CS4501: Introduction to Computer Vision

Filtering, Frequency, and Edges

Various slides from previous courses by:
D.A. Forsyth (Berkeley / UIUC), I. Kokkinos (Ecole Centrale / UCL). S. Lazebnik (UNC / UIUC), S. Seitz (MSR / Facebook), J. Hays (Brown / Georgia Tech), A. Berg (Stony Brook / UNC), D. Samaras (Stony Brook) . J. M. Frahm (UNC), V. Ordonez (UVA).

Last Class

- Image Processing: Brightness
- Image Filtering: Mean Filter
- Image Blurring
- Image Gradients: The Sobel Operator

Today's Class

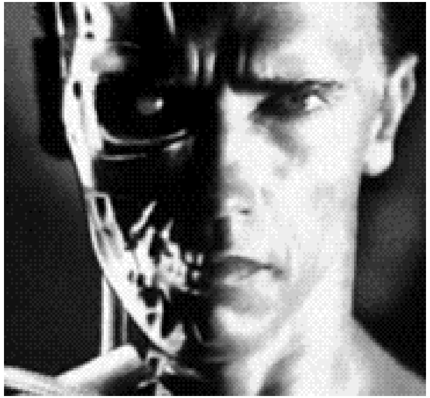
- Recap on Convolutional Operations
- Image Gradients: The Sobel Operator
- Frequency Domain
- Filtering in Frequency
- Google Colaboratory

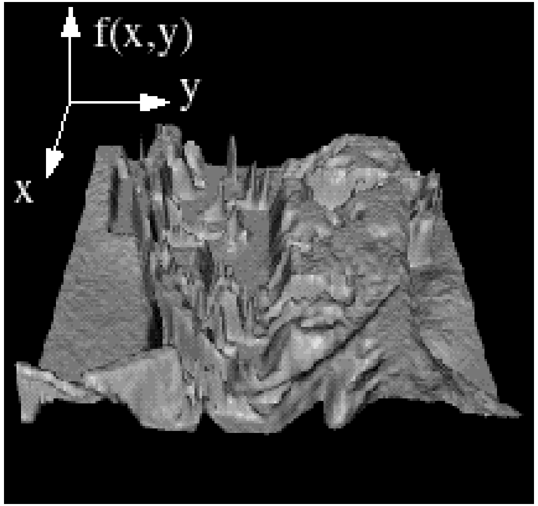
Images as Matrices

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

Images as Functions

$$z = f(x, y)$$

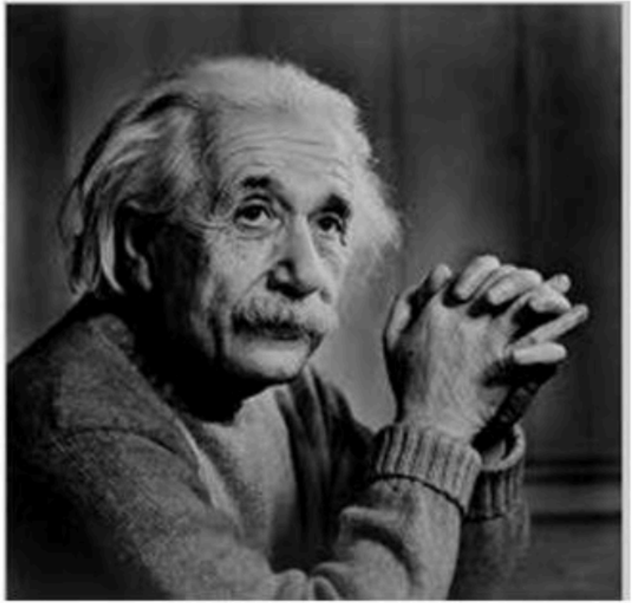




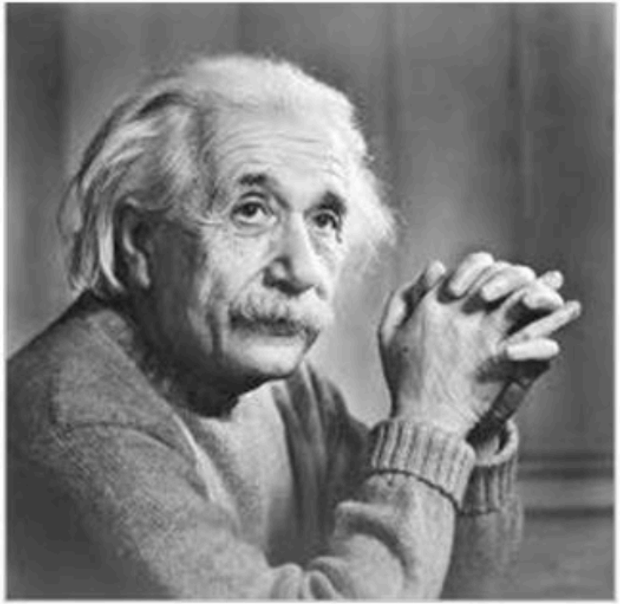
- The domain of x and y is [0, img-width) and [0 and img-height)
- x, and y are discretized into integer values.

