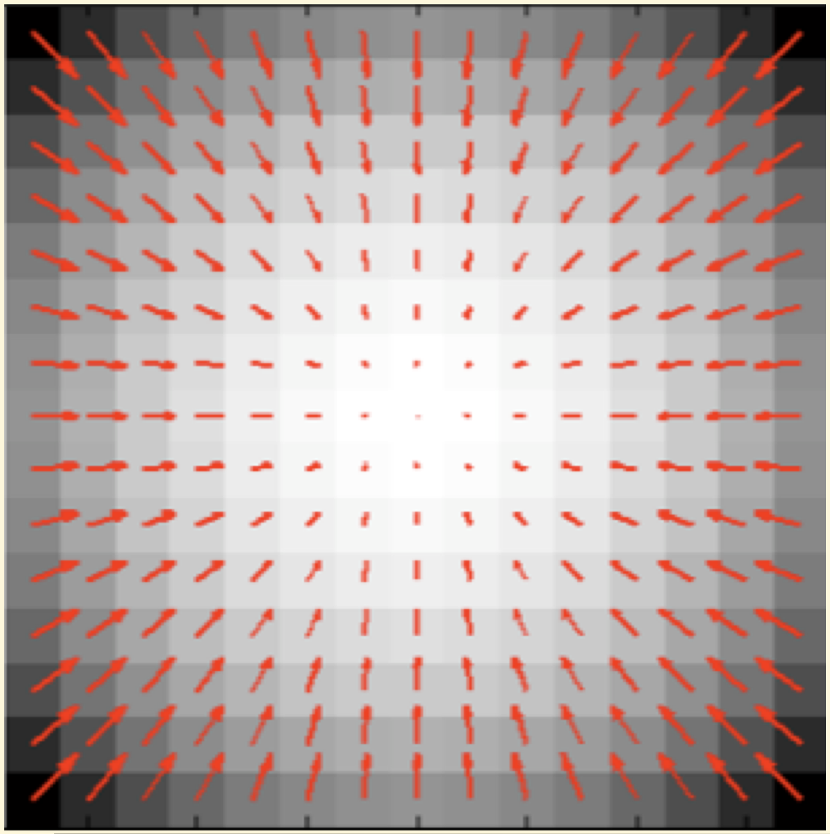


Gradient is vector of partial derivs wrt x and y axes

2 D Gradient

f(x,y) = 100 - 0.5 * x^2 - 0.5 * y^2

Gradient = [df(x,y)/dx, df(x,y)/dy] = [-x , -y]



Plotted as a vector field,
the gradient vector at each pixel points "uphill".

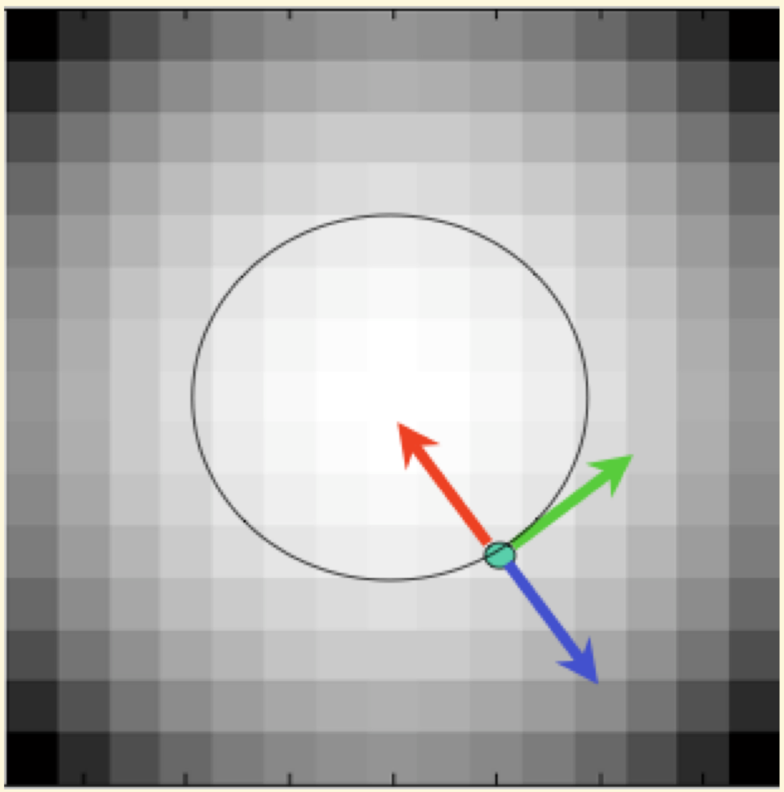
The gradient indicates the direction of steepest ascent.

The gradient is 0 at the peak
(also at any flat spots, and local minima)

2 D Gradient

$f(x,y) = 100 - 0.5 * x^2 - 0.5 * y^2$

Gradient = $[df(x,y)/dx, df(x,y)/dy] = [-x , -y]$



Let $g=[g_x,g_y]$ be the gradient vector at point/pixel (x_0,y_0)

Vector g points uphill

(direction of steepest ascent)

Vector -g points downhill

(direction of steepest descent)

Vector [gy, -gx] is perpendicular, and denotes direction of constant elevation. i.e.normal to contour line passing through point (x_0,y_0)

