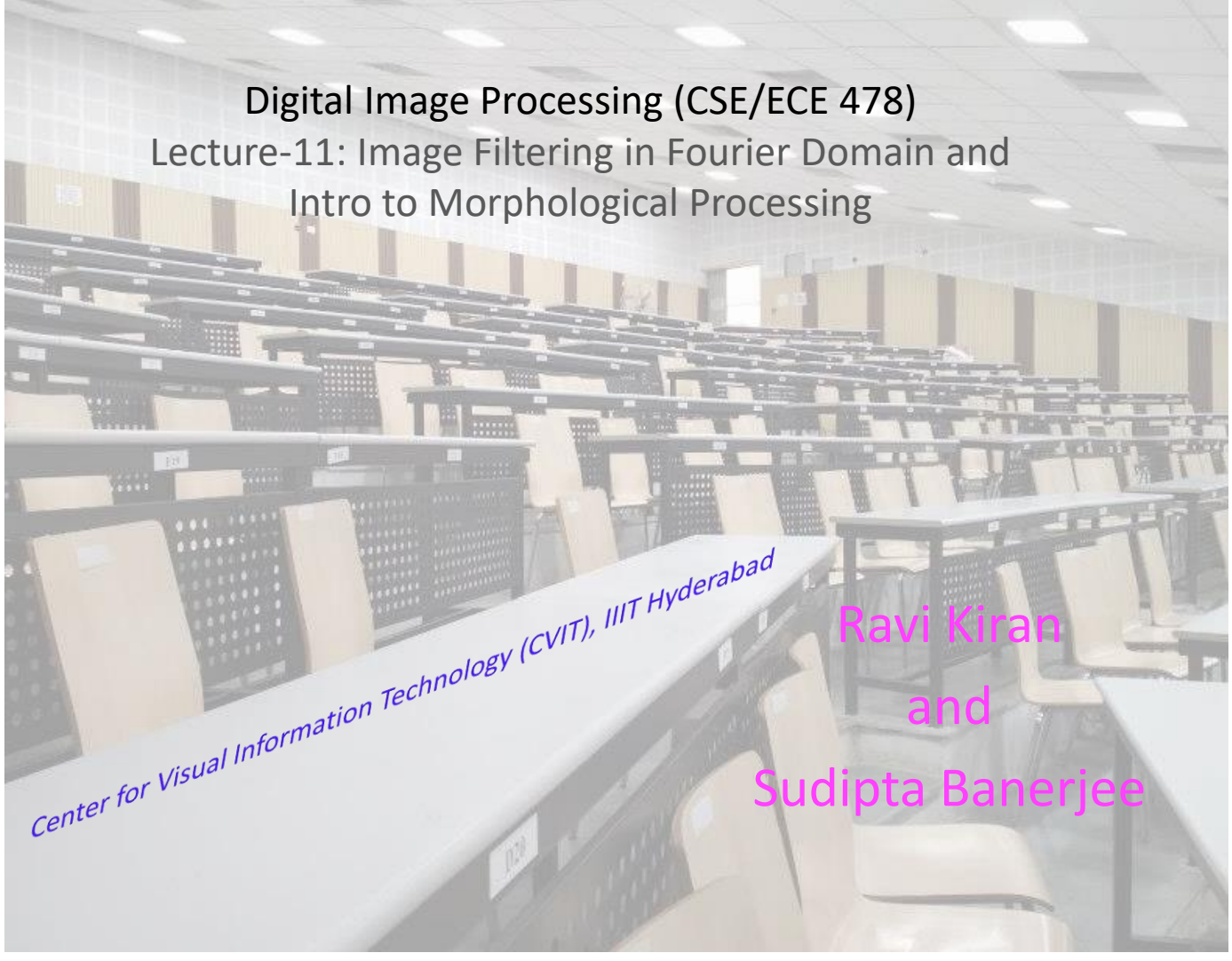
Digital Image Processing (CSE/ECE 478)

Lecture-11: Image Filtering in Fourier Domain and Intro to Morphological Processing



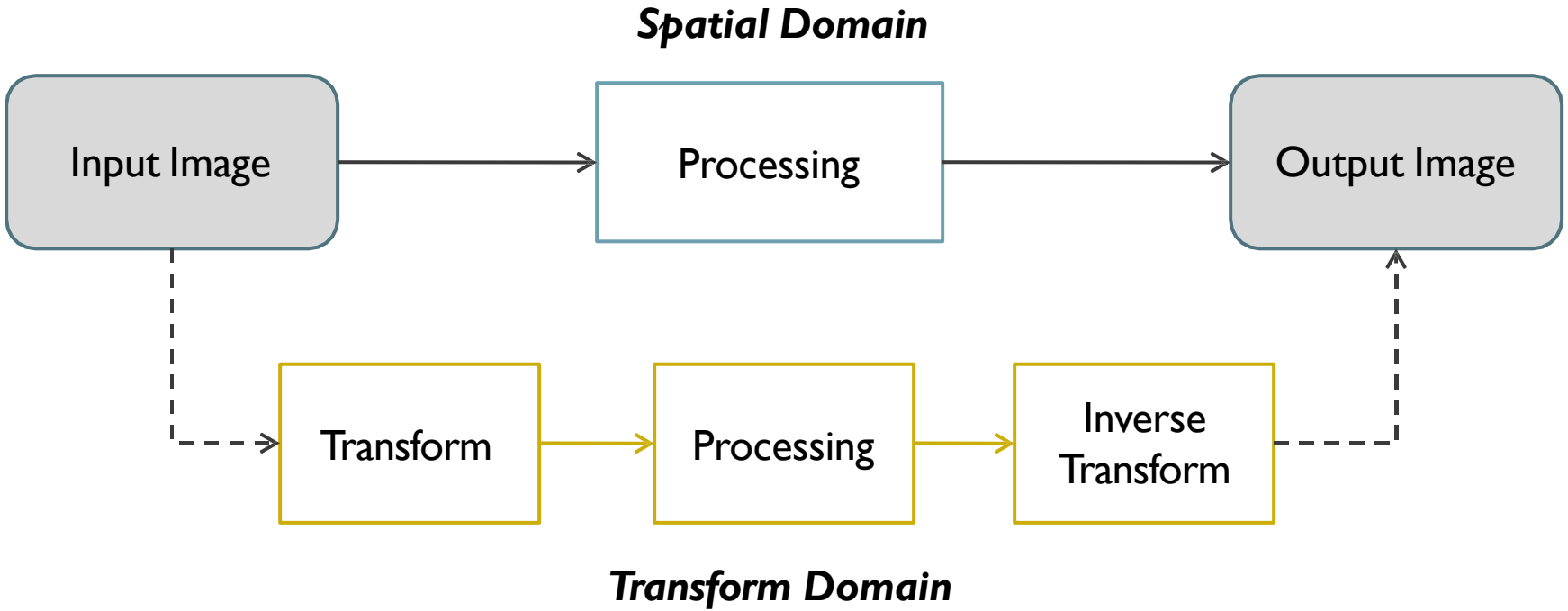
Center for Visual Information Technology (CVIT), IIIT Hyderabad

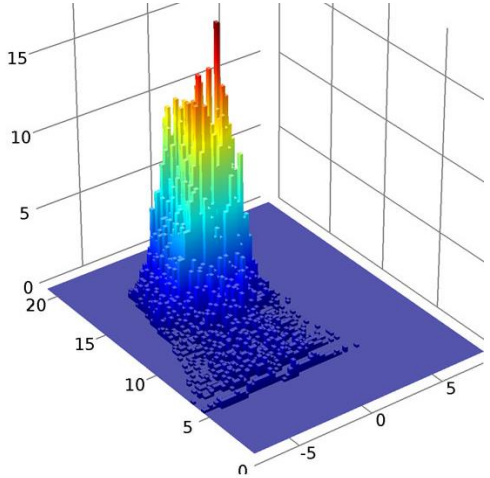
Ravi Kiran

and

Sudipta Banerjee

Spatial vs. Transform Domain Processing





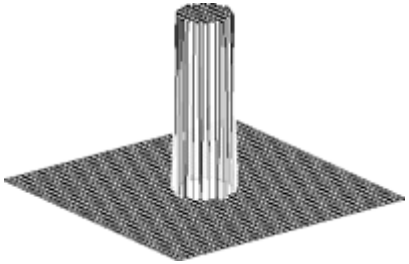
$$F[m,n] = \sum_{y=0}^{y=(N-1)} \sum_{x=0}^{x=(M-1)} f[x,y]\, e^{-2\pi j\left(\frac{mx}{M}+\frac{ny}{N}\right)}$$

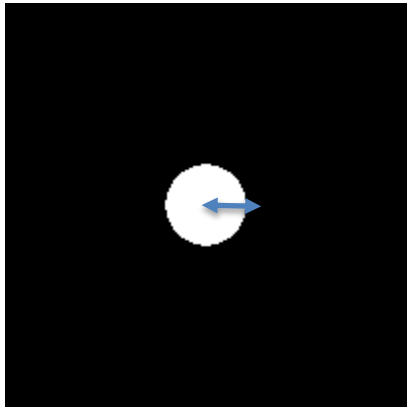
$$f[x,y] = \frac{1}{MN} \sum_{n=0}^{n=(N-1)} \sum_{m=0}^{m=(M-1)} F[m,n]\, e^{2\pi j\left(\frac{mx}{M}+\frac{ny}{N}\right)}$$