Basic Image Processing

I



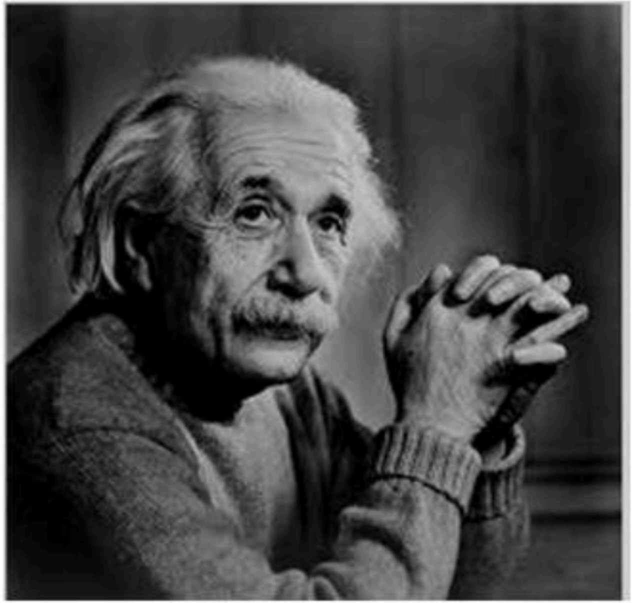
αI



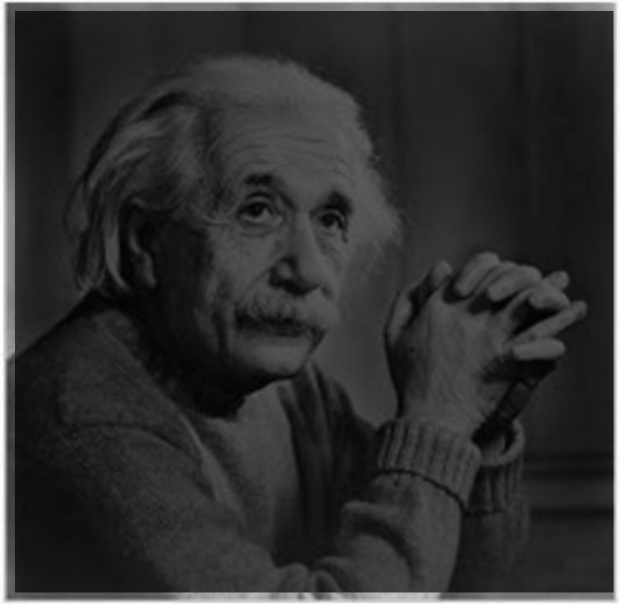
$\alpha > 1$

Basic Image Processing

I

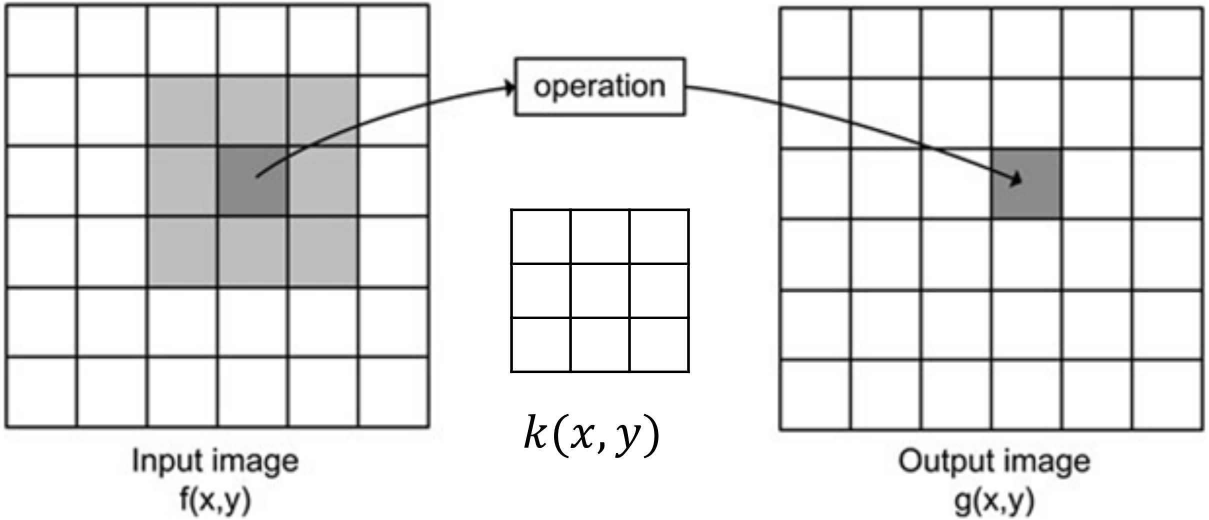


αI



$0 < \alpha < 1$

Image filtering: Convolution operator

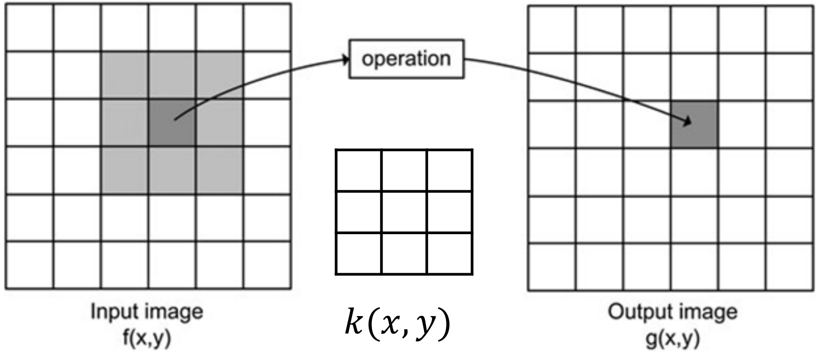


$k(x, y)$

$$g(x,y) = \sum_{v} \sum_{u} k(u,v) f(x - u, y - v)$$

Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Image filtering: Convolution operator
e.g. mean filter



$k(x, y)$ =

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

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

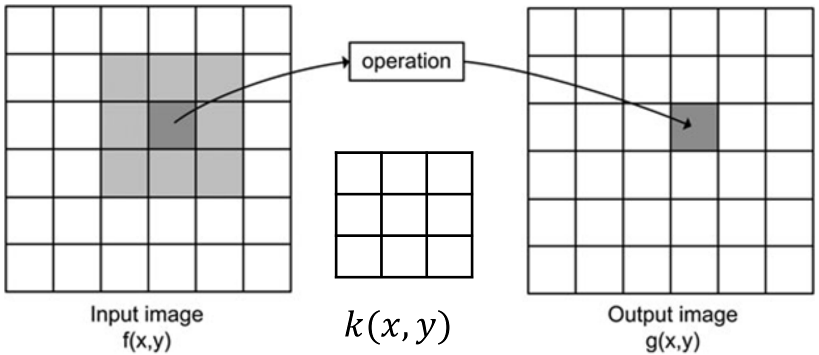
$h[.,.]$

	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

