Linear Filtering

Chul Min Yeum
Assistant Professor
Civil and Environmental Engineering
University of Waterloo, Canada

CIVE 497 – CIVE 700: Smart Structure Technology

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING

Last updated: 2018-02-06

Automated Brick Counting for Façade Construction Progress Estimation

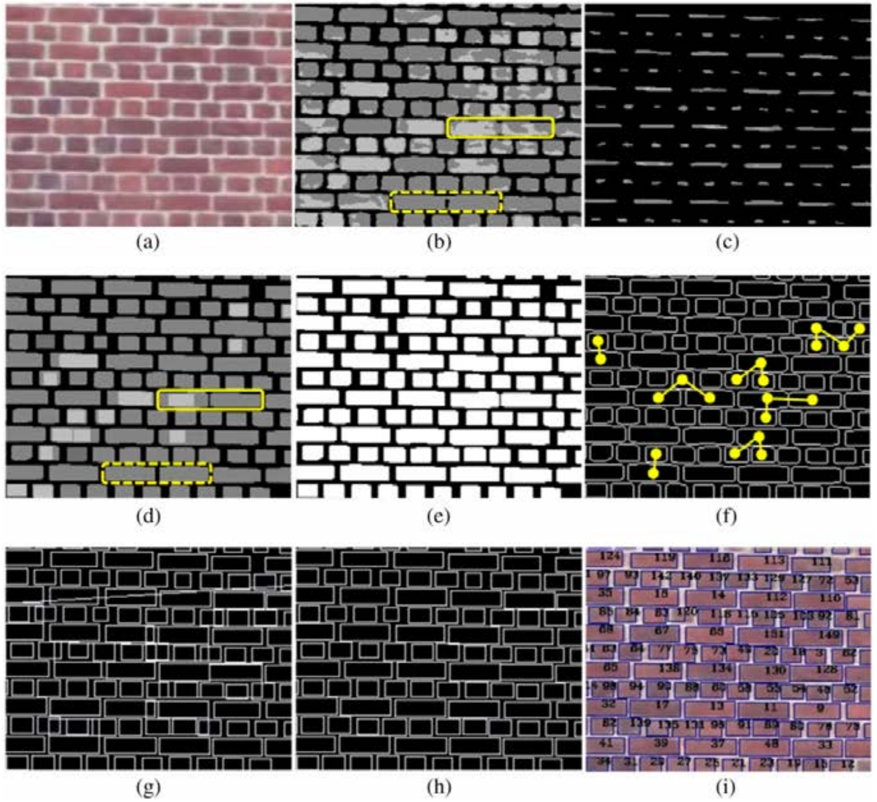
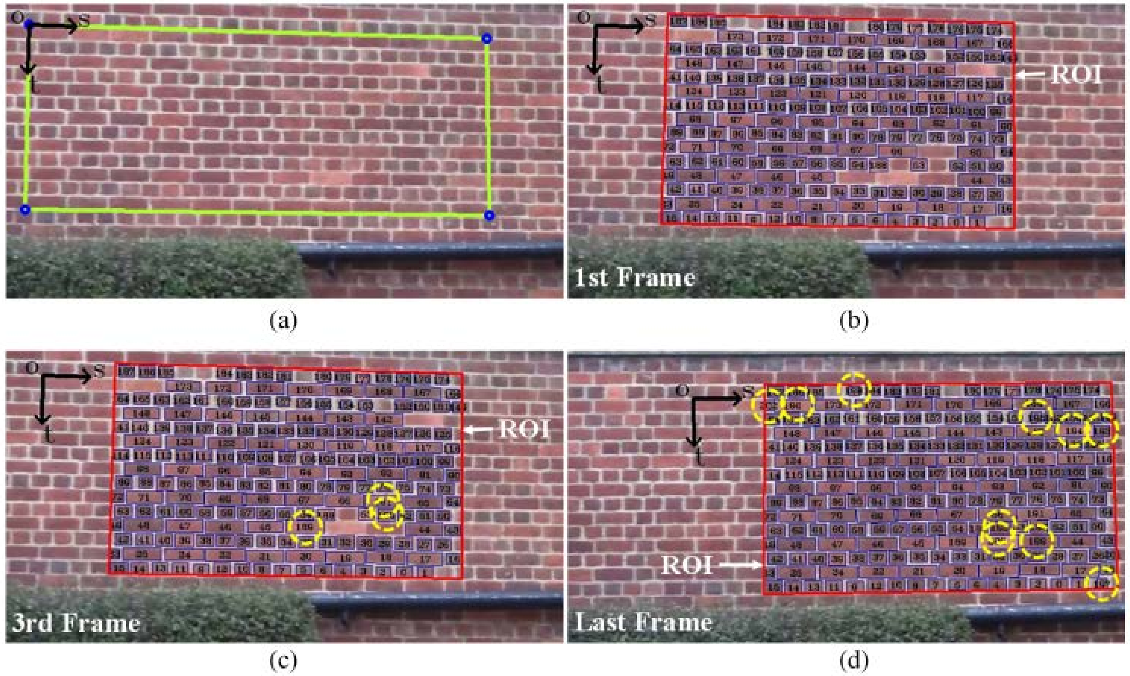


Fig. 3. Brick detection procedures: (a) Gaussian smoothing; (b) color thresholding; (c) erosion; (d) dilation; (e) gray thresholding; (f) Laplace filtering; (g) minimum area rectangle approximation; (h) rectangle size filtering; (i) final result



https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29CP.1943-5487.0000423

How Can We Count the Blocks?

Image as Functions

- We can think of an image as a function, f, from $\mathbb{R}^2 \rightarrow \mathbb{R}$:
  - $f(x, y)$ gives the intensity at position (x, y)
  - Realistically, we expect the image only to be defined over a rectangle, with a finite range:

$$f : [0, w] \times [0, h] \rightarrow [0,1]$$

- A color image is just three functions pasted together.

$$f(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix}$$

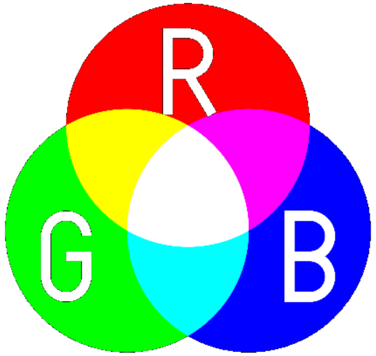
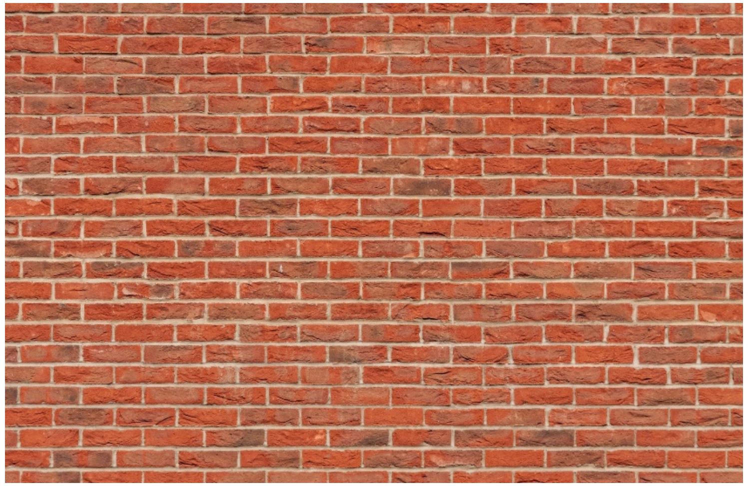
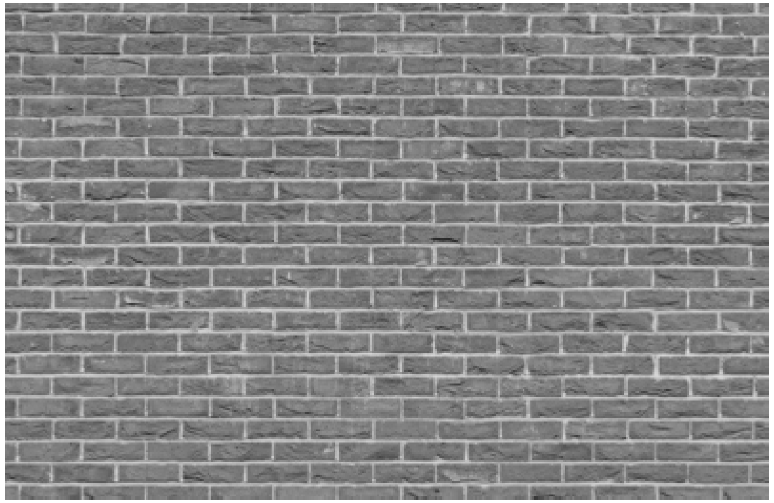