Image Enhancement and Filtering in Frequency Domain

Ideal Low Pass Filters

$$H(u,v) = \begin{cases} 1 \text{ if } D(u,v) \le D_0 \\ 0 \text{ if } D(u,v) > D_0 \end{cases}$$



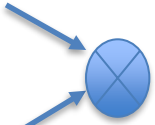


where $D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$

$D_0 \rightarrow$ cut off frequency

$$I(x,y) \xrightarrow{DFT} G[u,v]$$

$$h(x,y) \xrightarrow{DFT} H[u,v]$$



$$G[u,v]H[u,v] \xrightarrow{IDFT} I_{LPF}(x,y)$$

$$I_{LPF}(x,y) = I(x,y) * h(x,y)$$

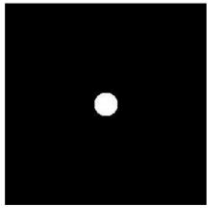
Convolution Theorem

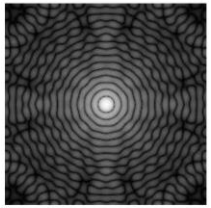
Image courtesy: Gonzalez and Woods

Ideal Low Pass Filters

- Low pass filtered image is inverse Fourier Transform of product of G and H

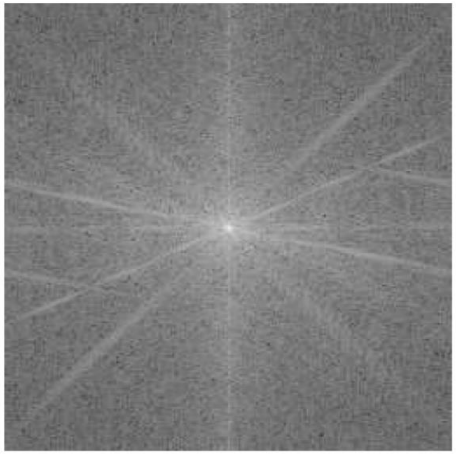
$$\mathcal{F}^{-1}(G.H)$$


Circle


DFT

- Example: Consider the following image and its DFT


Image


DFT

Ideal Low Pass Filters

Applying
low pass filter
to DFT
Cutoff D = 15

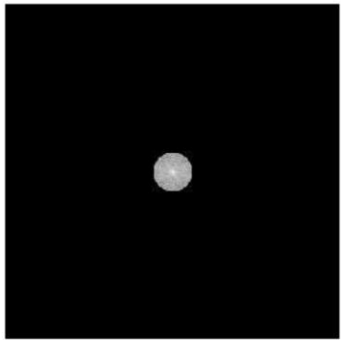
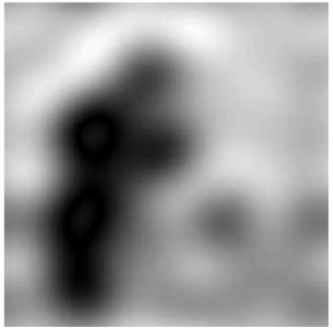


Image after
inversion

low pass filter
Cutoff D = 5

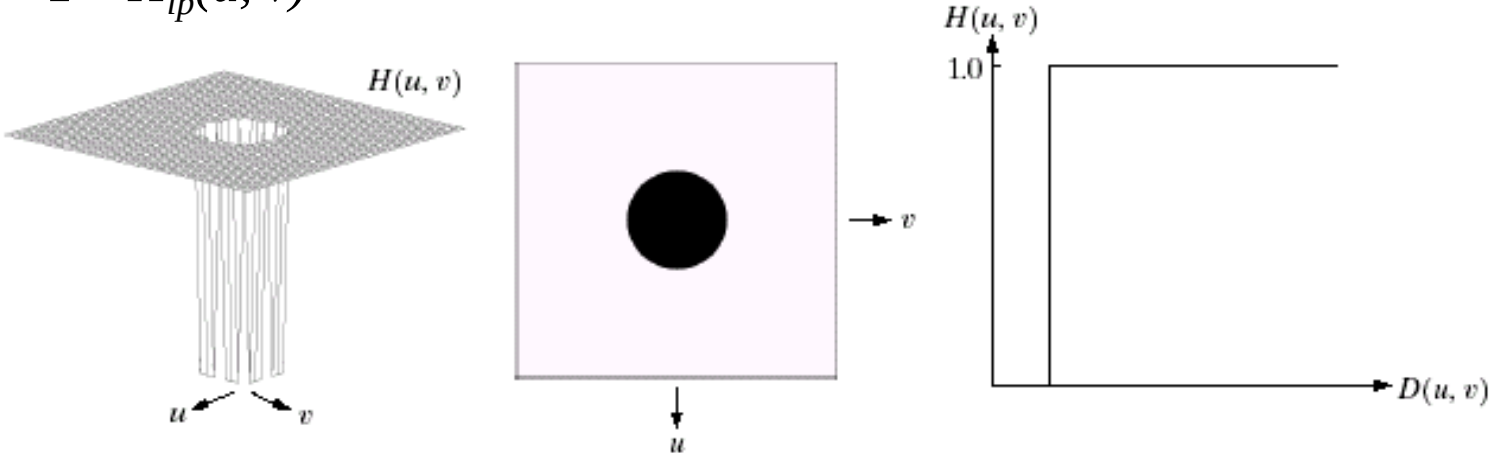


low pass filter
Cutoff D = 30

Note: Sharp filter
Cutoff causes
ringing

Ideal High Pass Filters

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

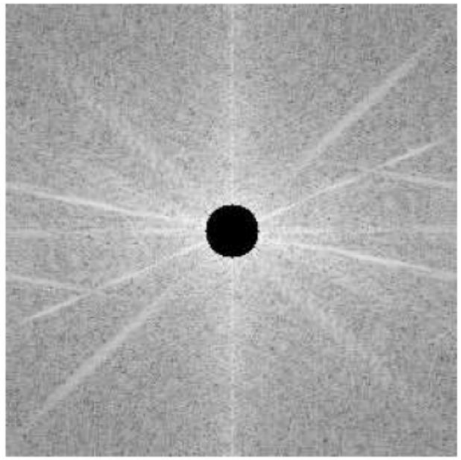


$$H(u,v) = \begin{cases} 0 \text{ if } D(u,v) \le D_0 \\ 1 \text{ if } D(u,v) > D_0 \end{cases}$$

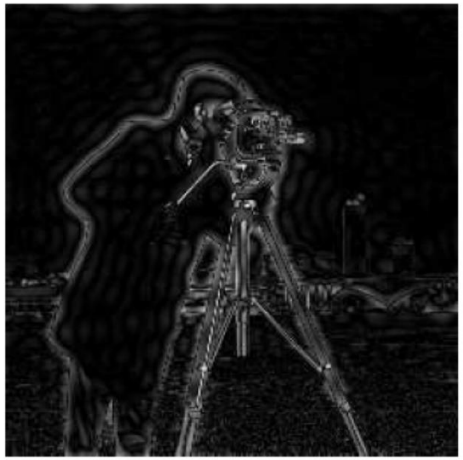
Image courtesy: Gonzalez and Woods

Ideal High Pass Filtering

- Opposite of low pass filtering: eliminate center (low frequency values), keeping others
- High pass filtering causes image **sharpening**
- If we use circle as cutoff again, size affects results
  - Large cutoff = More information removed



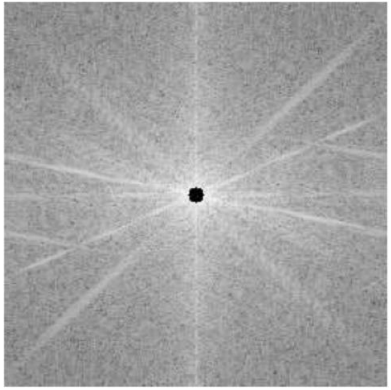
DFT of Image after high pass Filtering



Resulting image after inverse DFT

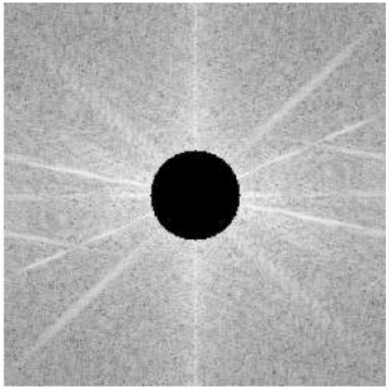
Ideal High Pass Filtering-Effects of cutoffs