Image filtering: Convolution operator
Important filter: gaussian filter (gaussian blur)



$$k(x,y) = \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}$$

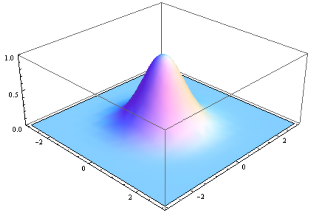
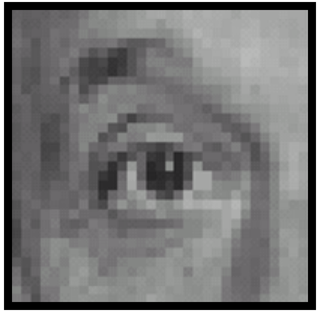


Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Image filtering: Convolution operator
e.g. gaussian filter (gaussian blur)

Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Sharpening Filter



Original

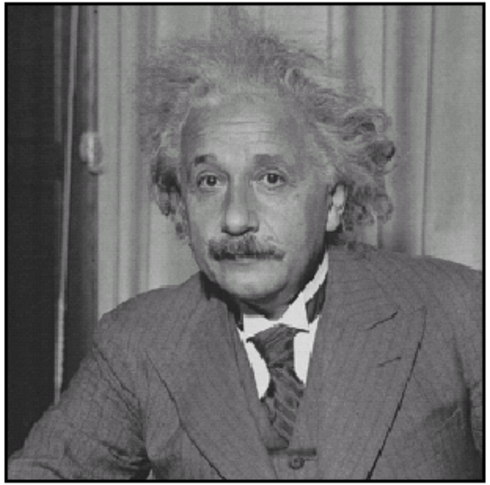
$$\begin{bmatrix}0&0&0\\0&2&0\\0&0&0\end{bmatrix} - \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

Sharpening filter

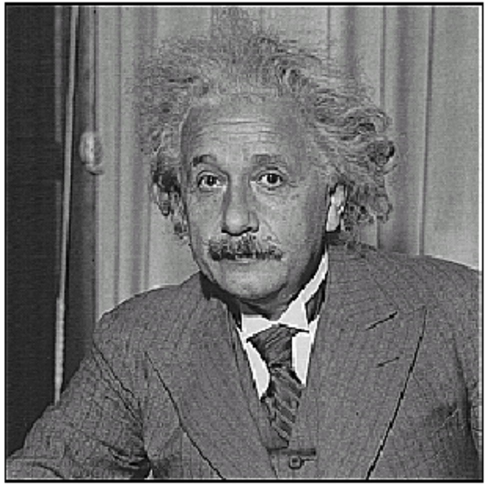
- Accentuates differences with local average