Numerical Derivatives

Taylor's Expansion

$$f(x+h) = f(x) + f'(x)h + \frac{1}{2} f''(x)h^2 + O(h^3)$$

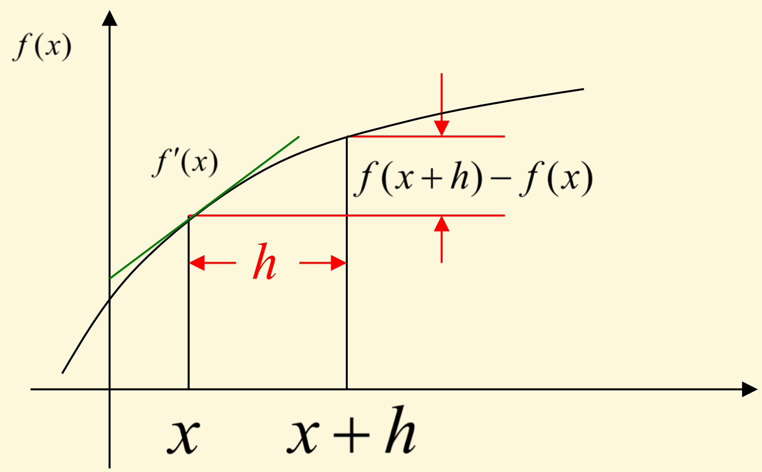
$$f'(x)h = f(x+h) - f(x) - \frac{1}{2} f''(x)h^2 + O(h^3)$$

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{1}{2} f''(x)h + O(h^2)$$

Forward Difference Formula for $f'(x)$

$$\boxed{f'(x) \approx \frac{f(x+h) - f(x)}{h} \qquad error = O(h)}$$

Geometrically



Backward Difference Formula for $f'(x)$

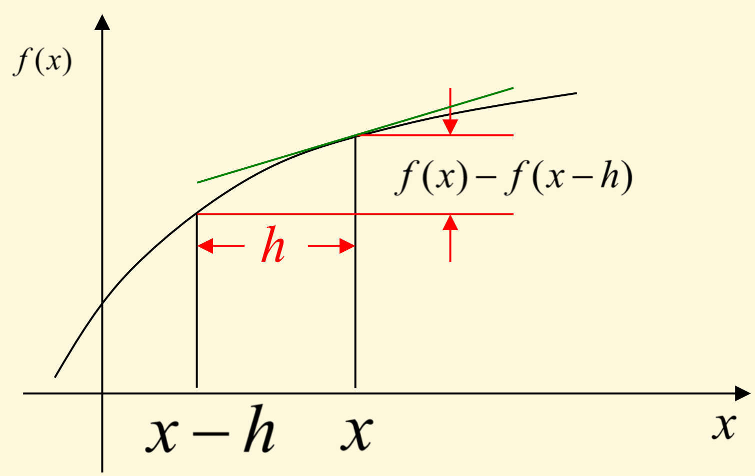
Similarly

$$f(x-h) = f(x) - f'(x)h + \frac{1}{2} f''(x)h^2 + O(h^3)$$

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

$$error = O(h)$$

Geometrically



Central Difference Formula for $f'(x)$

$$f(x+h) = f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \frac{1}{6}f'''(x)h^3 + \frac{1}{4!}f^{(4)}(x)h^4 + O(h^5)$$

$$-)\quad f(x-h) = f(x) - f'(x)h + \frac{1}{2}f''(x)h^2 - \frac{1}{6}f'''(x)h^3 + \frac{1}{4!}f^{(4)}(x)h^4 + O(h^5)$$

$$f(x+h) - f(x-h) = 2hf'(x) \qquad + \frac{1}{3}f'''(x)h^3 \qquad + O(h^5)$$

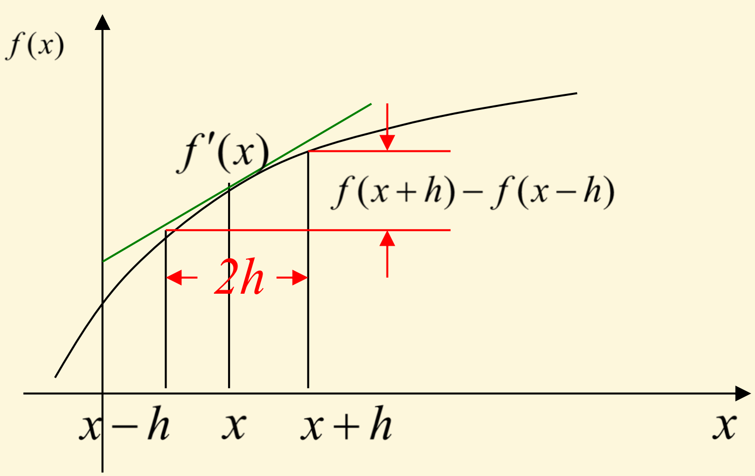
$$2hf'(x) = f(x+h) - f(x-h) - \frac{1}{3}f'''(x)h^3 + O(h^5)$$

Central Difference Formula for $f'(x)$

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

$$error = O(h^2)$$

Geometrically



Example

$$f(x) = x^3 \qquad (\therefore f'(x) = 3x^2 \quad , \quad f'(1) = 3)$$

Calculate f'(1) *using FD, BD, CD*

Example (cont)

FD:

$h=0.1$ $\quad f'(1)=\dfrac{f(1.1)-f(1)}{0.1}=3.31$ $\quad error = 0.31$

$h=0.05$ $\quad f'(1)=\dfrac{f(1.05)-f(1)}{0.05}=3.1525$ $\quad error = 0.1525$

$\Rightarrow error \propto h$

BD:

$h=0.1$ $\quad f'(1)=\dfrac{f(1)-f(0.9)}{0.1}=2.71$ $\quad error = 0.29$

$h=0.05$ $\quad f'(1)=\dfrac{f(1)-f(0.95)}{0.05}=2.8453$ $\quad error = 0.1547$

$\Rightarrow error \propto h$

CD:

$h=0.1$ $\quad f'(1)=\dfrac{f(1.1)-f(0.9)}{0.2}=3.01$ $\quad error = 0.01$

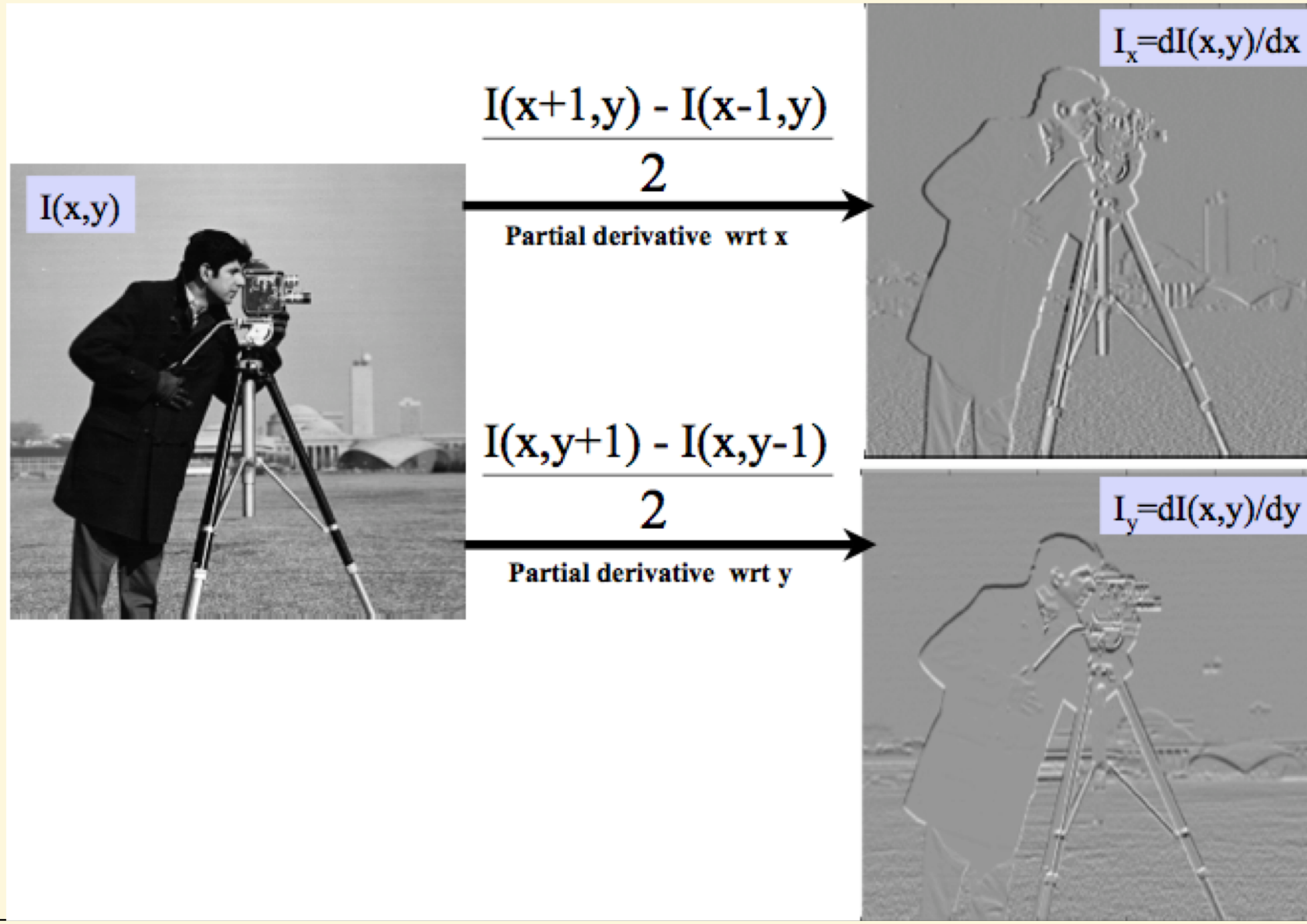
$h=0.05$ $\quad f'(1)=\dfrac{f(1.05)-f(0.95)}{0.1}=3.00250$ $\quad error = 0.00250$

$\Rightarrow error \propto h^2$

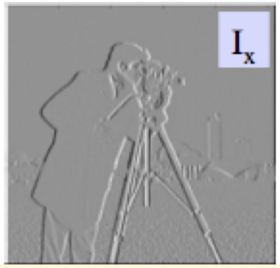
Example (cont)

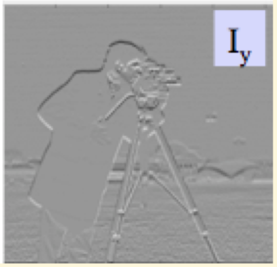
- Remarks:
  - FD, BD, CD each involves 2 function calls, 1 subtraction, and 1 division: same computation time
  - CD is the most accurate (hence, the most recommended method)
  - *However*, sometimes, CD cannot be applied