High pass filtering of DFT with filter Cutoff D = 5



Low cutoff frequency removes Only few lowest frequencies

High pass filtering of DFT with filter Cutoff D = 30

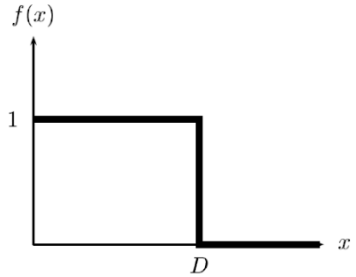


High cutoff frequency removes many frequencies, leaving only edges

Butterworth Filtering

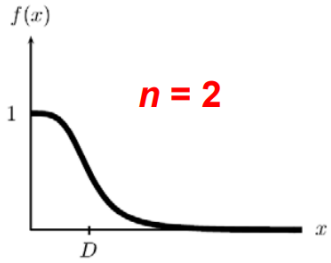
- To avoid ringing, can use circle with more gentle cutoff slope
- **Butterworth filters** have more gentle cutoff slopes
- Ideal low pass filter

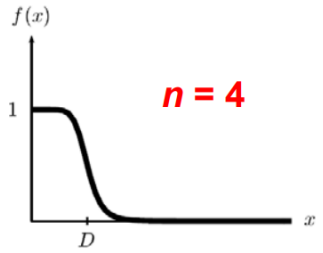
$$f(x) = \begin{cases} 1 & \text{if } x < D \\ 0 & \text{if } x \geq D \end{cases}$$



- Butterworth low pass filter

$$f(x) = \frac{1}{1 + (x/D)^{2n}}$$



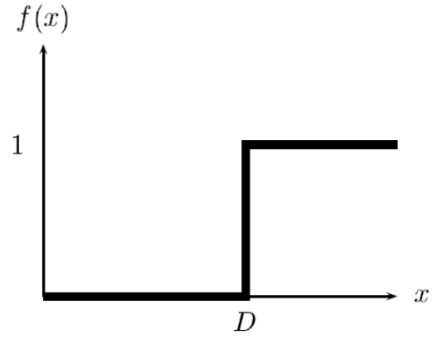


- *n* called **order** of the filter, controls sharpness of cutoff

Butterworth High Pass Filtering

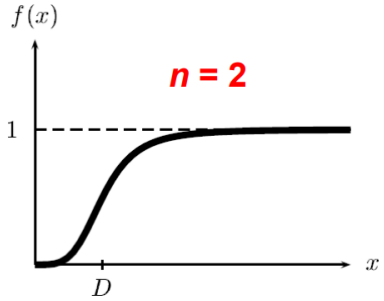
- Ideal high pass filter

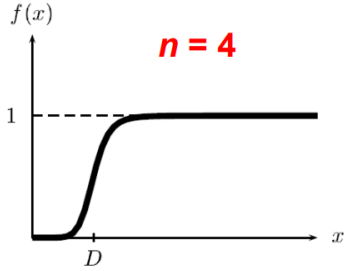
$$f(x) = \begin{cases} 1 & \text{if } x > D \\ 0 & \text{if } x \leq D \end{cases}$$



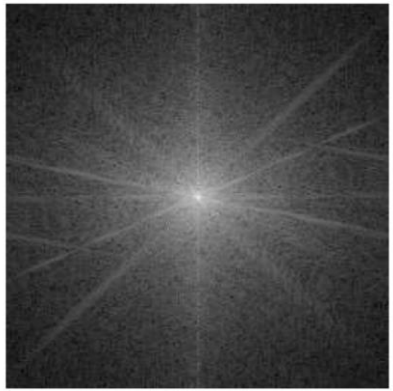
- Butterworth high pass filter

$$f(x) = \frac{1}{1 + (D/x)^{2n}}$$



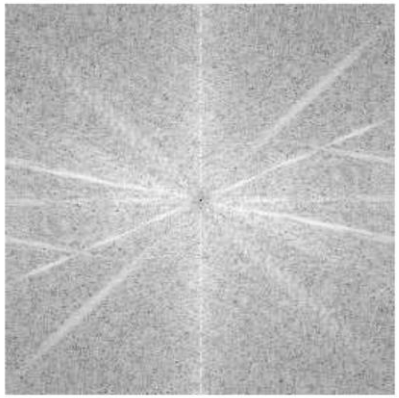


Effect of Butterworth Filtering



DFT of Image after low pass Butterworth filtering

Resulting image after inverse DFT



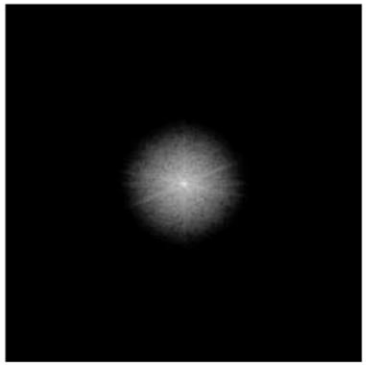
DFT of Image after high pass Butterworth filtering

Resulting image after inverse DFT

Gaussian Filtering

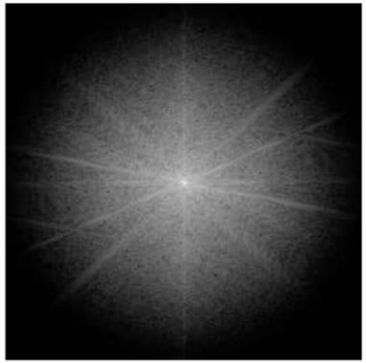
- Gaussian filters can be applied in frequency domain
- Same steps
  - Create gaussian filter
  - Multiply (**DFT of image**) by (**gaussian filter**)
  - Invert result
- **Note:** Fourier transform of gaussian is also a gaussian,
- Just apply gaussian multiply directly (no need to find Fourier transform)

Gaussian Filtering



(a) $\sigma = 10$

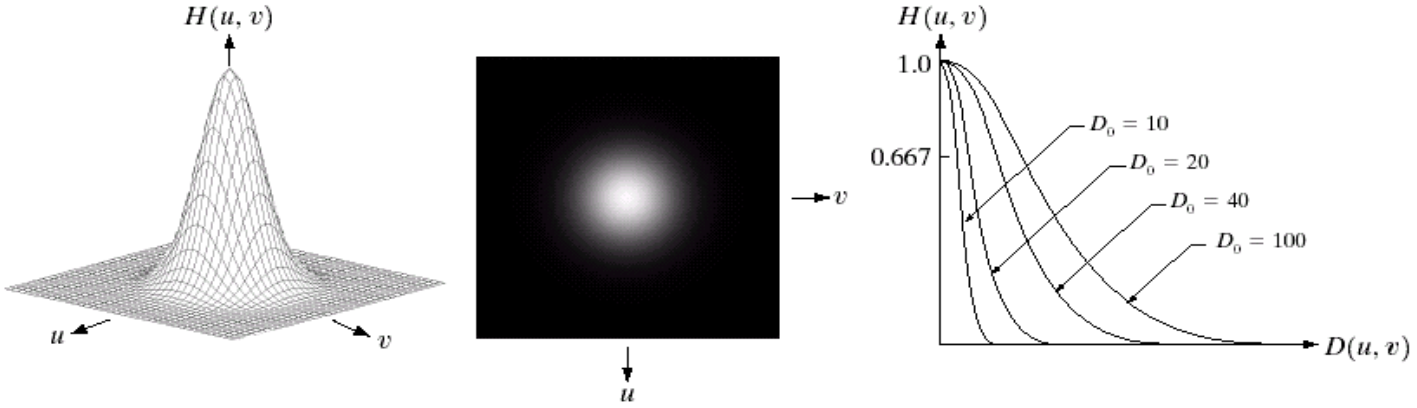
(b) Resulting image



(c) $\sigma = 30$

(d) Resulting image

Gaussian Low Pass Filters



$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

Image courtesy: Gonzalez and Woods

Gaussian Low Pass Filters (GLPF)

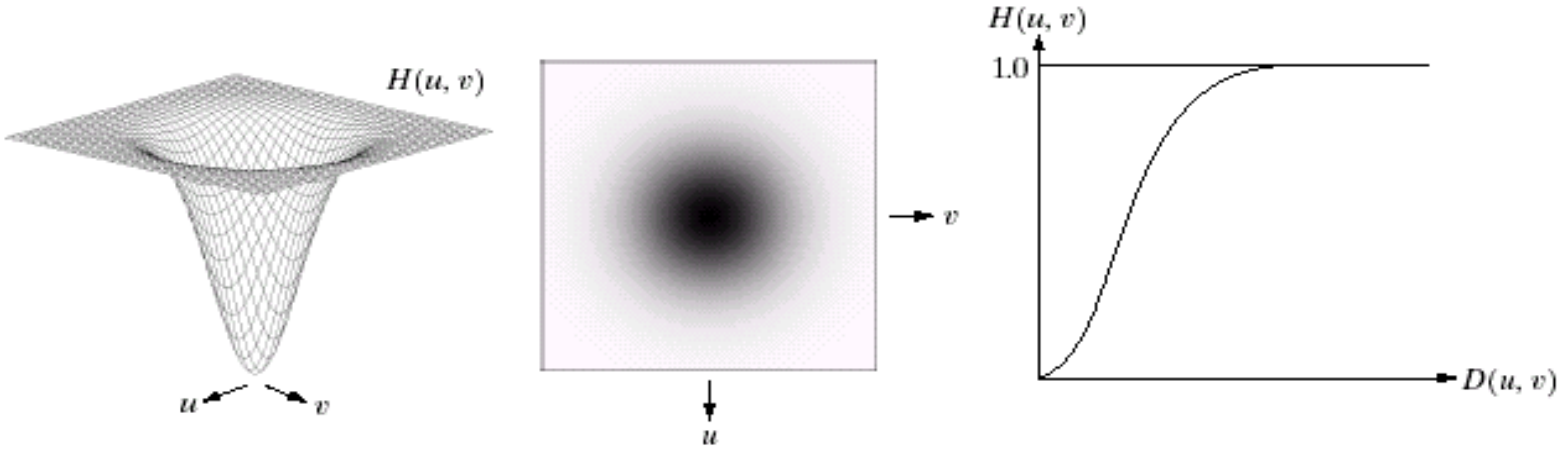
Different lowpass Gaussian filters used to remove blemishes in a photograph