Source: D. Lowe

Sharpening Filter



before

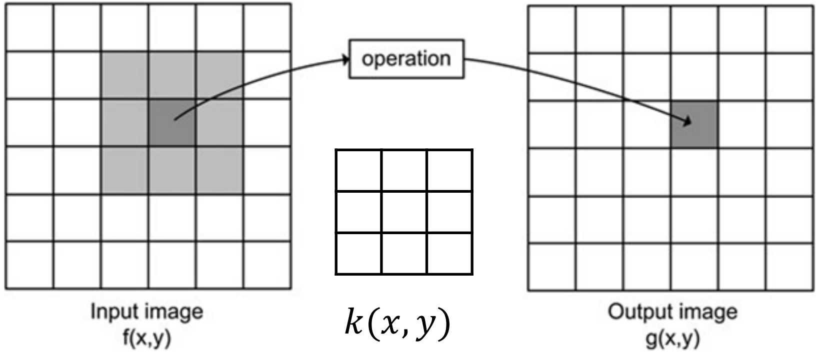


after

Source: D. Lowe

Image filtering: Convolution operator

Important Filter: Sobel operator

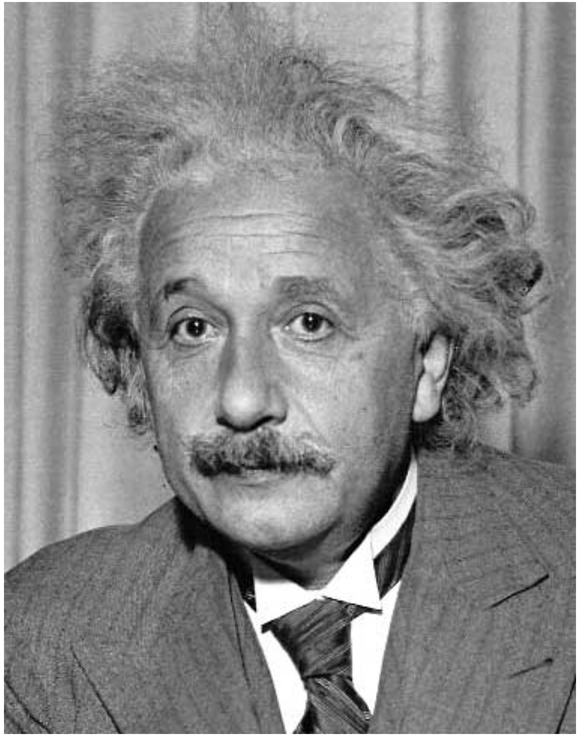


$k(x, y)$ =

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

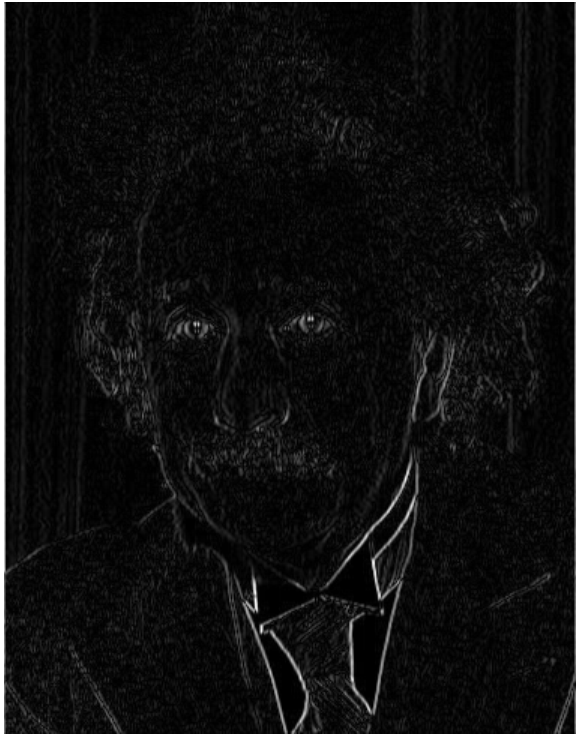
Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Sobel in X



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

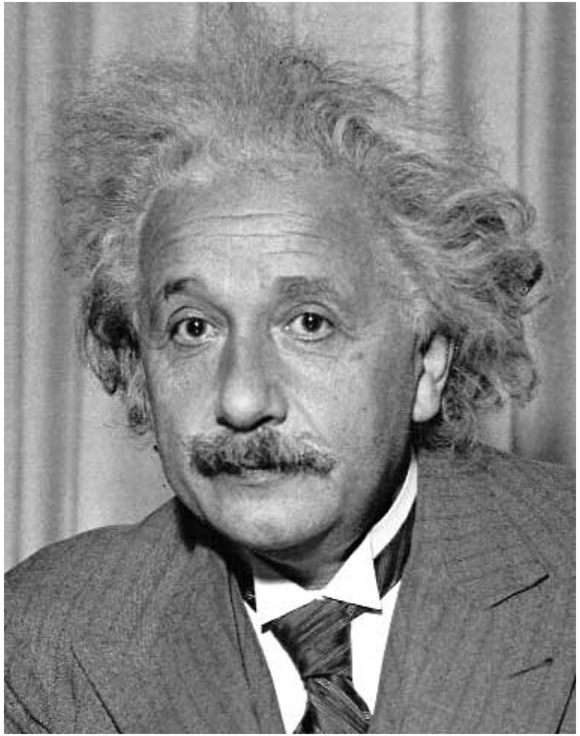
Sobel



Vertical Edge
(absolute value)