Example - Spatial Image Gradients



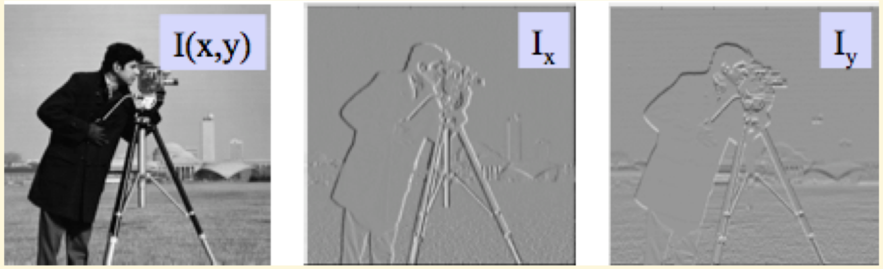
Functions of Gradients



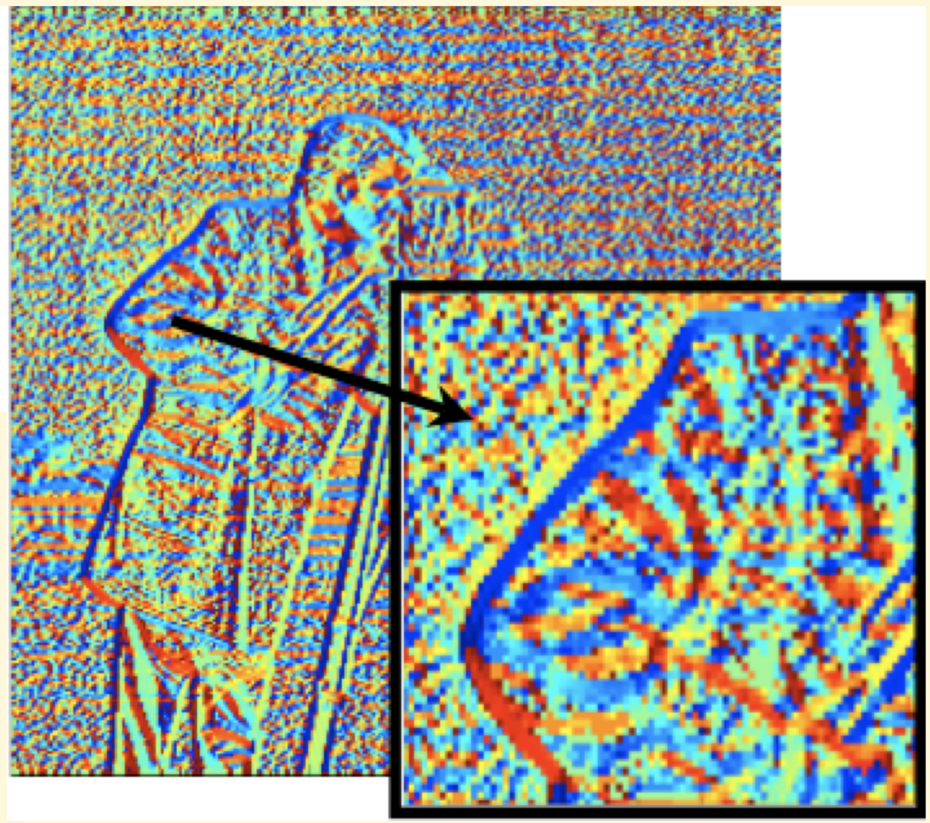


Magnitude of gradient
sqrt(Ix.^2 + Iy.^2)