Image courtesy: Gonzalez and Woods

Comparison (ILPF, BLPF, GLPF)

Image courtesy: Gonzalez and Woods

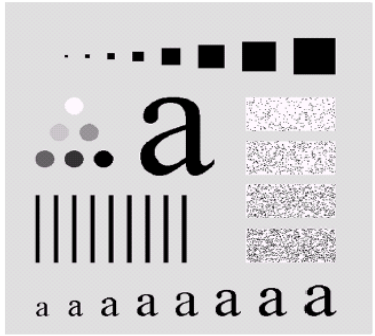
Gaussian High Pass Filters

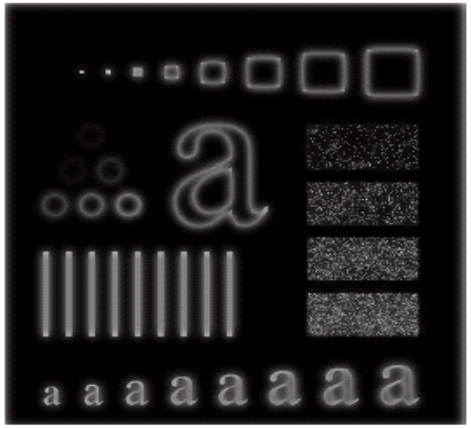


$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0{}^2}$$

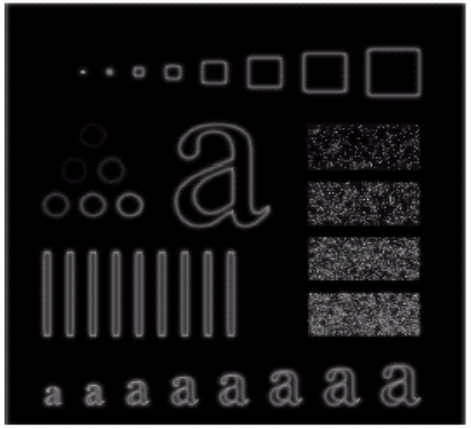
Image courtesy: Gonzalez and Woods

Gaussian High Pass Filters





GHPL with $D_0 = 30$



GHPF with $D_0 = 60$

GHPF with $D_0 = 160$

Image courtesy: Gonzalez and Woods

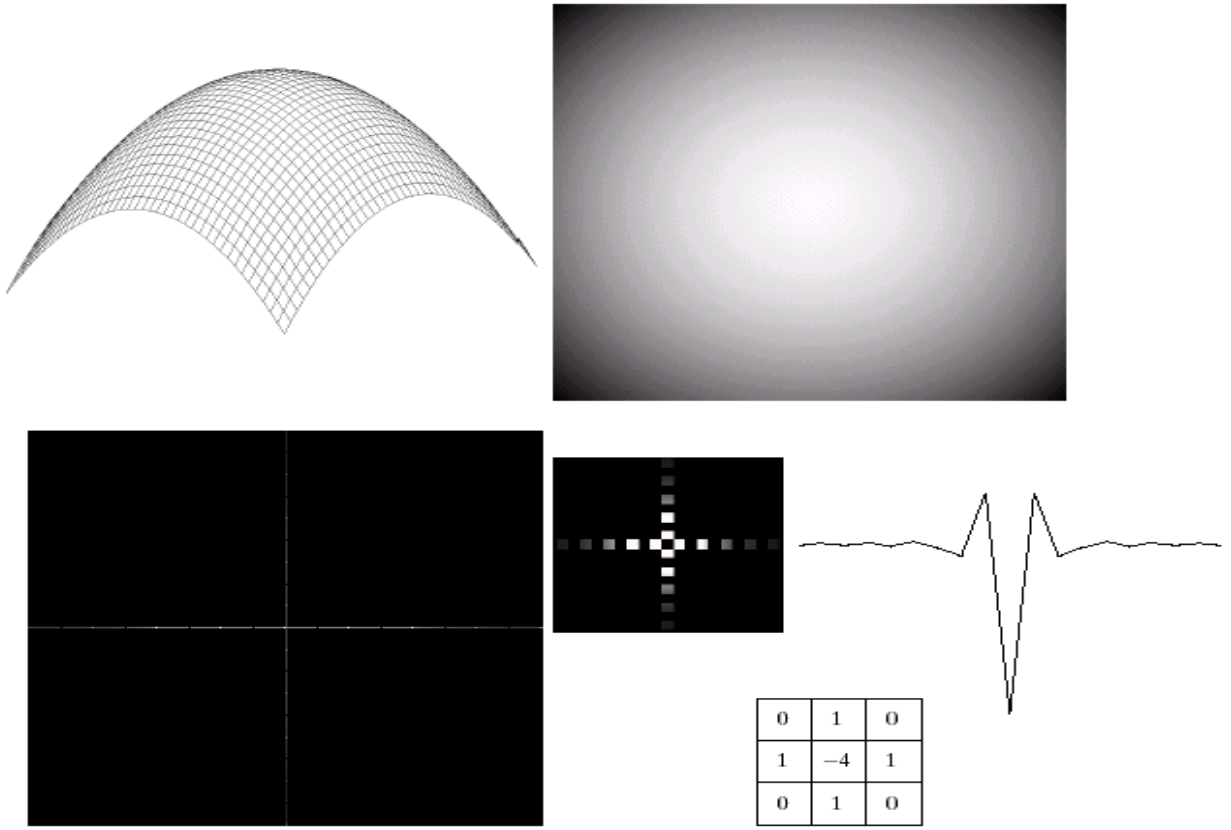
Laplacian in frequency domain

$$\Im[\frac{d^n f(x)}{dx^n}] = (ju)^n F(u)$$

$$\Im[\frac{\partial^2(f(x,y))}{\partial x^2} + \frac{\partial^2(f(x,y))}{\partial y^2}] = (ju)^2 F(u,v) + (jv)^2 F(u,v)$$

$$= -(u^2 + v^2) F(u,v)$$

Laplacian in frequency domain



0	1	0
1	−4	1
0	1	0

Image courtesy: Gonzalez and Woods

Notch Filter

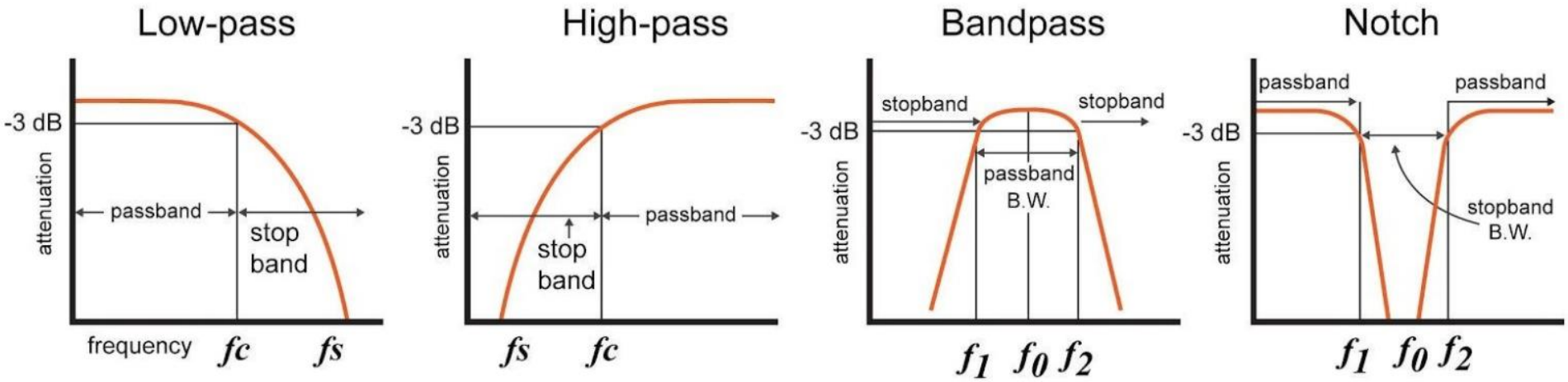


Image credits go to All About Circuits.