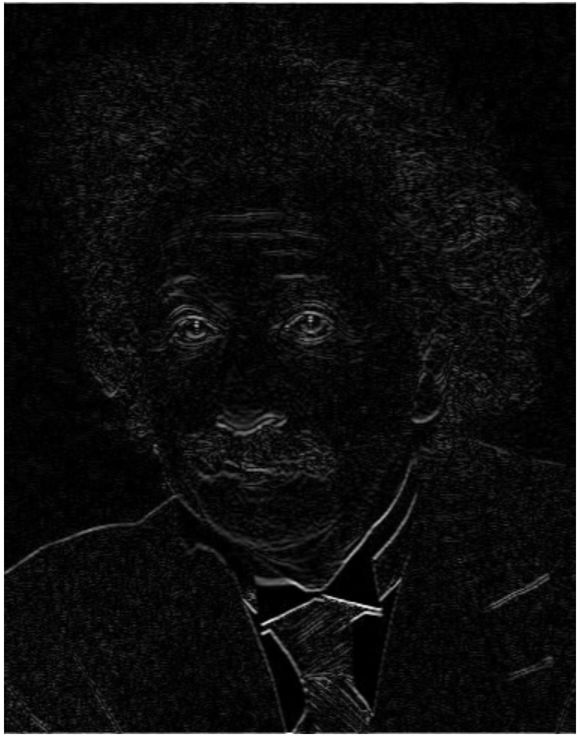
Slide by James Hays

Sobel in Y



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

Sobel



Horizontal Edge
(absolute value)

Slide by James Hays

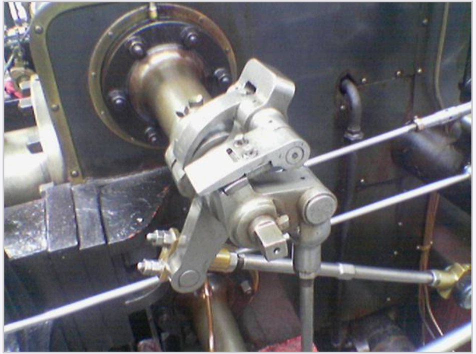
Sobel operators are equivalent to 2D partial derivatives of the image

- Vertical sobel operator – Partial derivative in X (width)
- Horizontal sobel operator – Partial derivative in Y (height)

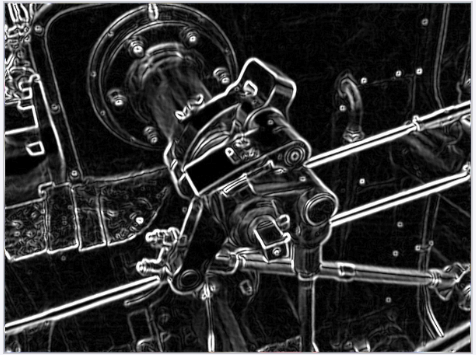
- Can compute magnitude and phase at each location.

- Useful for detecting edges

https://en.wikipedia.org/wiki/Sobel_operator



A color picture of a steam engine



The Sobel operator applied to that image

Sobel filters are (approximate) partial derivatives of the image

Let $f(x, y)$ be your input image, then the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{h \to 0} \frac{f(x+h, y) - f(x, y)}{h}$$

Also:
$$\frac{\partial f(x,y)}{\partial x} = \lim_{h \to 0} \frac{f(x+h, y) - f(x-h, y)}{2h}$$

But digital images are not continuous, they are discrete

Let $f[x, y]$ be your input image, then the partial derivative is:

$$\Delta_x f[x, y] = f[x + 1, y] - f[x, y]$$

Also: $$\Delta_x f[x, y] = f[x + 1, y] - f[x - 1, y]$$

But digital images are not continuous, they are discrete

Let $f[x, y]$ be your input image, then the partial derivative is:

$$\Delta_x f[x, y] = f[x+1, y] - f[x, y]$$

k(x, y) =

-1	1

Also: $$\Delta_x f[x, y] = f[x+1, y] - f[x-1, y]$$

k(x, y) =

-1	0	1

Sobel Operators Smooth in Y and then Differentiate in X

$$k(x, y) = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Similarly to differentiate in Y