Image as Functions (Continue)

RGB



Gray



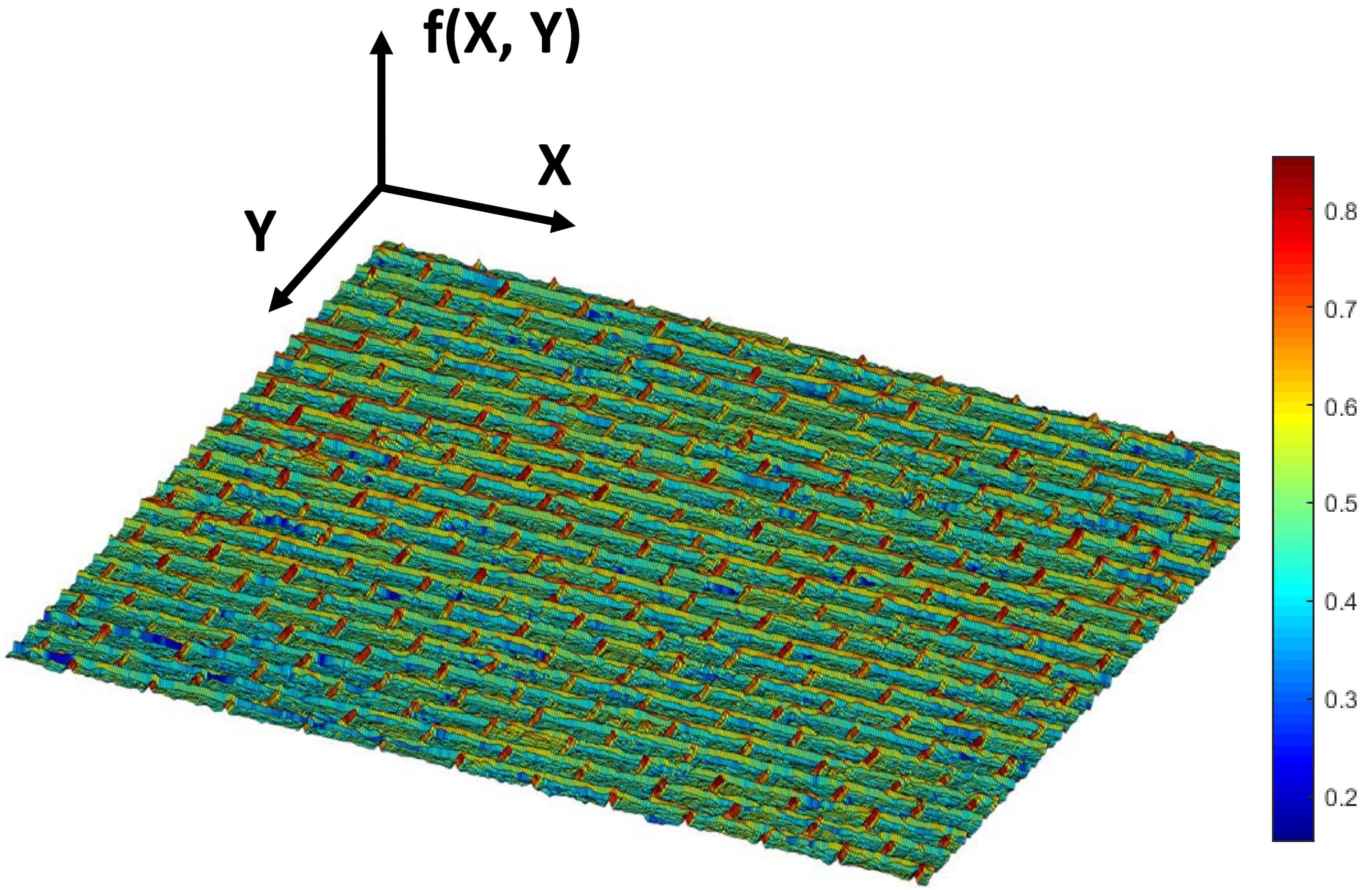


Image Interpreted in a Frequency Domain

Original image | Power spectrum with mask that filters low frequencies | Result of inverse transform | Power spectrum with mask that passes low frequencies | Result of inverse transform

Convolution!!

https://imagej.nih.gov/ij/docs/images/fft.jpg

Linear Filtering

- One simple version : linear filtering (cross-correlation or convolution)

- Replace each pixel by a linear combination of its neighbors

- The prescription for the linear combination is called the “kernel” (or “mask” or “filtering”)

1	1	1
1	1	1
1	1	1

Local image data

1	1	1
1	1	1
1	1	1

Kernal

0	0	0
0	9	0
0	0	0

Modified image data

Cross-Correlation

Let F be the image, H be the kernel (of size 2k+1 x 2k+1),

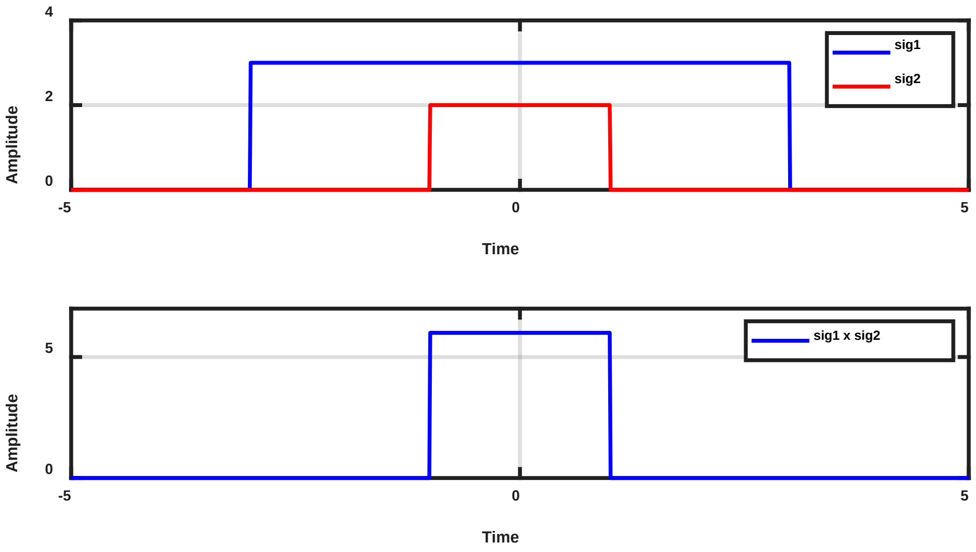
G be the output image

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u,j+v]$$

This is called a cross-correlation operation:

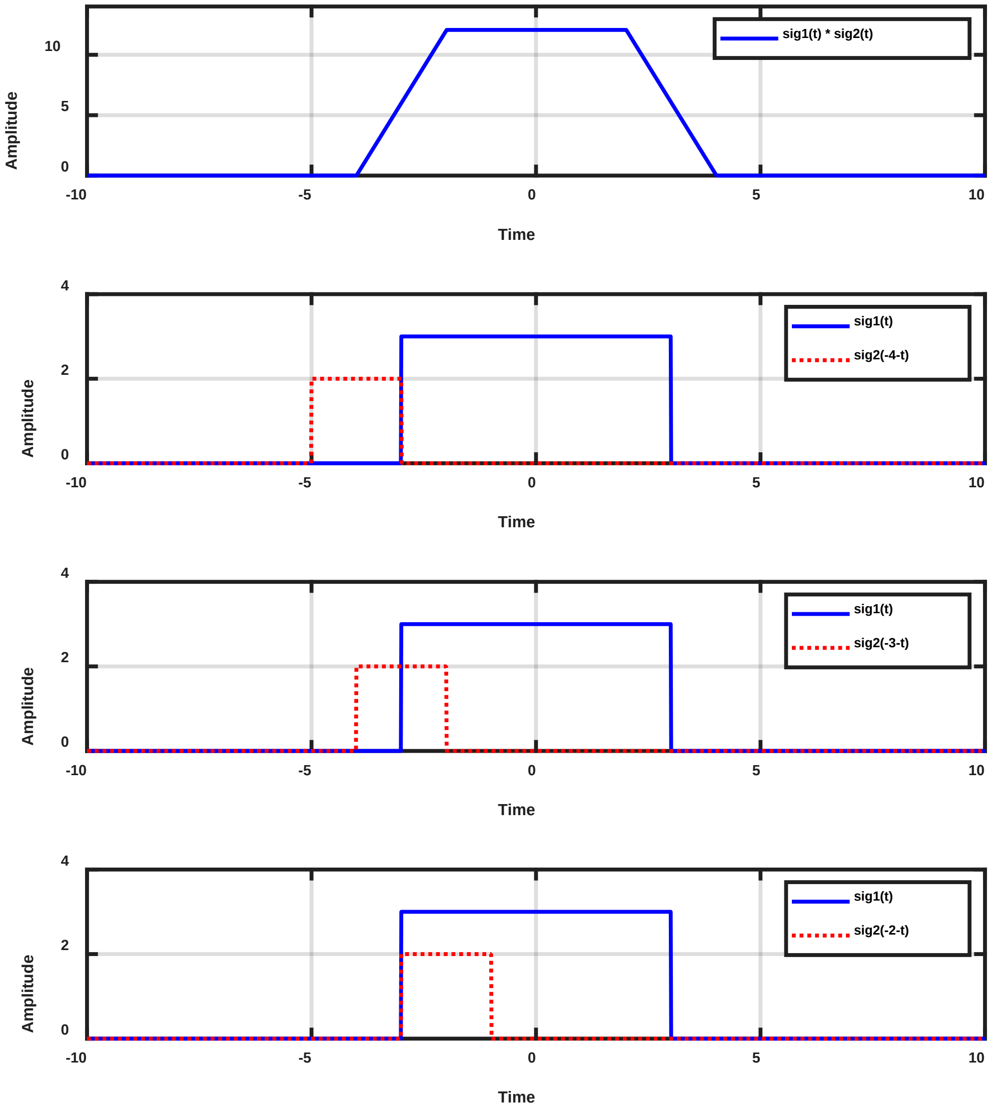
$$G = H \otimes F$$

Example: Signal Processing



$$h(t)x(t)$$

$$h(t) * x(t)) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$$



Cross-Correlation (Continue)

Cross-correlation is a measure of similarity between two images. Cross correlation with a kernel can be viewed as comparing a litter “picture” or “region” of what you want to find across all sub-regions in the image.

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u,j+v]$$

0	0	0	1	0	0	0
0	1	1	1	0	0	0
0	1	1	1	0	1	1
0	1	1	1	1	0	0
0	0	0	0	1	0	0
0	1	0	0	0	0	0
0	0	0	0	1	1	1

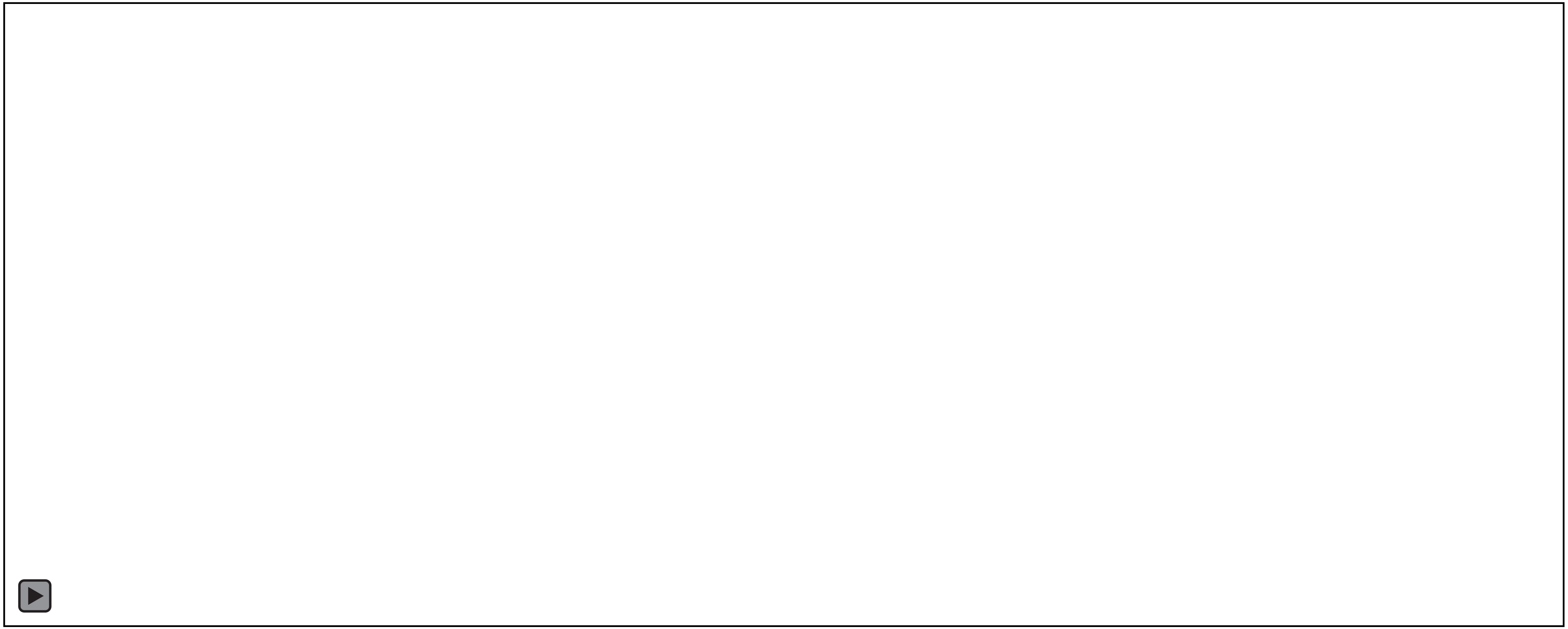
$\otimes$

1	1	1
1	1	1
1	1	1

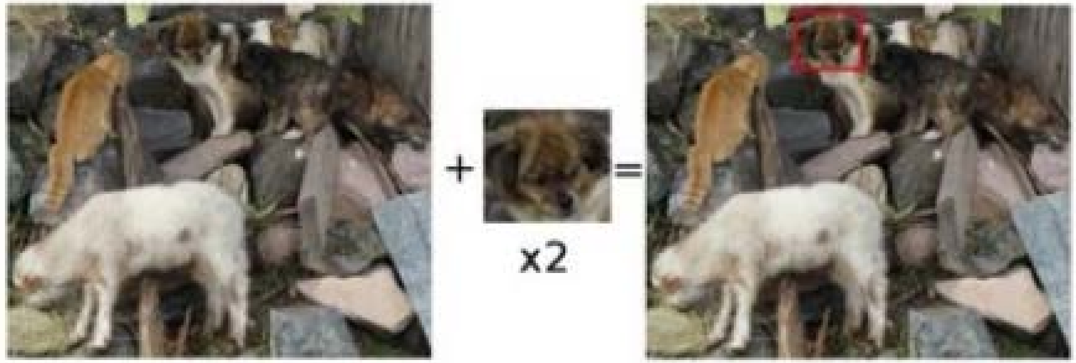
$=$

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

Computation of the Cross-Correlation Between Two Matrices



Example: Finding Same Objects on Images



https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html

Convolution

Let F be the image, H be the kernel (of size 2k+1 x 2k+1),

G be the output image

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v] \boxed{F[i-u,j-v]}$$

Only difference is that the kernel is "flipped" horizontally and vertically.

This is called a convolution operation:

$$G = H * F$$

Property: Commutative, Associative, and Linearity

- Commutative: $F * H = H * F$
- Associative: $F * (H * L) = (H * F) * L$
- Linearity: $F * (H_1 + H_2) = F * H_1 + F * H_2$
- Relationship with differentiation: $(F * H)' = F' * H = F * H'$