Frequency Domain Removal of Periodic Noise

- Periodic noise can be removed in frequency domain
- Periodic noise shows up as spikes away from origin in DFT
- Higher frequency noise = further away from origin

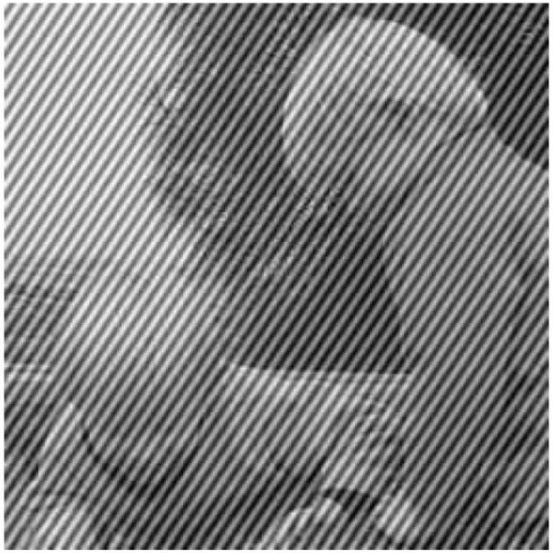
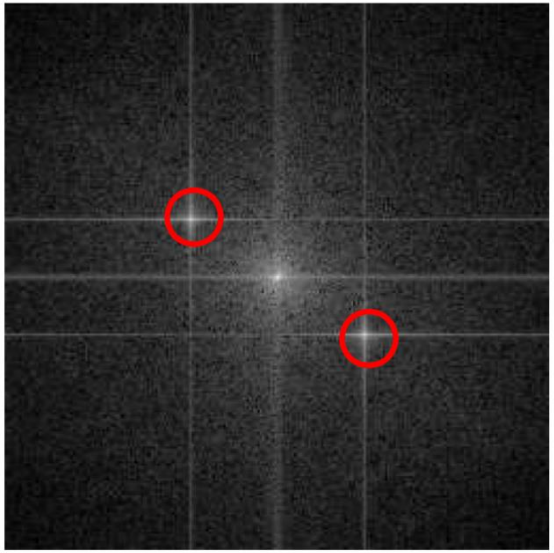


Image with periodic Noise

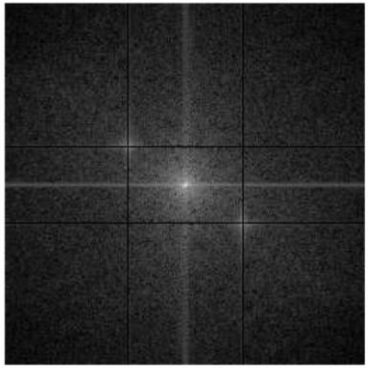


DFT of Image

Frequency Domain Removal of Periodic Noise

Notch Filter

- Notch filter: Set rows, columns of DFT corresponding to noise = 0
- Removes much of the periodic noise

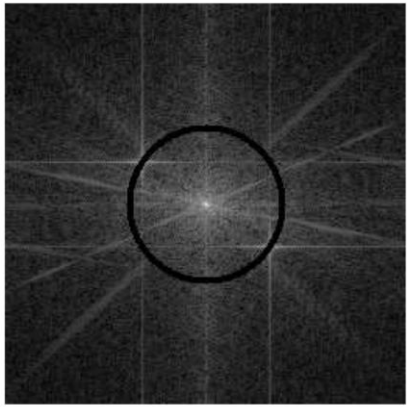


Notch Filter

Result after notch filter applied then inverted

Band Reject Filter

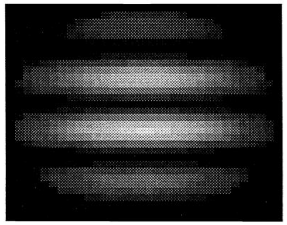
- Create filter with 0's at radius of noise from center, 1 elsewhere
- Apply filter to DFT

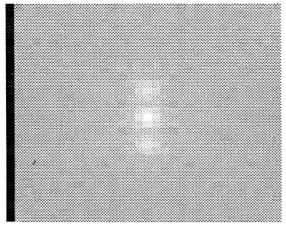


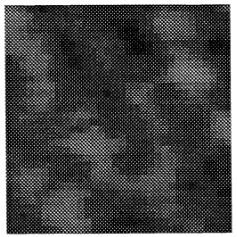
Band Reject Filter

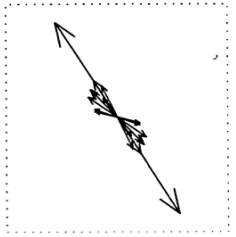
Result after band reject filter applied then inverted

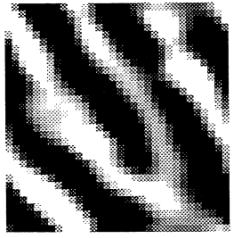
Enhancement in Frequency Domain

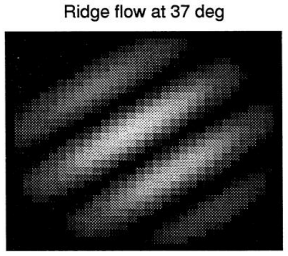


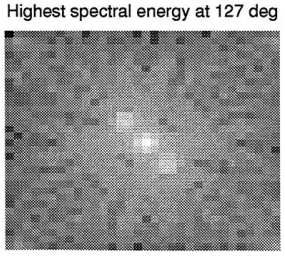


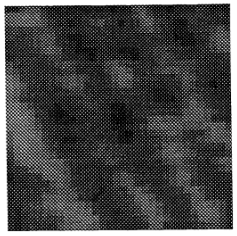


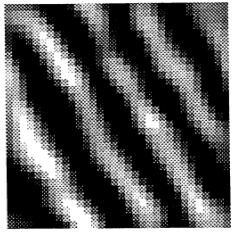








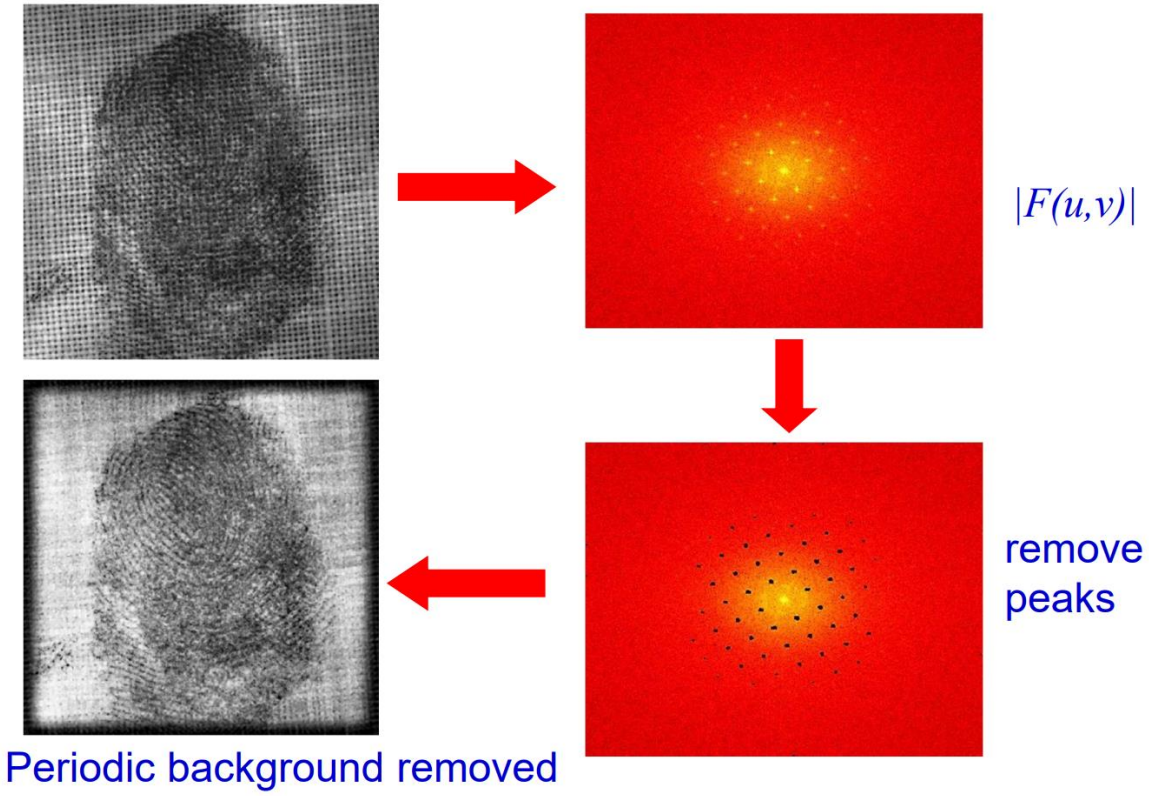




Pradenas et al., “Directional enhancement in the frequency domain in the fingerprint images,” Proc. SPIE Enabling Technologies for Law Enforcement and Security, 1996