Frequency

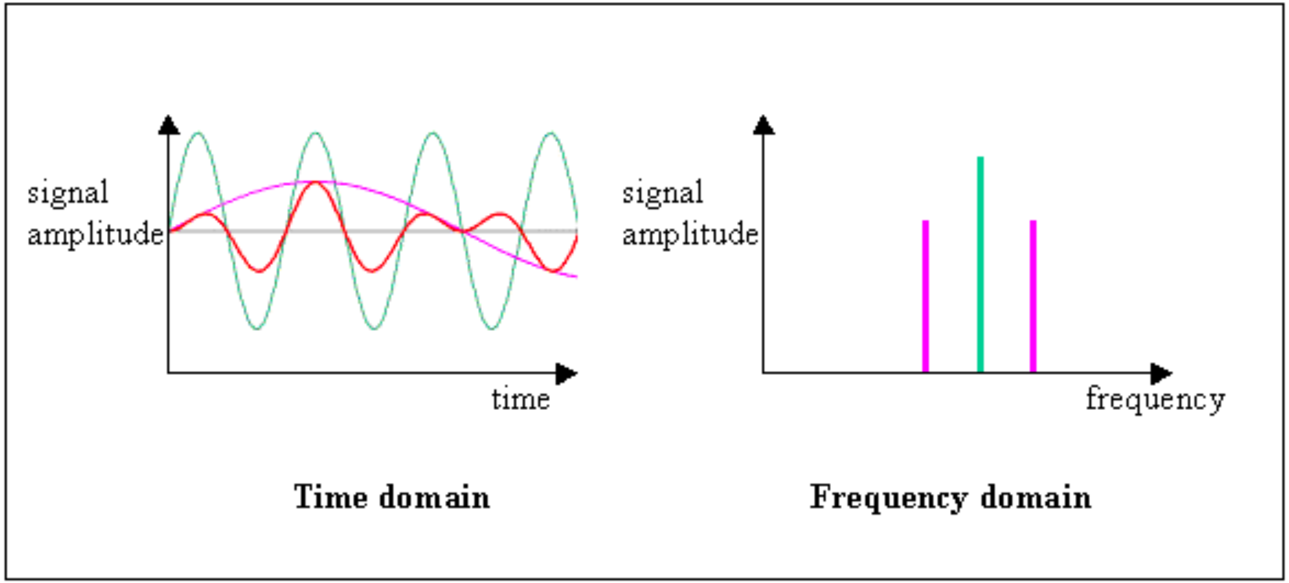


Figure by National Instruments

Any function can be approximated by a polynomial function

Taylor Series expansion

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \cdots.$$

…if you let your polynomial have a high degree

…AND you can compute the derivatives of the original function easily.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

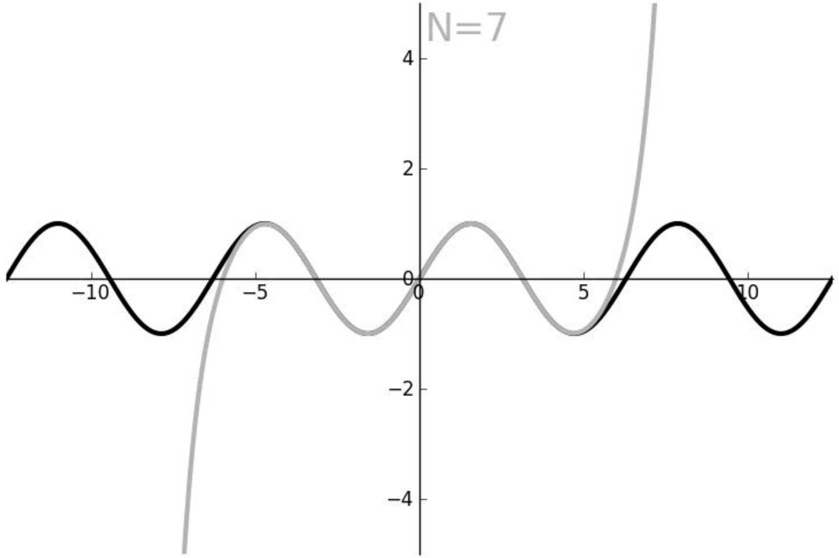
$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots \quad for\,|x| < 1$$

$$\tan^{-1}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots \quad for\,|x| < 1$$

Difficult in practice



https://brilliant.org/wiki/taylor-series-approximation/

Jean Baptiste Joseph Fourier (1768-1830)

had crazy idea (1807):

> *...the manner in which the author arrives at these equations is not exempt of difficulties and...his analysis to integrate them still leaves something to be desired on the score of generality and even rigour.*

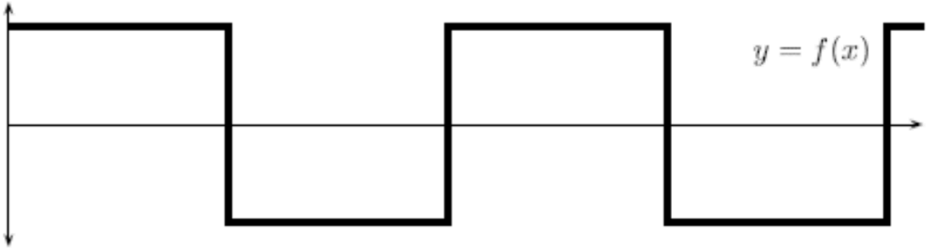
Any *univariate function can b rewritten as a weighted sum sines and cosines of different frequencies.*

- Don’t believe it?
  - Neither did Lagrange, Laplace, Poisson and other big wigs
  - Not translated into English until 1878!
- But it’s (mostly) true!
  - called Fourier Series
  - there are some subtle restrictions