Example: Signal Filtering

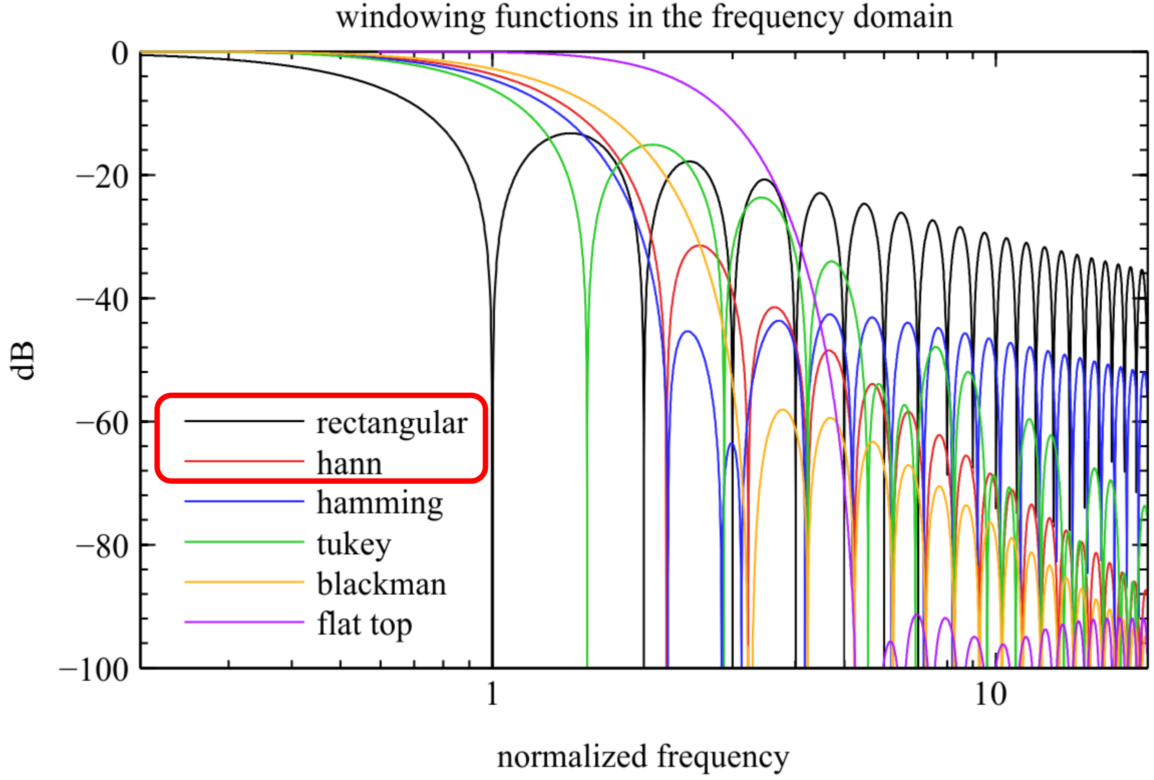
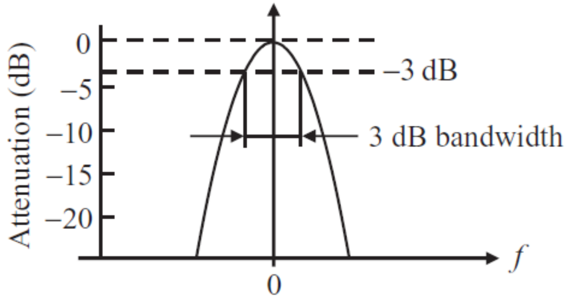


Table 4.2 Properties of some window functions

Window (length T)	Highest side lobe (dB)	Asymptotic roll-off (dB/octave)	3 dB bandwidth	Noise bandwidth	First zero crossing (freq.)
Rectangular	−13.3	6	$0.89\frac{1}{T}$	$1.00\frac{1}{T}$	$\frac{1}{T}$
Bartlett (triangle)	−26.5	12	$1.28\frac{1}{T}$	$1.33\frac{1}{T}$	$\frac{2}{T}$
Hann(ing) (Tukey or cosine squared)	−31.5	18	$1.44\frac{1}{T}$	$1.50\frac{1}{T}$	$\frac{2}{T}$
Hamming	−43	6	$1.30\frac{1}{T}$	$1.36\frac{1}{T}$	$\frac{2}{T}$
Parzen	−53	24	$1.82\frac{1}{T}$	$1.92\frac{1}{T}$	$\frac{4}{T}$



Difference between Cross-Correlation and Convolution

X

A	B	C
D	E	F
G	H	I

Y

I	H	G
F	E	D
C	B	A

$$G = H \otimes X = H * Y$$

$\otimes$ cross-correlation

$\otimes$ convolution

The basic difference between convolution and cross-correlation is that the convolution process rotates the kernel by 180 degrees and conduct cross-correlation.

Usage

Cross-Correlation: Process to measure a similarity between two signals.

Convolution: Process to transform a signal to another signal.

Filters

0	0	0
0	1	0
0	0	0

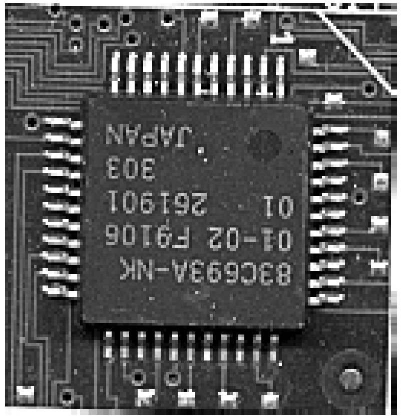
0	0	0
0	0	1
0	0	0

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

-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

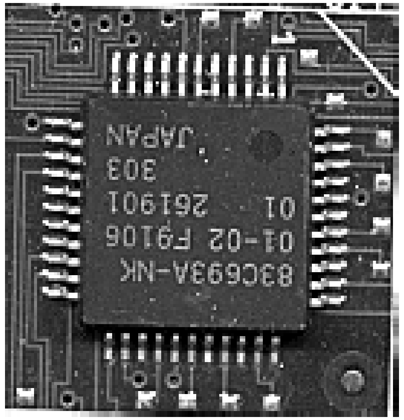
Linear Filter: Generating an Identical Image