Artifact removal

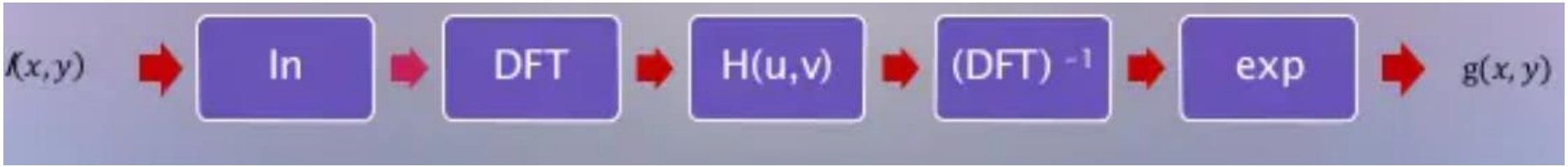
Example – Forensic application



Filtering in frequency domain

- Band reject (Band pass filters)
- Unsharp Masking and High boost filtering
- Homomorphic filtering
  - An image can be considered as product of luminance, L and reflectance, R

$$I(x,y) = L(x,y).R(x,y)$$

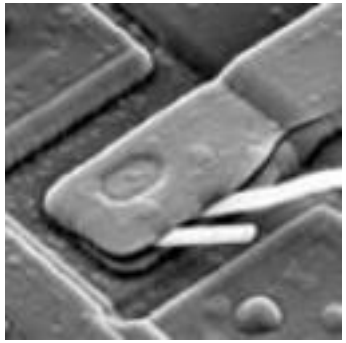


Additional considerations

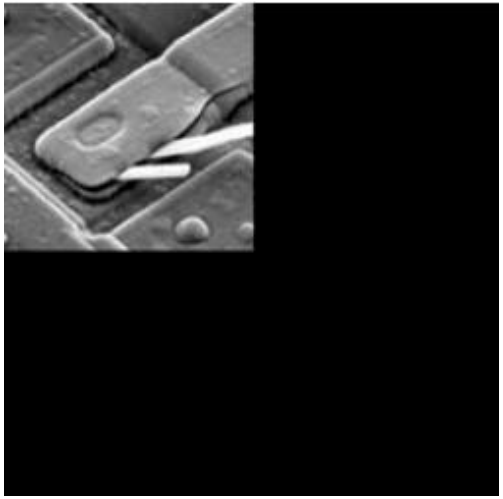
- Circular convolution → Wraparound error
  - Zero padding

Read: https://www.sciencedirect.com/topics/computer-science/cyclic-convolution
https://www.ft.unicamp.br/docentes/magic/khoros/html-dip/c6/s3/front-page.html
https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-341-discrete-time-signal-processing-fall-2005/lecture-notes/lec16.pdf

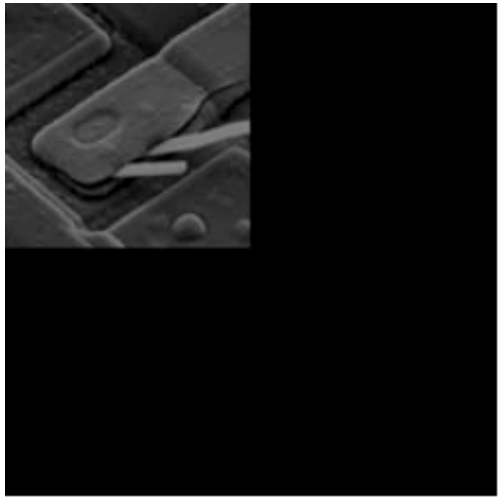
Recipe for transform domain processing



Given: M x N image f



1: pad f_p to size P x Q
where P = 2M, Q = 2N

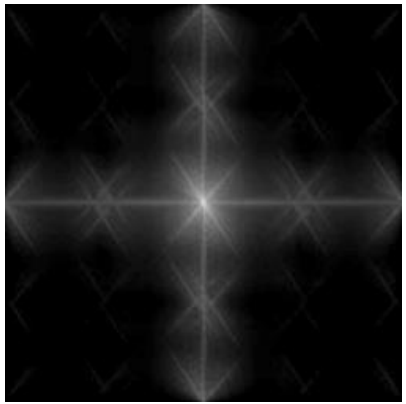


2: Multiply f_p by $(-1)^{(x+y)}$

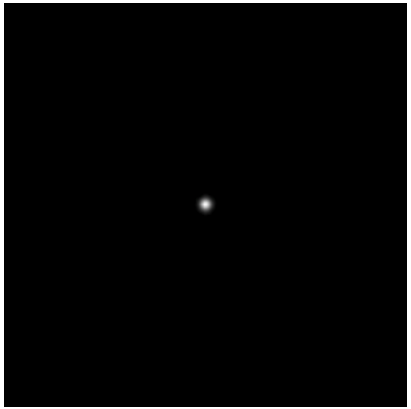
3: Compute $F_p = DFT(f_p)$

$$f(x)e^{\frac{j2\pi u_0 x}{M}} \leftrightarrow F[u-u_0], where\ u_0 = \left(\frac{M}{2}\right)$$

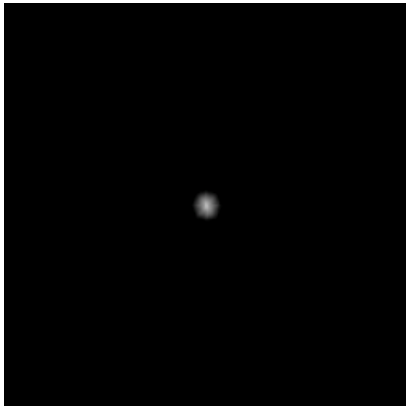
Recipe for transform domain processing



F_p - Fourier Spectrum of f_p

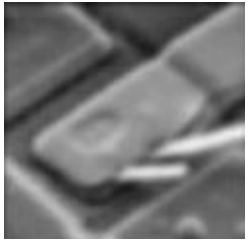


4: Centered Gaussian low pass spectral filter H



5: Compute $G_p = \mathrm{H}F_p$

6: Compute $\mathrm{Re}[IDFT\{G_p\}](-1)^{(x+y)}$



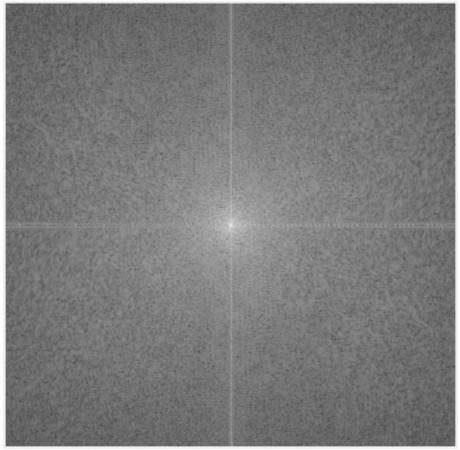
7: Filtered result

Correspondence to spatial filtering

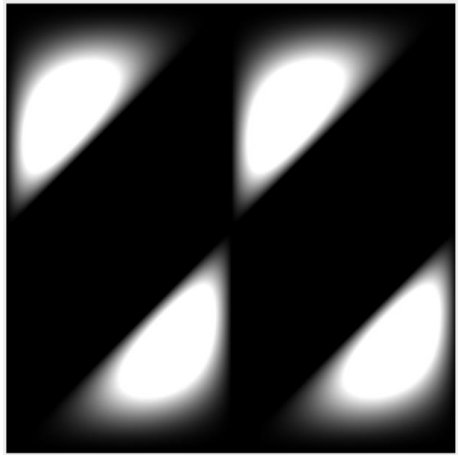
```
f = rgb2gray(imread('boy.jpg'));
```

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

```
h = [-1 0 1; -2 0 2; -1 0 1];
```



```
F = fft2(double(f), 402, 402);
```



```
H = fft2(double(h), 402, 402);
```

```
F_fH = fftshift(H).*fftshift(F);
ffi = ifft2(ifftshift(F_fH));
```

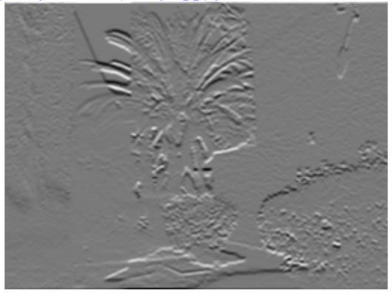
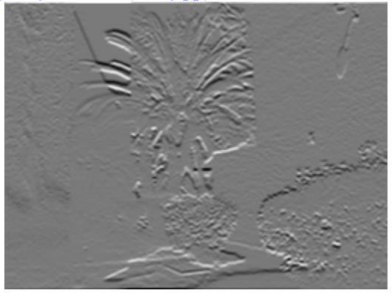
Correspondence to spatial filtering

```
%Sobel filter in frequency domain
f = rgb2gray(imread('boy.jpg'));
h = [-1 0 1; -2 0 2; -1 0 1];
F = fft2(double(f), 402, 402);
H = fft2(double(h), 402, 402);
F_fH = fftshift(H).*fftshift(F);
ffi = ifft2(ifftshift(F_fH));
```

https://www.cs.uregina.ca/Links/class-info/425/Lab5/index.html

Spatial Domain Filtering	Frequency Domain Filtering

`%Create the Spacial Filtered Image`
`f = imread('entry2.png');`
`h = fspecial('sobel');`
`sfi = imfilter(double(f),h, 0, 'conv');`
	`%Create the Frequency Filtered Image`
	`f = imread('entry2.png');`
	`h = fspecial('sobel');`
	`PQ = paddedsize(size(f));`
	`F = fft2(double(f), PQ(1), PQ(2));`
	`H = fft2(double(h), PQ(1), PQ(2));`
	`F_fH = H.*F;`
	`ffi = ifft2(F_fH);`
	`ffi = ffi(2:size(f,1)+1, 2:size(f,2)+1);`