Functions of Gradients

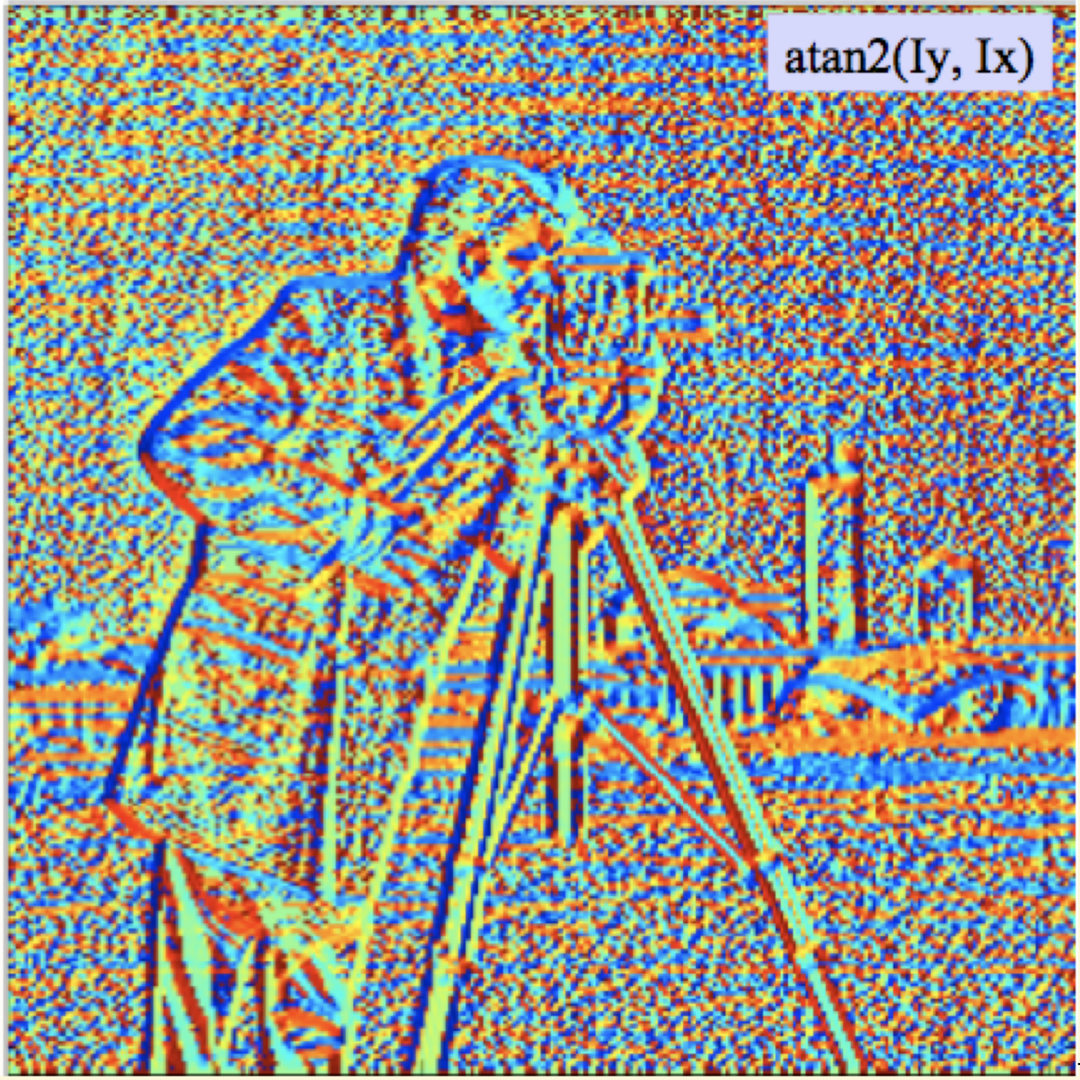


Angle of gradient
atan2(Iy, Ix)



Functions of Gradients

What else do we observe in this image?



Gradients as linear operators

Gradients are an example of linear operators, i.e. value at a pixel is computed as a linear combination of values of neighboring pixels.

Image filtering

- Image filtering: compute function of local neighborhood at each position

- Really important!
  - Enhance images
    - Denoise, resize, increase contrast, etc.
  - Extract information from images
    - Texture, edges, distinctive points, etc.
  - Detect patterns
    - Template matching

Example: box filter

$$g[\cdot\,,\cdot\,]$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Slide credit: David Lowe (UBC)

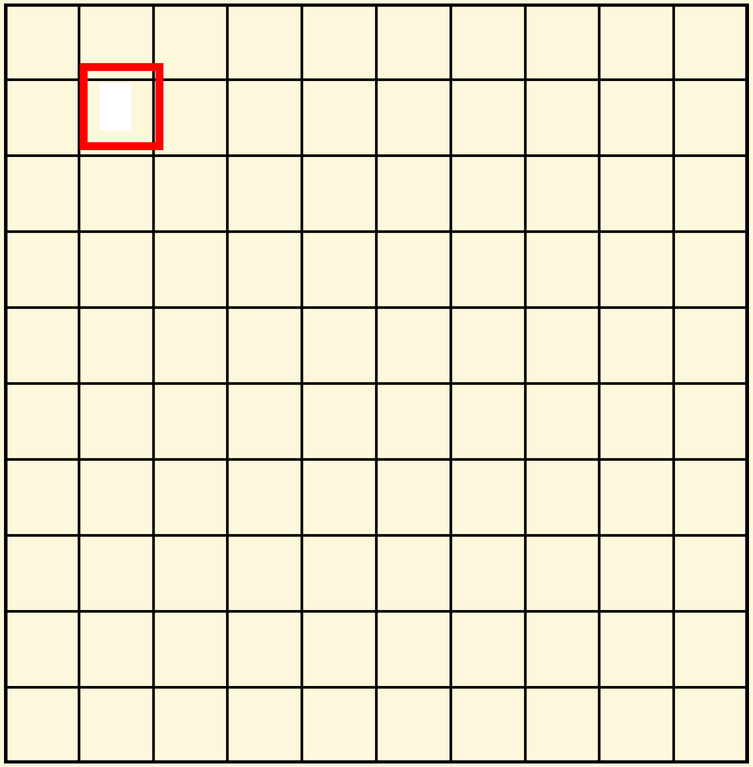
Image filtering

$$g[\cdot\,,\cdot\,] \qquad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[.,.]$$



$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

Image filtering

$$g[\cdot,\cdot]\ \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[.,.]$$

	0	10							

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20						

$$h[m,n]=\sum_{k,l} g[k,l]\,f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[\cdot,\cdot]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$h[.,.]$

	0	10	20	30					

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

Credit: S. Seitz

Image filtering

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

University of Colorado **Boulder**

Credit: S. Seitz

Image filtering

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				
				?					

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				
						?			
				50					

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Image filtering

$$g[\cdot,\cdot] \quad \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[.,.]$$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

Credit: S. Seitz

Box Filter

What does it do?

- Replaces each pixel with an average of its neighborhood
- Achieve smoothing effect (remove sharp features)

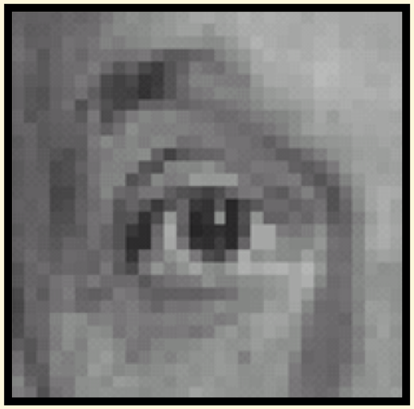
$$g[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Slide credit: David Lowe (UBC)

Smoothing with box filter

Practice with linear filters



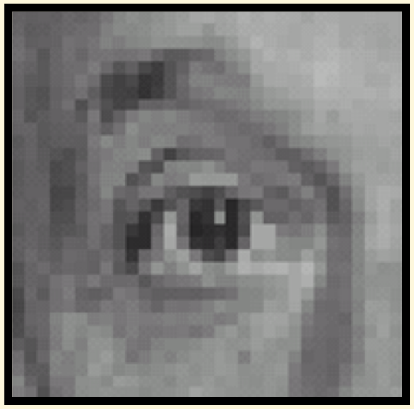
Original

0	0	0
0	1	0
0	0	0

?