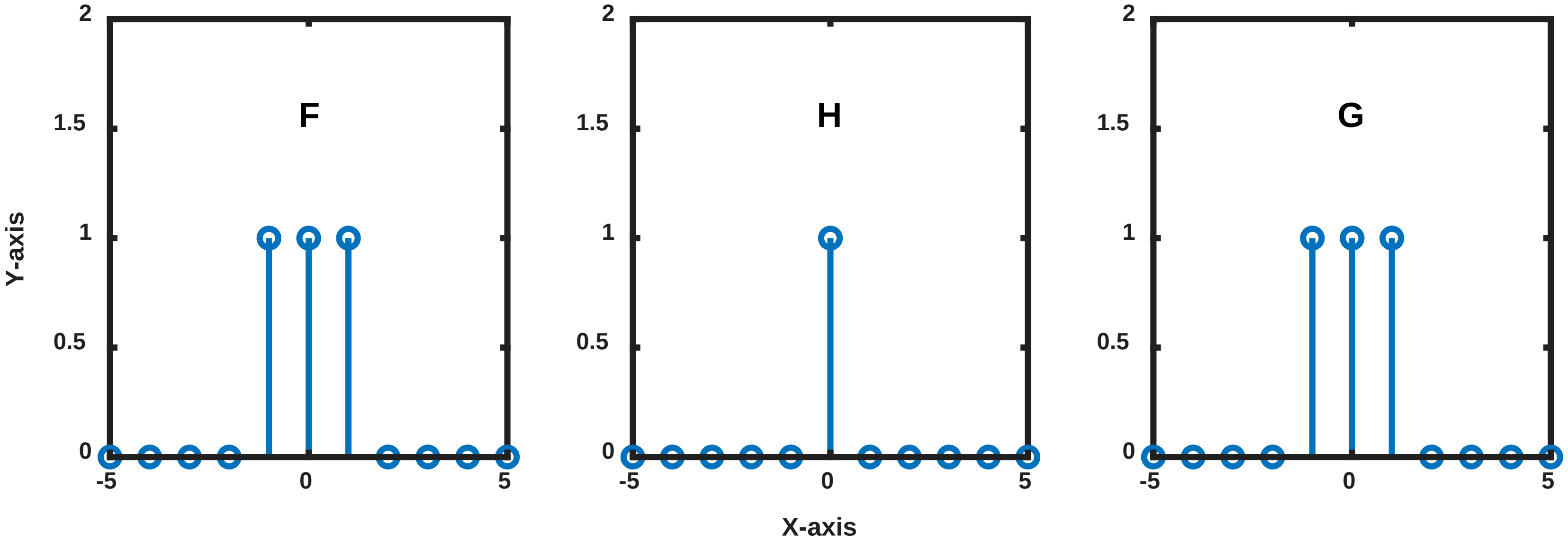


$*$

0	0	0
0	1	0
0	0	0

$=$





$$F * H = G$$

Filters

0	0	0
0	1	0
0	0	0

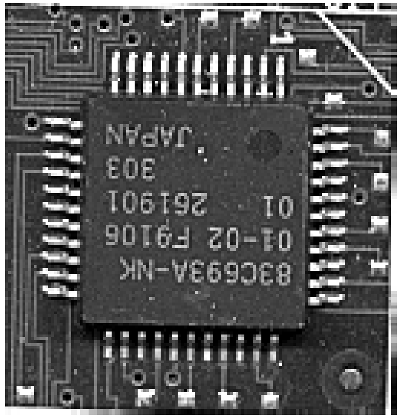
0	0	0
0	0	1
0	0	0

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

-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

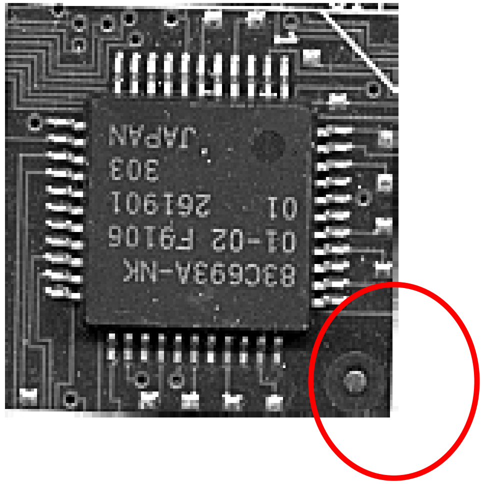
Linear Filter: Shifting Pixels

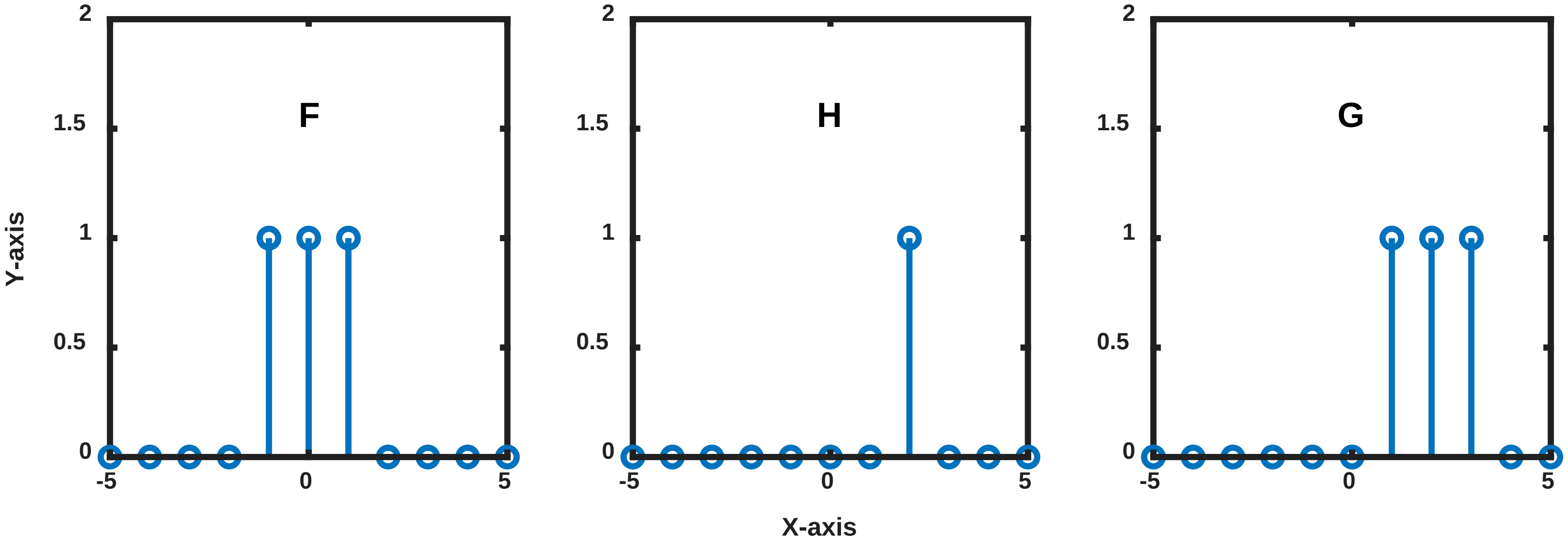


$*$

0	0	0
0	0	1
0	0	0

$=$





$$F * H = G$$

Filters

0	0	0
0	1	0
0	0	0

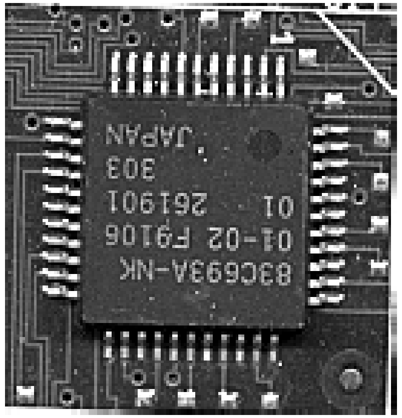
0	0	0
0	0	1
0	0	0

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

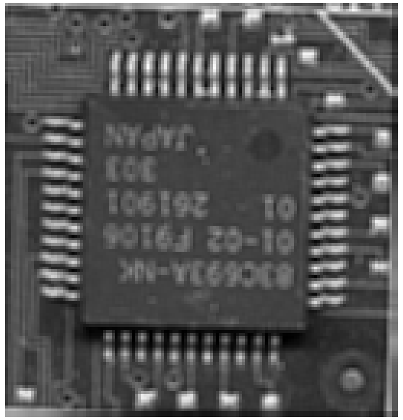
-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

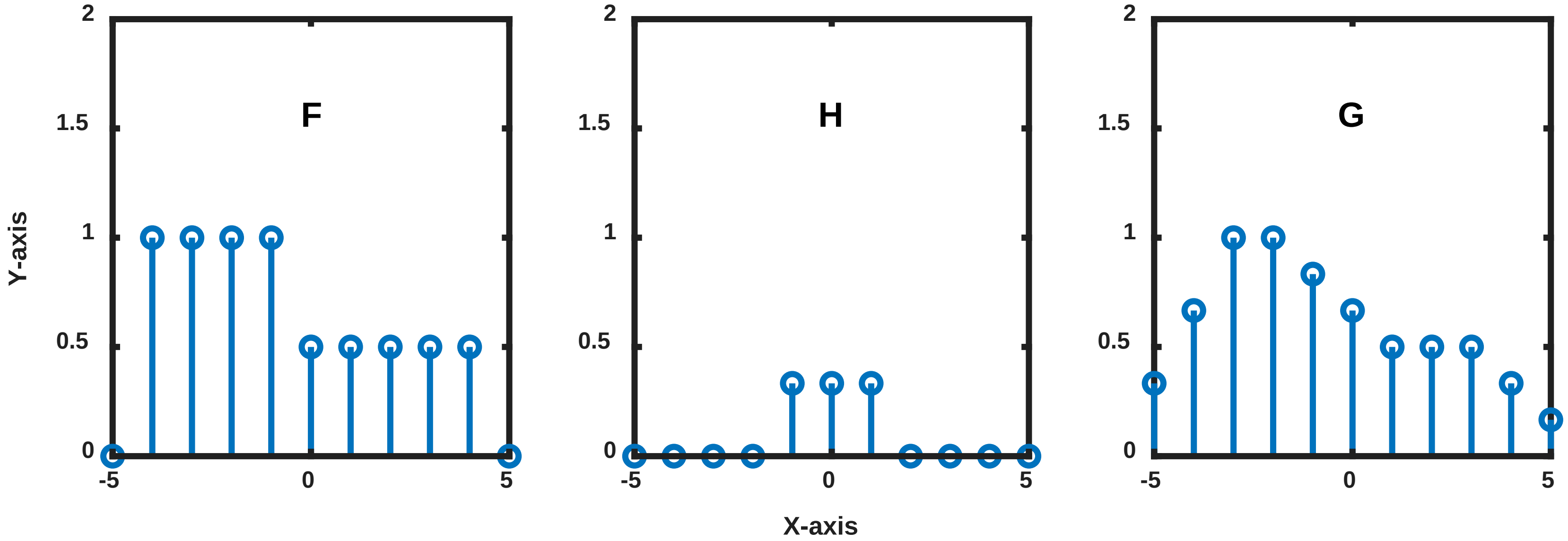
0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