`%Display results (show all values)`
`figure,imshow(sfi, []);`
 
	`%Display results (show all values)`
	`figure, imshow(ffi,[])`
	 

`%The abs function gets correct magnitude`
`%when used on complex numbers`
`sfim = abs(sfi);`
`figure, imshow(sfim, []);`
 
	`%The abs function gets correct magnitude`
	`%when used on complex numbers`
	`ffim = abs(ffi);`
	`figure, imshow(ffim, []);`
	 

Frequency Domain vs Spatial Domain Filtering

- Any linear spatial filter
- Guide the process of spatial filter design

Alberto Carini, Giovanni L. Sicuranza, *Fourier nonlinear filters*, Signal Processing, Vol. 94, pp. 183-194, 2014

Related Topics

- Gabor filters
- Wavelets
- Shape descriptors

References

- G & W (4.5.1, 4.5.2, 4.5.5, 4.6 – 4.11)
- http://mstrzel.eletel.p.lodz.pl/mstrzel/pattern_rec/fft_ang.pdf
- http://www.laurentnajman.org/uploads/ImageCourse/filtering_ln.pdf

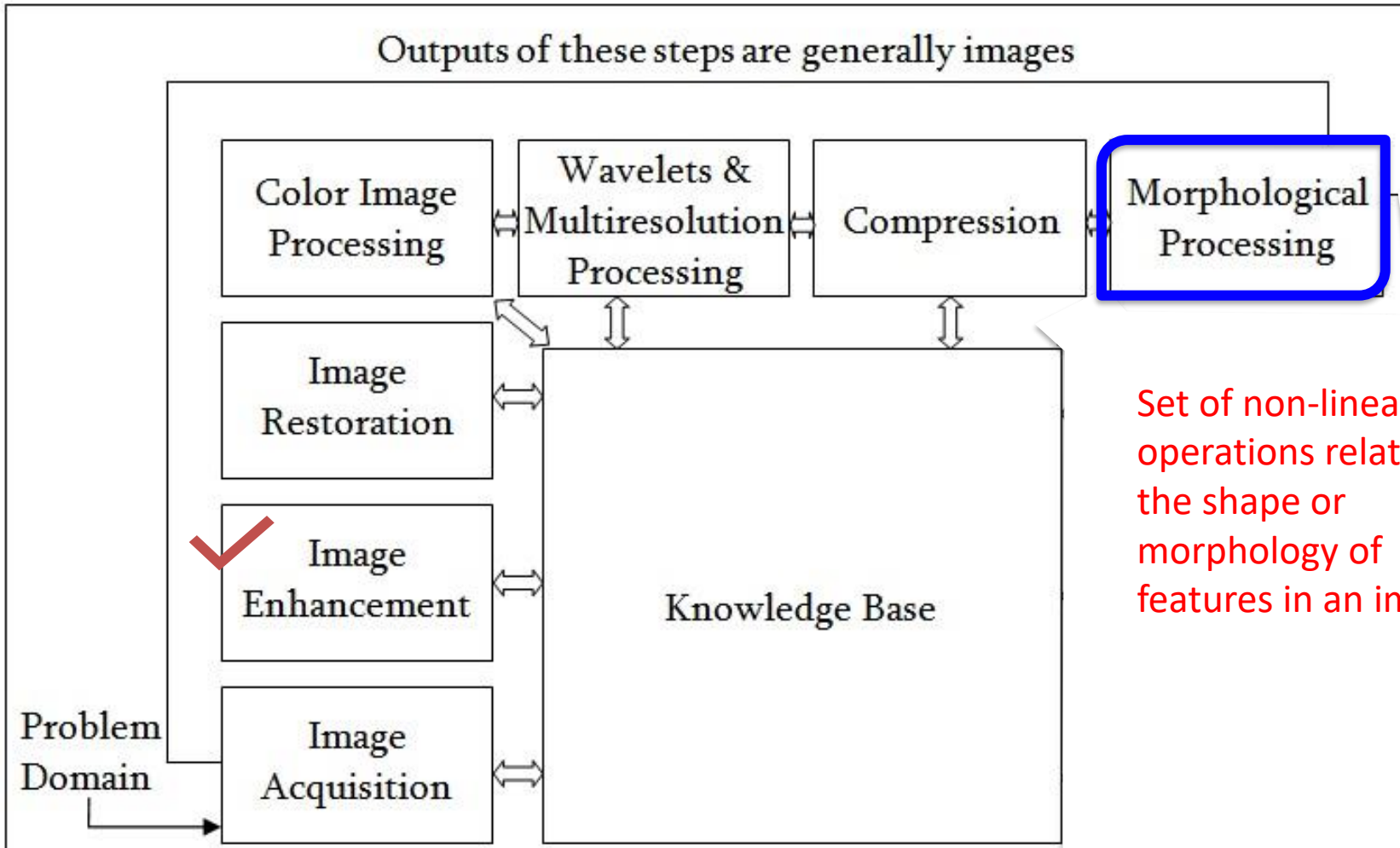
Digital Image Processing (CSE/ECE 478)
Morphological Processing

Center for Visual Information Technology (CVIT), IIIT Hyderabad

Ravi Kiran

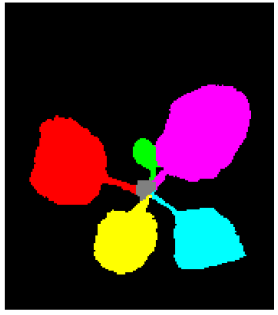
and

Sudipta Banerjee



Set of non-linear operations related to the shape or morphology of features in an image

Plant Phenotyping



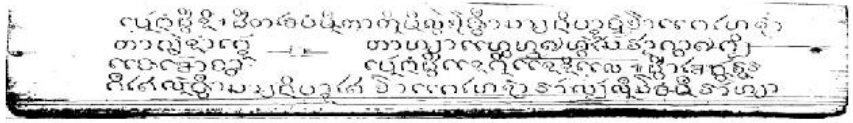
Recognizing Scene Text

https://cvit.iiit.ac.in/research/projects/cvit-projects/scene-text-understanding

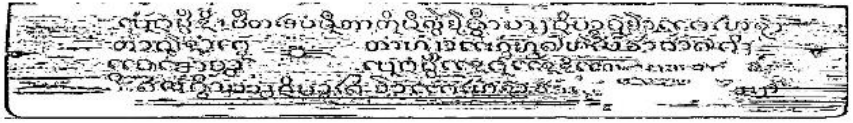
Document Image Analysis

a) RGB image

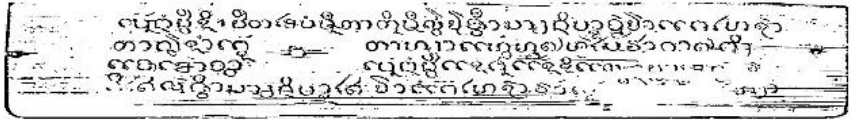
b) Noise reduction image



c) Binary image by Otsu's algorithm



d) Binary image by Niblack's algorithm



e) Binary image by Sauvola's algorithm

Figure 2. Samples of palm leaf images

Background Subtraction

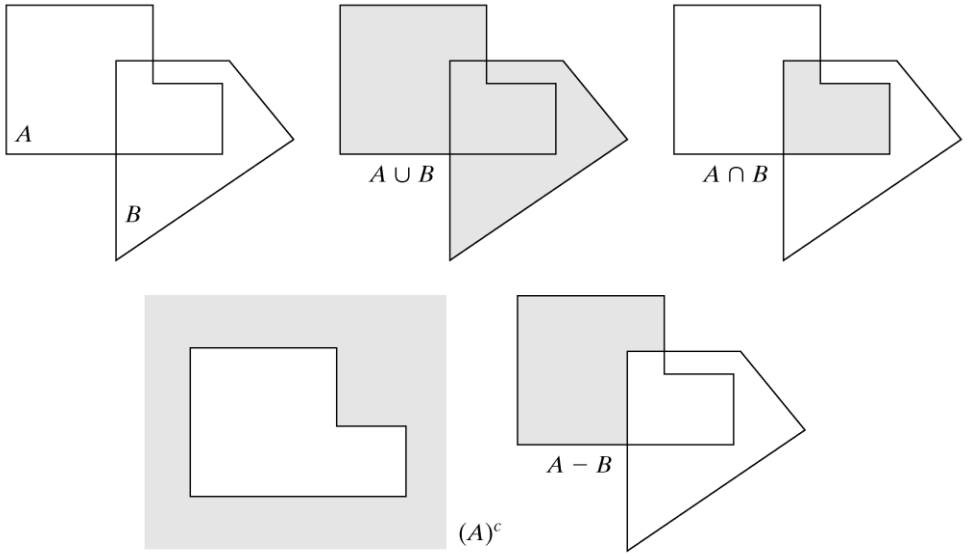
Introduction to Morphological Operators

Image – Set of Pixels

- Basic idea:
  - Object/Region = set of pixels (or coordinates of pixels)