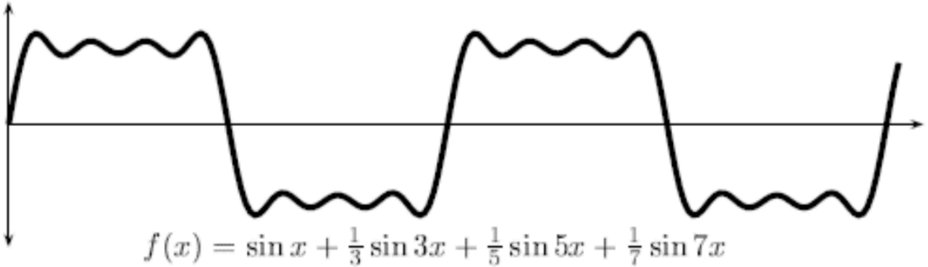
Slide by James Hays

Example

Square wave



Approximation
Using sines



Slide by Emmanuel Agu

Discrete Fourier Transform

$$F(u) = \sum_{x=0}^{N-1} f(x)\left[\cos\left[-2\pi\left(\frac{xu}{N}\right)\right] + i\sin\left[-2\pi\left(\frac{xu}{N}\right)\right]\right]$$

$$F(u) = \sum_{x=0}^{N-1} f(x)\exp\left[-2\pi i\left(\frac{xu}{N}\right)\right]$$

Discrete Fourier Transform

$$F(u) = \sum_{x=0}^{N-1} f(x) \exp\left[-2\pi i \left(\frac{xu}{N}\right)\right]$$

Inverse Discrete Fourier Transform

$$f(x) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) \exp\left[2\pi i \left(\frac{xu}{N}\right)\right]$$

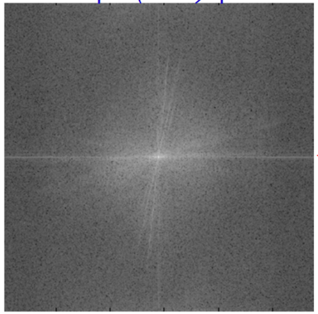
More generally for images (2D DFT and iDFT)

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp\left[-2\pi i \left(\frac{xu}{M} + \frac{yv}{N}\right)\right]$$

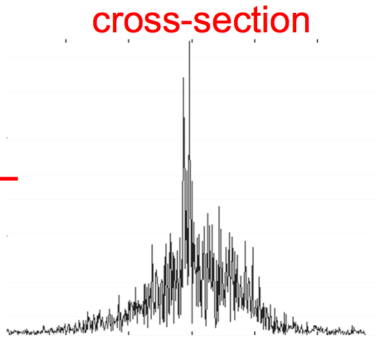
$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp\left[2\pi i \left(\frac{xu}{M} + \frac{yv}{N}\right)\right]$$

Discrete Fourier Transform - Visualization

$f(x,y)$

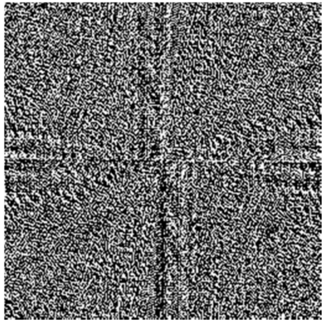


$|F(u,v)|$



- $|f(u,v)|$ generally decreases with higher spatial frequencies
- phase appears less informative

$phase\ F(u,v)$



Slide by A. Zisserman

Fourier Transform

- Fourier transform stores the magnitude and phase at each frequency
  - Magnitude encodes how much signal there is at a particular frequency
  - Phase encodes spatial information (indirectly)
  - For mathematical convenience, this is often notated in terms of real and complex numbers

Amplitude: $A = \pm\sqrt{R(\omega)^2 + I(\omega)^2}$

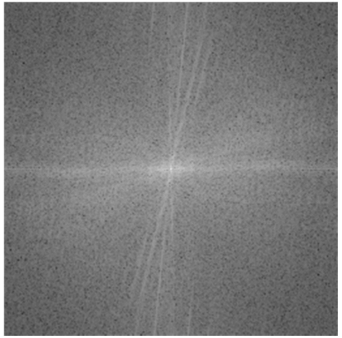
Phase: $\phi = \tan^{-1} \frac{I(\omega)}{R(\omega)}$