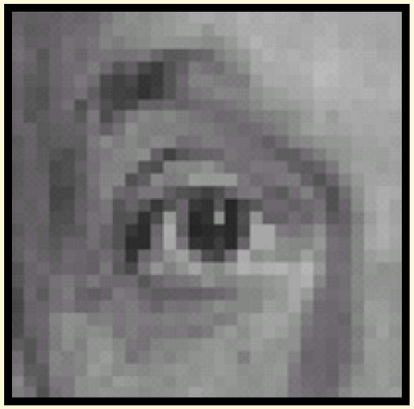
Source: D. Lowe

Practice with linear filters



Original

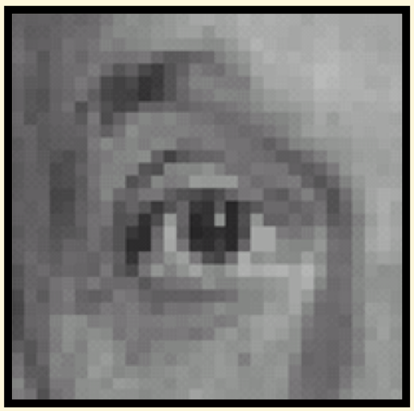
0	0	0
0	1	0
0	0	0



Filtered
(no change)

Source: D. Lowe

Practice with linear filters



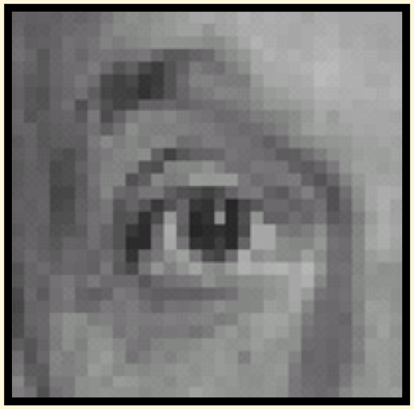
Original

0	0	0
0	0	1
0	0	0

?

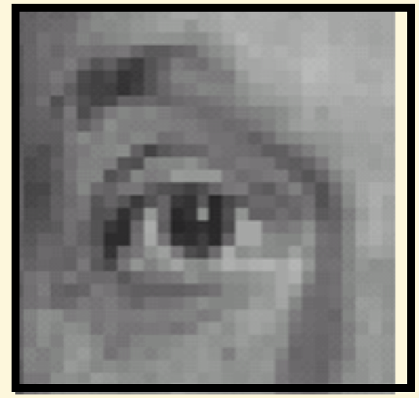
Source: D. Lowe

Practice with linear filters



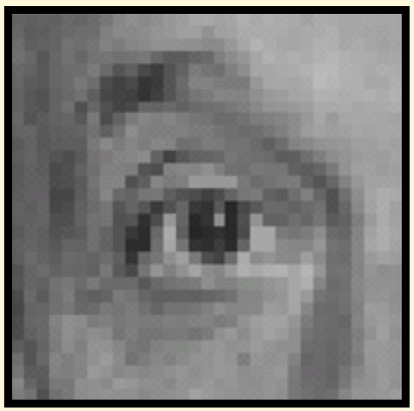
Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Practice with linear filters



Original

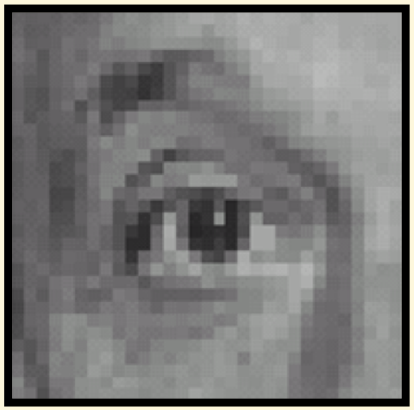
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

?

(Note that filter sums to 1)

Source: D. Lowe

Practice with linear filters



Original

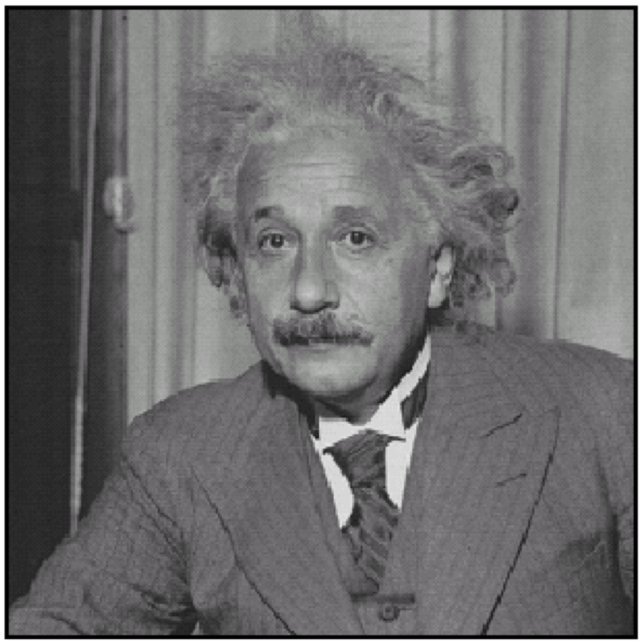
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpening filter