Linear Filter: Blurring



$$* \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad =$$





$$F * H = G$$

Filters

0	0	0
0	1	0
0	0	0

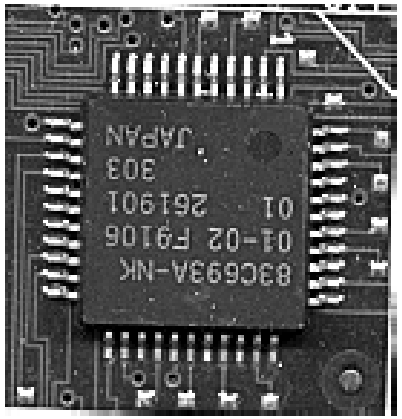
0	0	0
0	0	1
0	0	0

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

-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

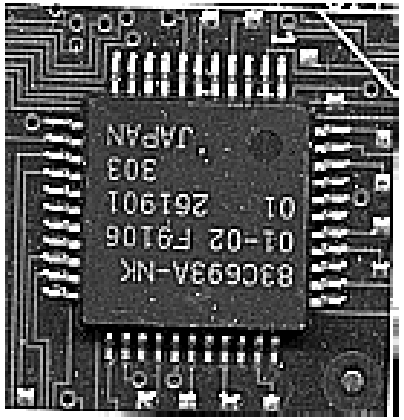
Linear Filter: Sharpening

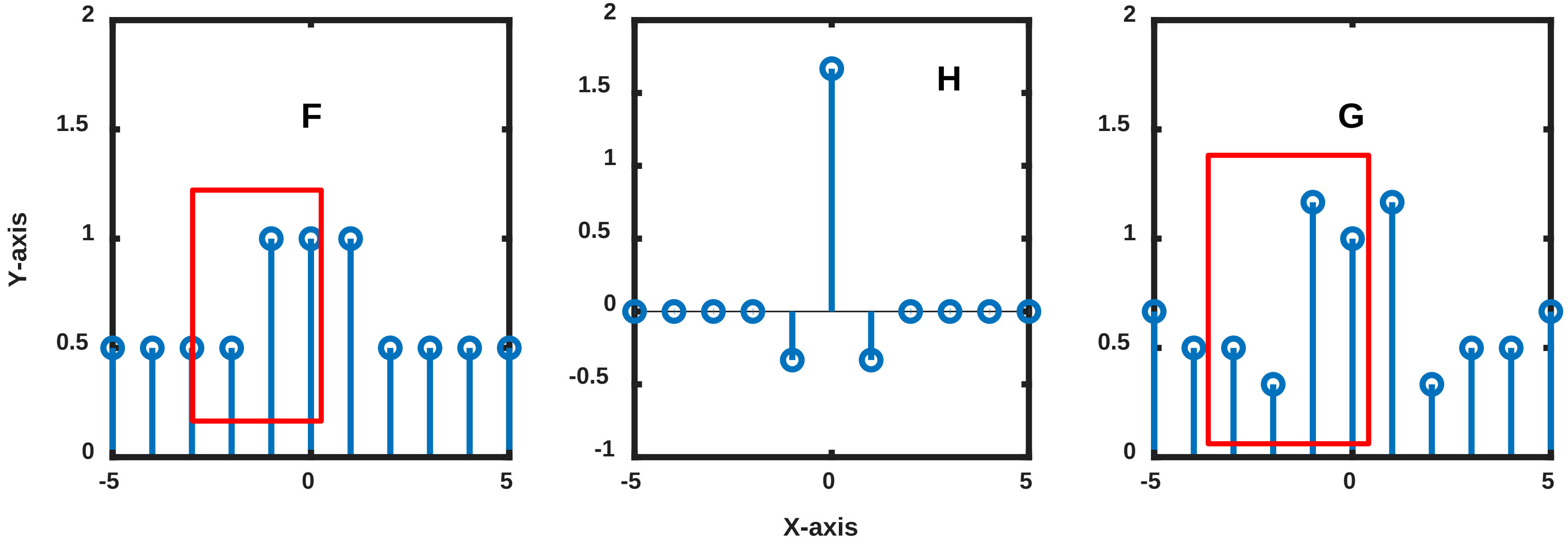


$*$

-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

$=$





$$F * H = G$$

How the Sharpening Filter Works

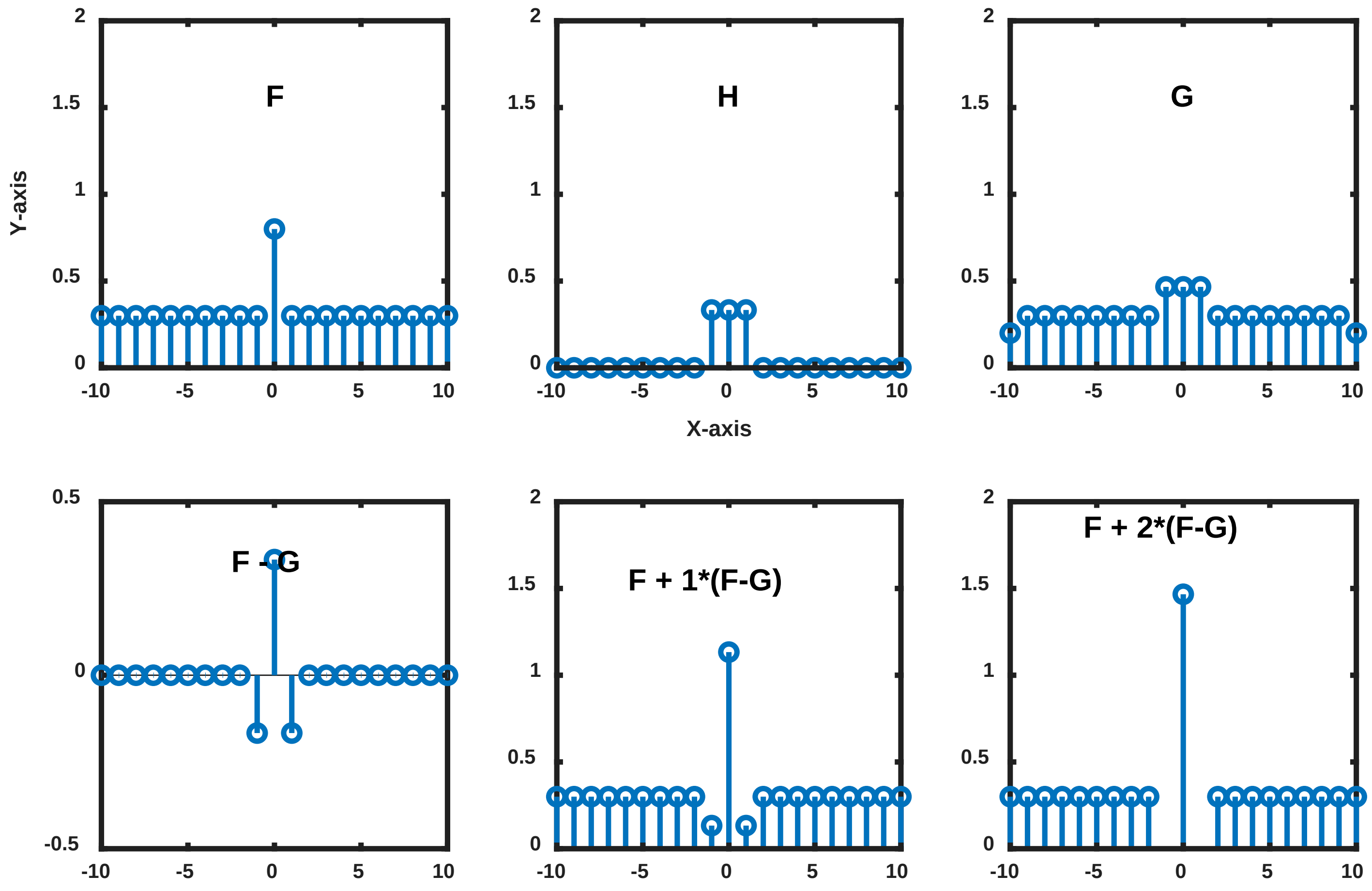
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \end{bmatrix} = \begin{bmatrix} -0.11 & -0.11 & -0.11 \\ -0.11 & 1.89 & -0.11 \\ -0.11 & -0.11 & -0.11 \end{bmatrix}$$