- 0 = background
- 1 = foreground

Object = set of pixels (or coordinates of pixels)



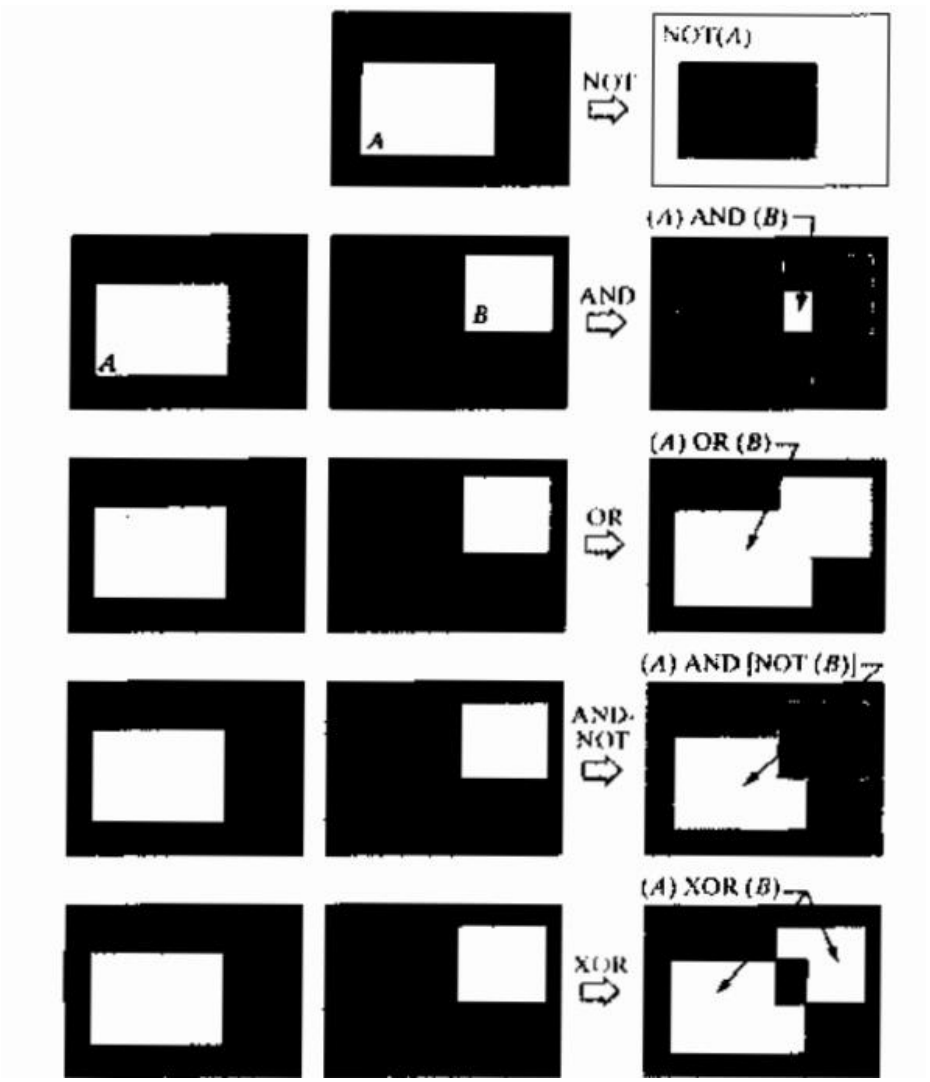
a b c
d e

FIGURE 9.1
(a) Two sets A and B. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B.

Basic operations on shapes

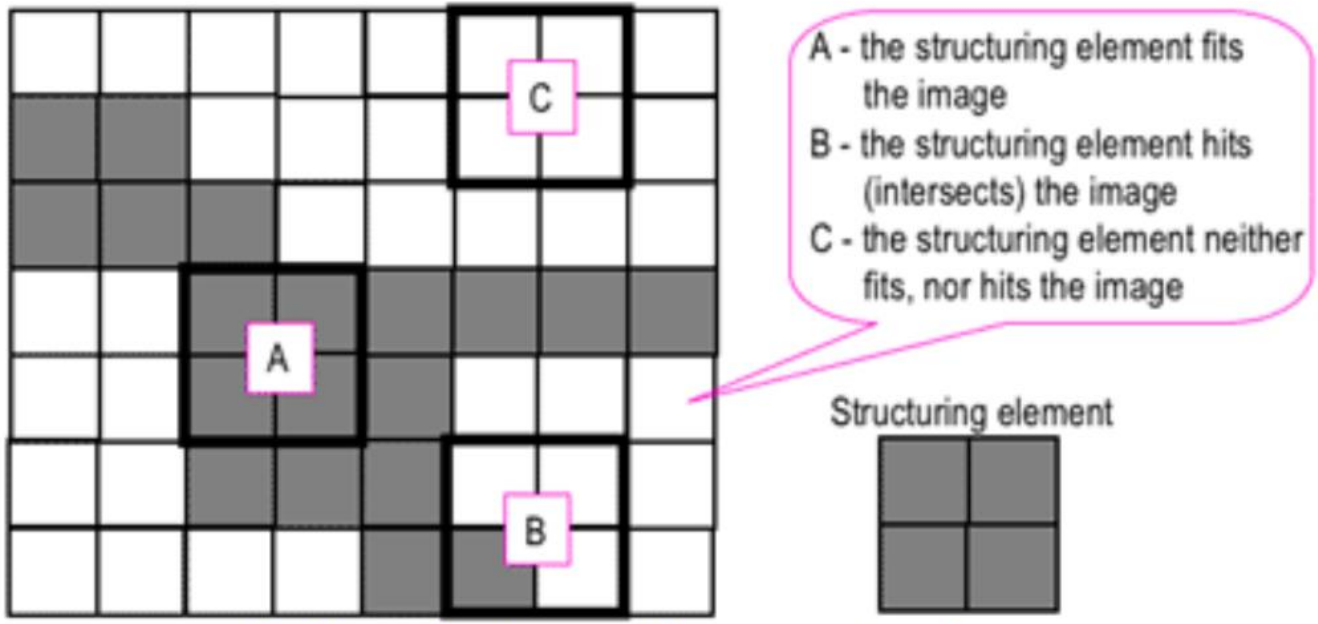
From: Digital Image Processing, Gonzalez,Woods And Eddins

Logical Operations on Binary Images



From: Digital Image Processi
And Eddins

Structuring Element (Kernel)



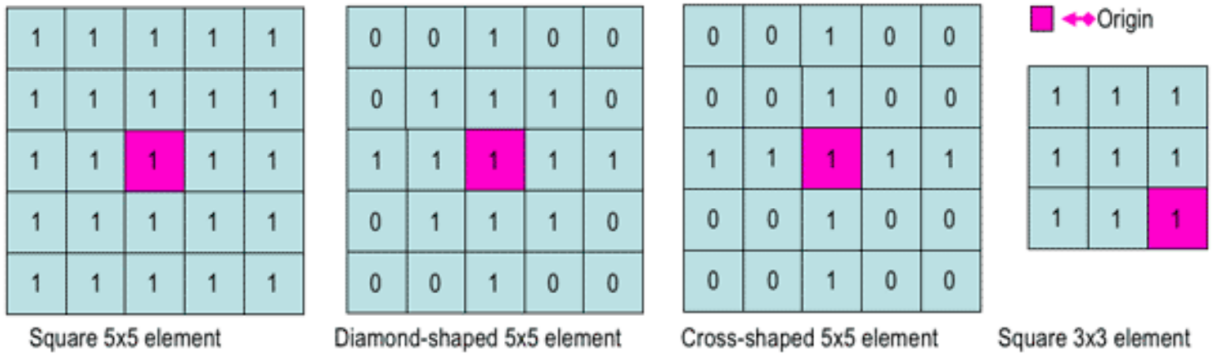
Probing of an image with a structuring element
(white and grey pixels have zero and non-zero values, respectively).

Image Courtesy: https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm

Structuring Element

The **structuring element** is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one:

- The matrix dimensions specify the *size* of the structuring element.
- The pattern of ones and zeros specifies the *shape* of the structuring element.
- An *origin* of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.



Examples of simple structuring elements.

Image Courtesy: https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm

Structuring Element