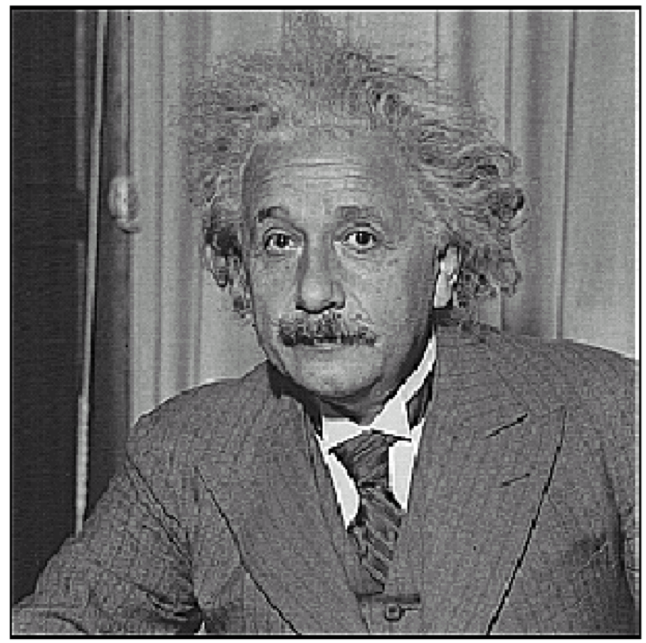
- Accentuates differences with local average