$$F + (F - F * H) = G$$

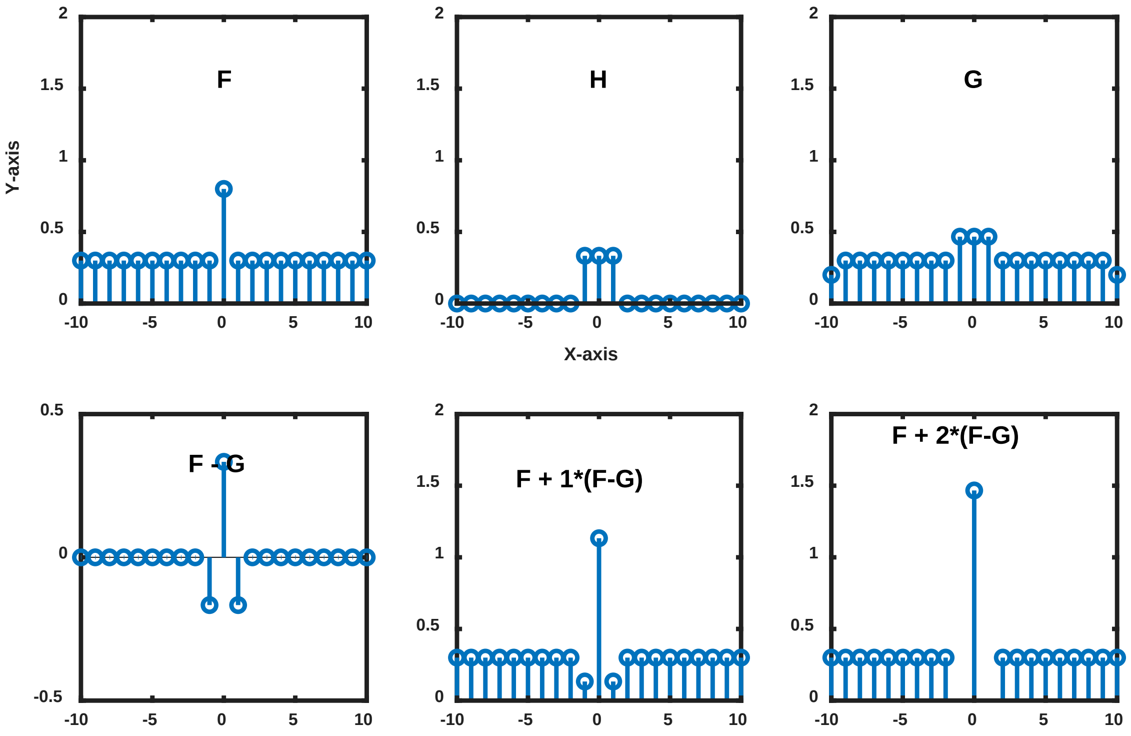
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \alpha+1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \alpha \begin{bmatrix} 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \end{bmatrix} = \begin{bmatrix} -0.11\alpha & -0.11\alpha & -0.11\alpha \\ -0.11\alpha & 1+0.89\alpha & -0.11\alpha \\ -0.11\alpha & -0.11\alpha & -0.11\alpha \end{bmatrix}$$

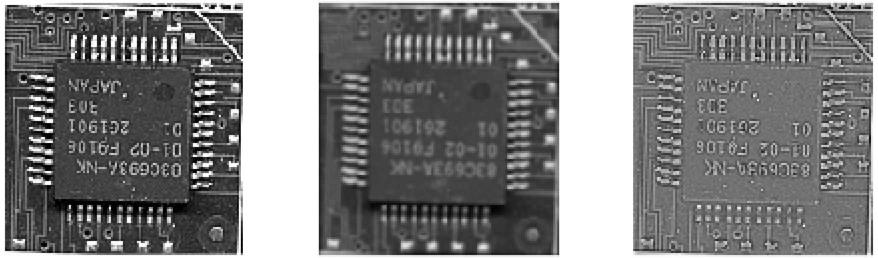
$$F + \alpha(F - F * H) = G$$

How the Sharpening Filter Works (Signal Example) (Continue)

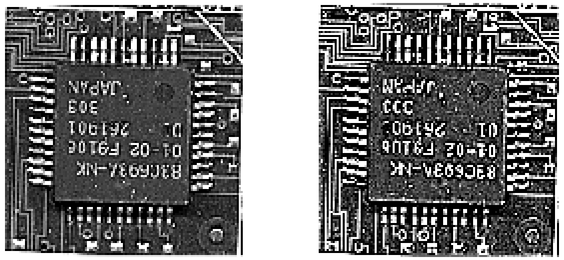


How the Sharpening Filter Works (Image Example)





F G $F - G$



$F + (F - G)$

$F + 5(F - G)$

Filters

0	0	0
0	1	0
0	0	0

0	0	0
0	0	1
0	0	0

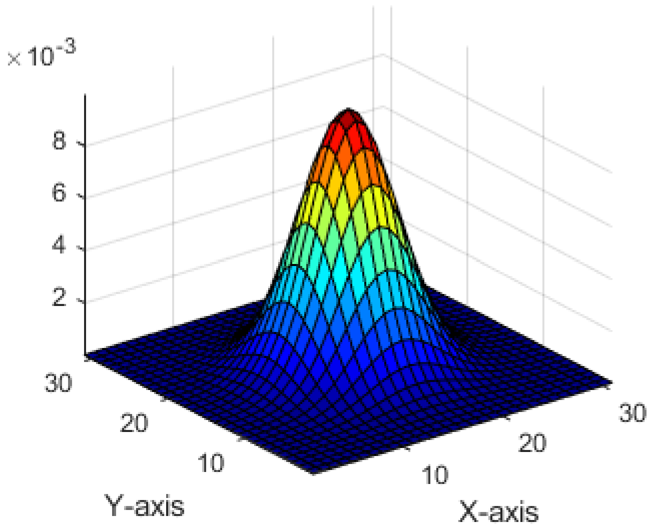
$\frac{1}{9}$

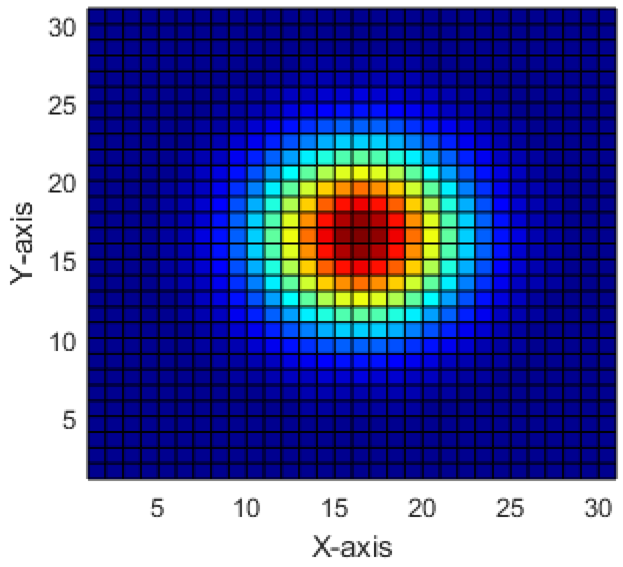
1	1	1
1	1	1
1	1	1

-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

Gaussian Kernel





$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

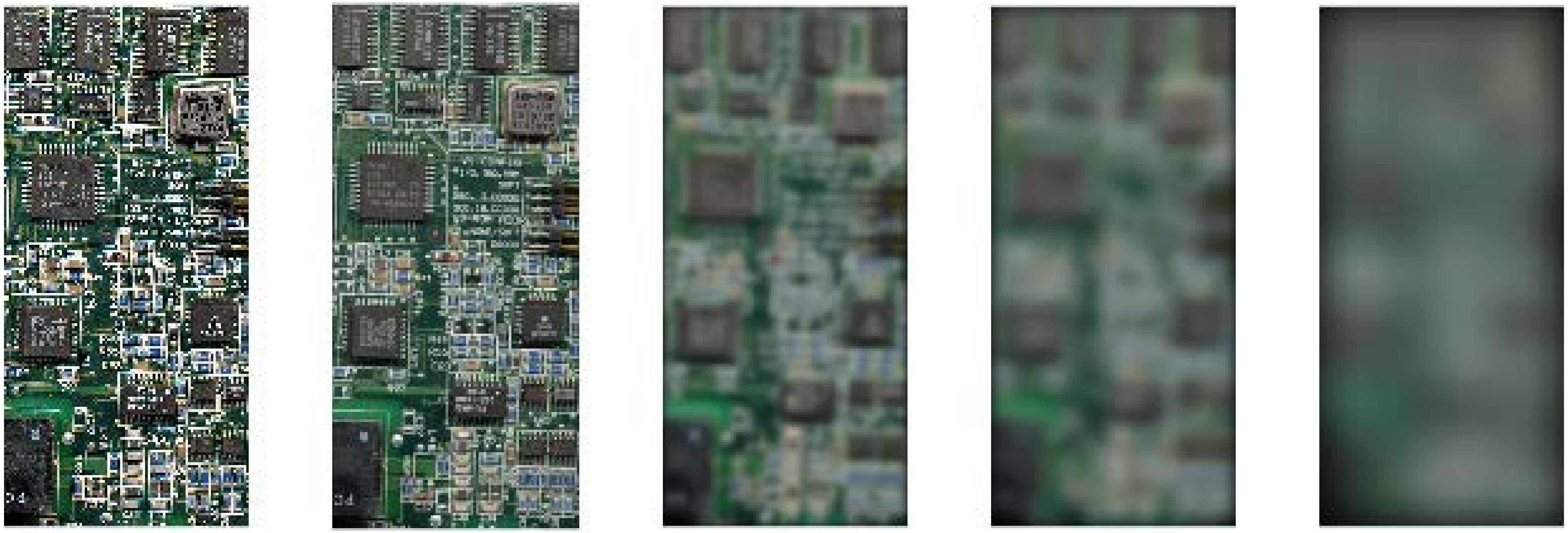
Gaussian Filter

```
clear; close all; clc;

h = fspecial('gaussian',3,1);

numSize = 3;
[x, y] = meshgrid(1:numSize);
x = x-(round(numSize/2));
y = y-(round(numSize/2));
sigma = 1;

G_sigma = 1/(2*pi*sigma^2)*exp(-(x.^2 + y.^2)/(2*sigma^2));
G_sigma = G_sigma/sum(G_sigma,'all');

figure(1);
subplot(121); PlotMat(h,gca,'float');
subplot(122); PlotMat(G_sigma,gca,'float');

fig2 = figure(2);
subplot(121); h = fspecial('gaussian', 31,4);
surf(h); axis tight; colormap(jet)
set(fig2,'Position', [100 100 800 300]);
xlabel('X-axis'); ylabel('Y-axis');
subplot(122); surf(h); view(0,90);
xlabel('X-axis'); ylabel('Y-axis'); axis tight
```