Image Filtering in the Frequency Domain

original

$f(x,y)$

$|F(u,v)|$



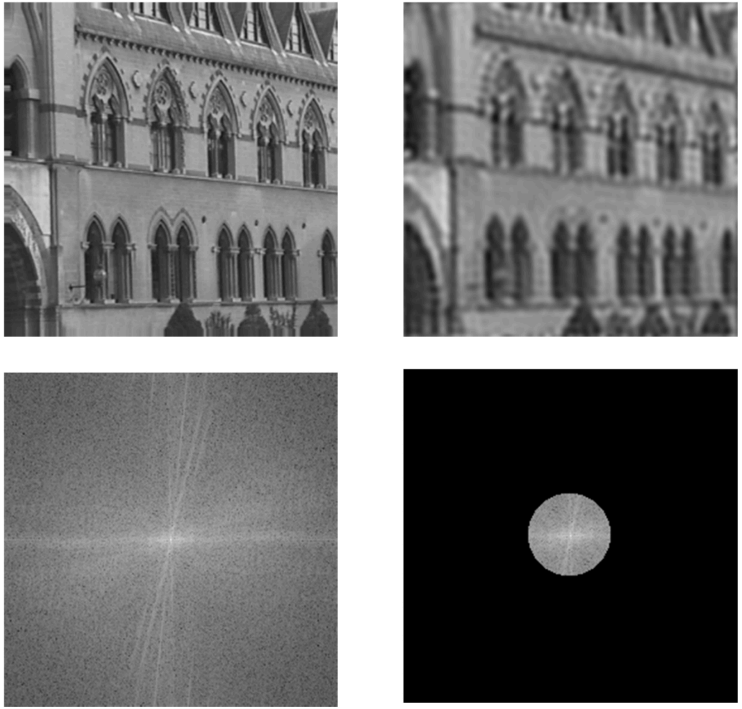
Slide by A. Zisserman

Image Filtering in the Frequency Domain

original | low pass

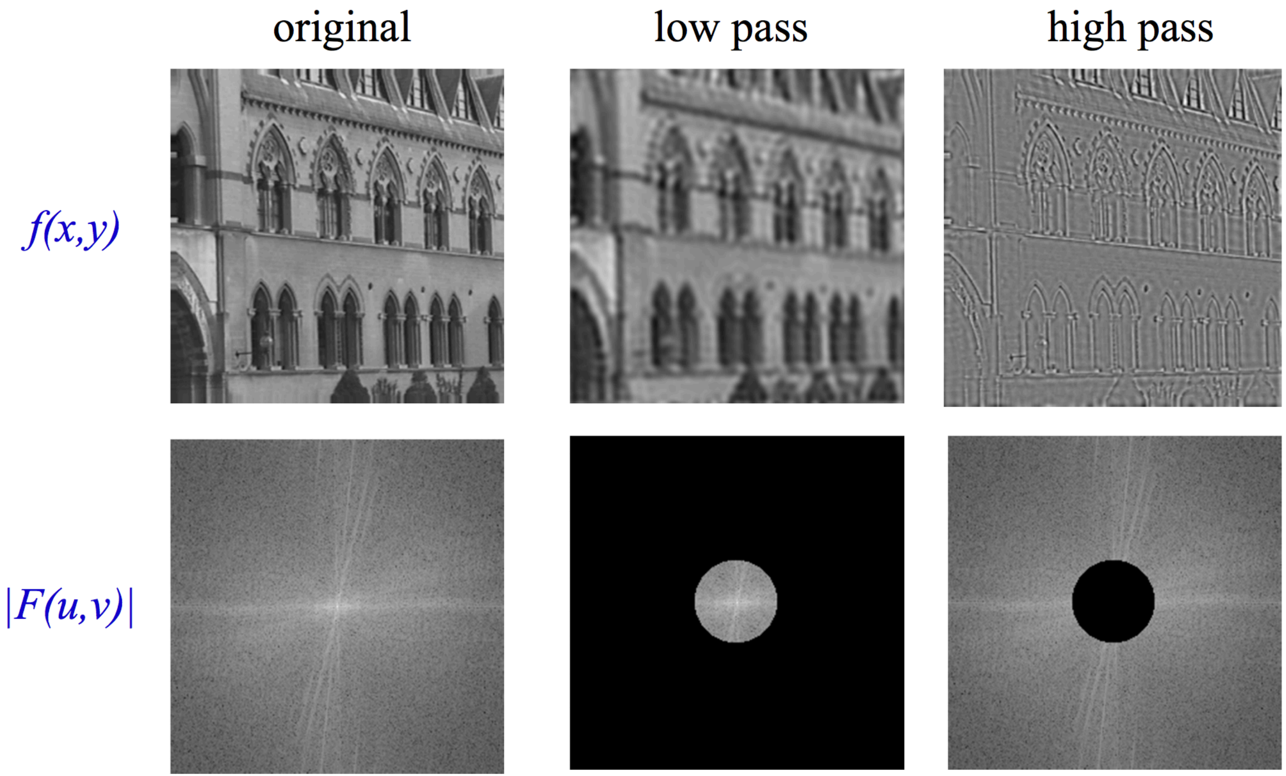
$f(x,y)$

$|F(u,v)|$



Slide by A. Zisserman

Image Filtering in the Frequency Domain



Slide by A. Zisserman

The Convolution Theorem

- The Fourier transform of the convolution of two functions is the product of their Fourier transforms

$$\mathrm{F}[g * h] = \mathrm{F}[g]\,\mathrm{F}[h]$$

- **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!

$$g * h = \mathrm{F}^{-1}[\mathrm{F}[g]\,\mathrm{F}[h]]$$

How can this be useful?

Questions?