Source: D. Lowe

Sharpening



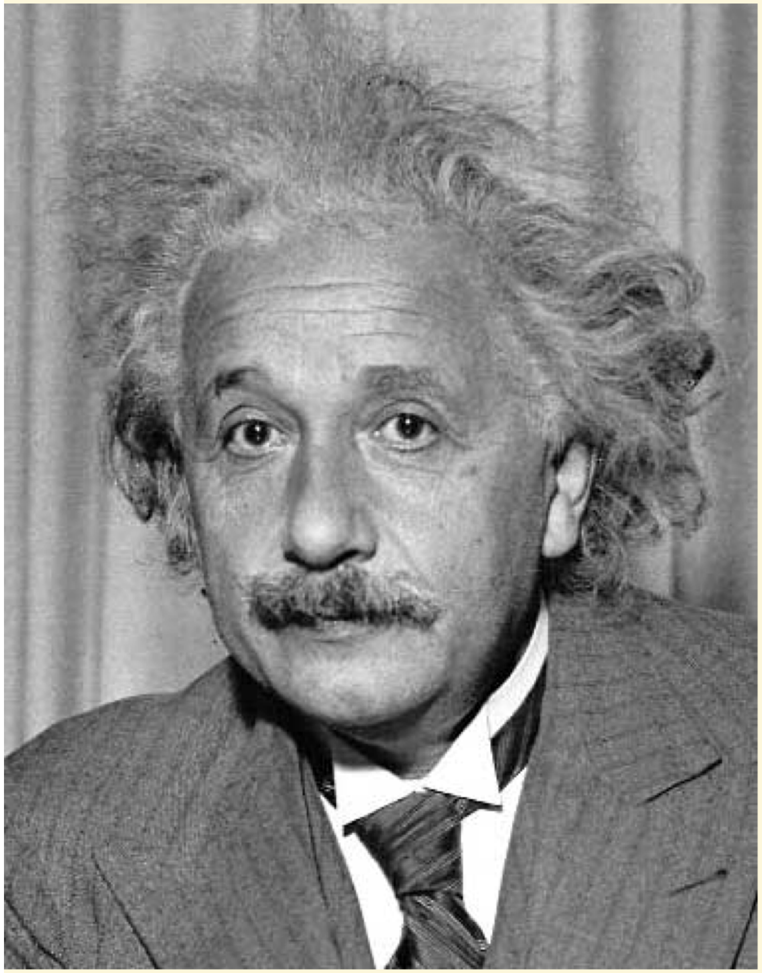
before



after

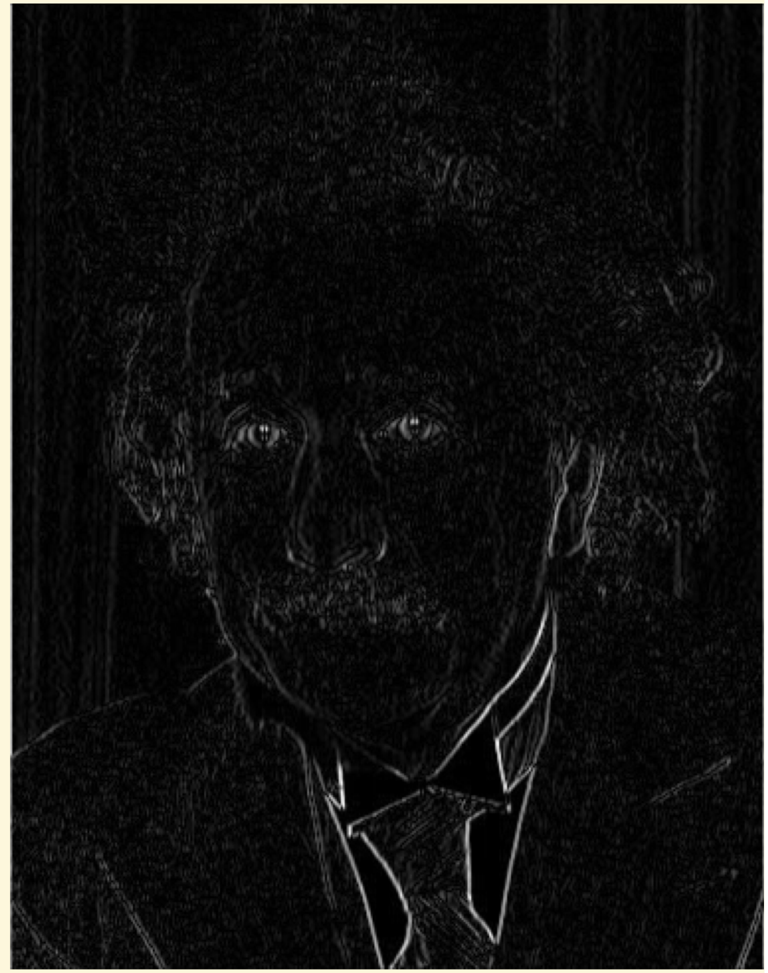
Source: D. Lowe

Other filters



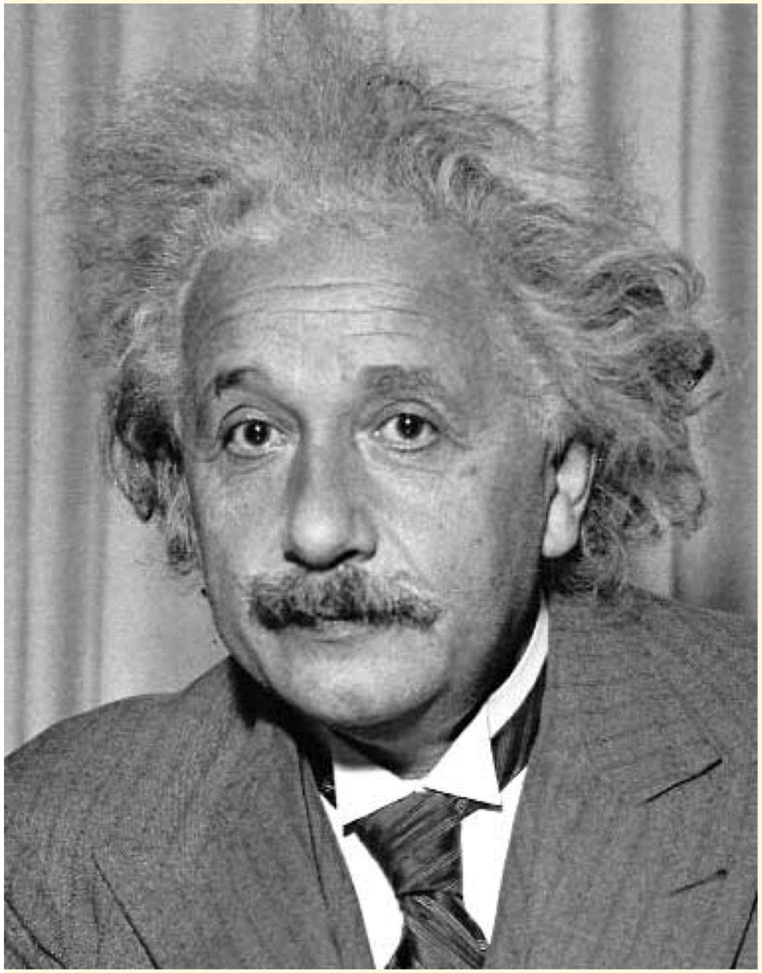
1	0	-1
2	0	-2
1	0	-1

Sobel



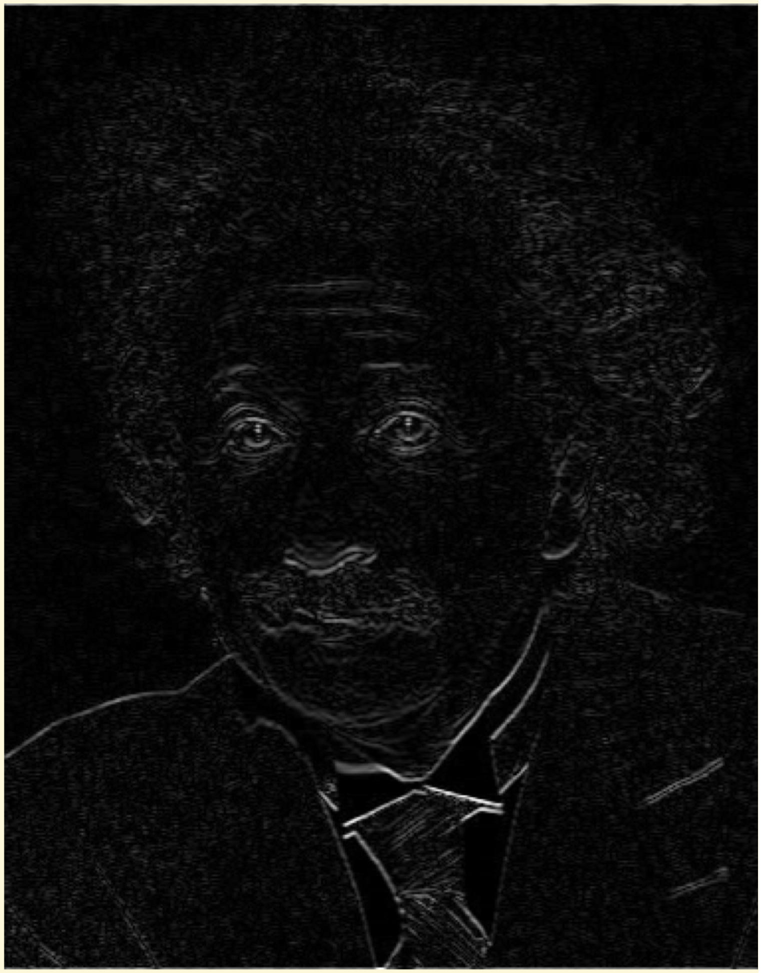
Vertical Edge
(absolute value)

Other filters



1	2	1
0	0	0
-1	-2	-1

Sobel



Horizontal Edge
(absolute value)