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

Effect of Gaussian Window Sizes



$f1$ $f2$ $f3$ $f4$

```
f1 =  fspecial('gaussian', 101,1);
f2 =  fspecial('gaussian', 101,5);
f3 =  fspecial('gaussian', 101,10);
f4 =  fspecial('gaussian', 101,30);
```

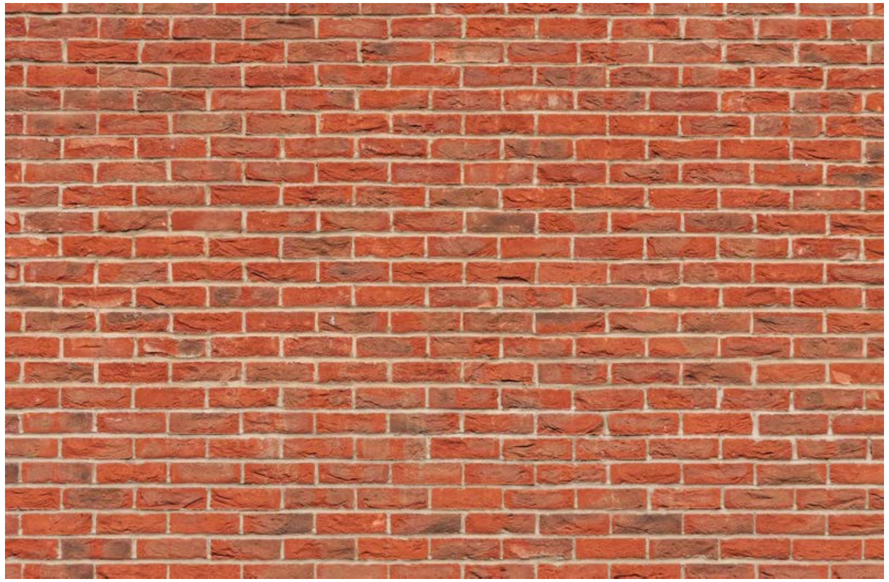
Difference of the Between a Gaussian Filter and Box Filter

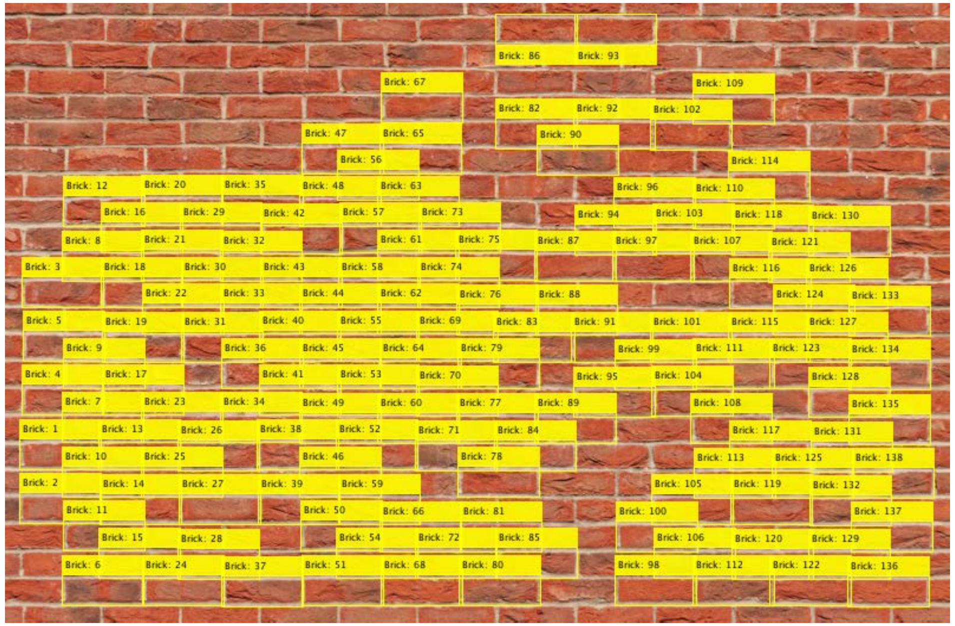
```
f1 = fspecial('gaussian', 19, 3);
f2 = fspecial('average', 19);

figure(3);
subplot(121); imshow(imfilter(img,f1));
subplot(122); imshow(imfilter(img,f2));
```

https://dsp.stackexchange.com/questions/208/what-should-be-considered-when-selecting-a-windowing-function-when-smoothing-a-t

https://stackoverflow.com/questions/31131672/difference-between-mean-and-gaussian-filter-in-result

Template Matching



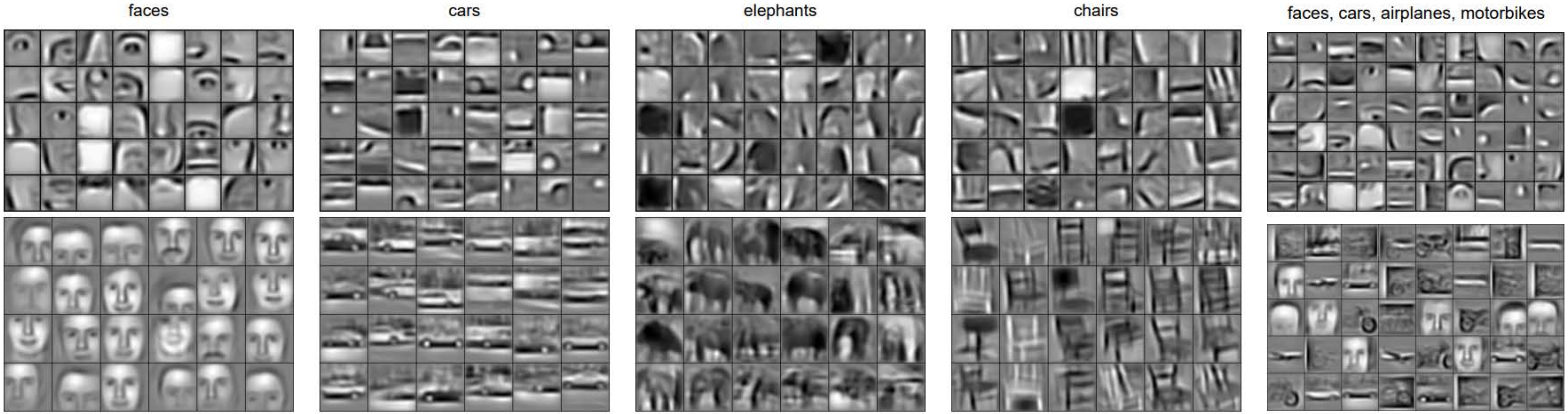


Count_brick_wall.m

Challenges (template matching) !!!!!!

Preview: Convolution Neural Network

Preview: Convolution Neural Network (Continue)



http://web.eecs.umich.edu/~honglak/icml09-ConvolutionalDeepBeliefNetworks.pdf

Slide Credits and References

- Lecture notes: Rob Fergus.
- Lecture notes: Steve Seitz
- Lecture notes: Mohammad Jahanshahi
- Lecture notes: Svetlana Lazebnik
- Lecture notes: Derek Hoiem
- Lecture notes: Ioannis (Yannis) Gkioulekas
- Lecture notes: Robert Collins
- Lecture notes: Jason Corso
- Lecture notes: Gordon Wetzstein
- Lecture notes: Noah Snavely