Outline

- Highpass Filters in the Frequency domain
  - Ideal Highpass Filters
  - Butterworth Highpass Filters
  - Gaussian Highpass Filters
  - Laplacian in the Frequency Domain
- Homomorphic Filtering
- Implementation Issues for Frequency Domain Filters
- Correlation of Images
- Practice Problems

Highpass Filters in the Frequency Domain

- Considering that the image consists of some low-frequency components and high frequency components: if we have a filter preserving the low frequency components, then one minus that filter will preserve the high-frequency components
- Then, we use the general form of highpass filters as follows

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

Highpass Filters in the Frequency Domain

- Ideal highpass filter has the form

$$H(u,v) = \begin{cases} 0 & D(u,v) \leq D_0 \\ 1 & D(u,v) \geq D_0 \end{cases}$$

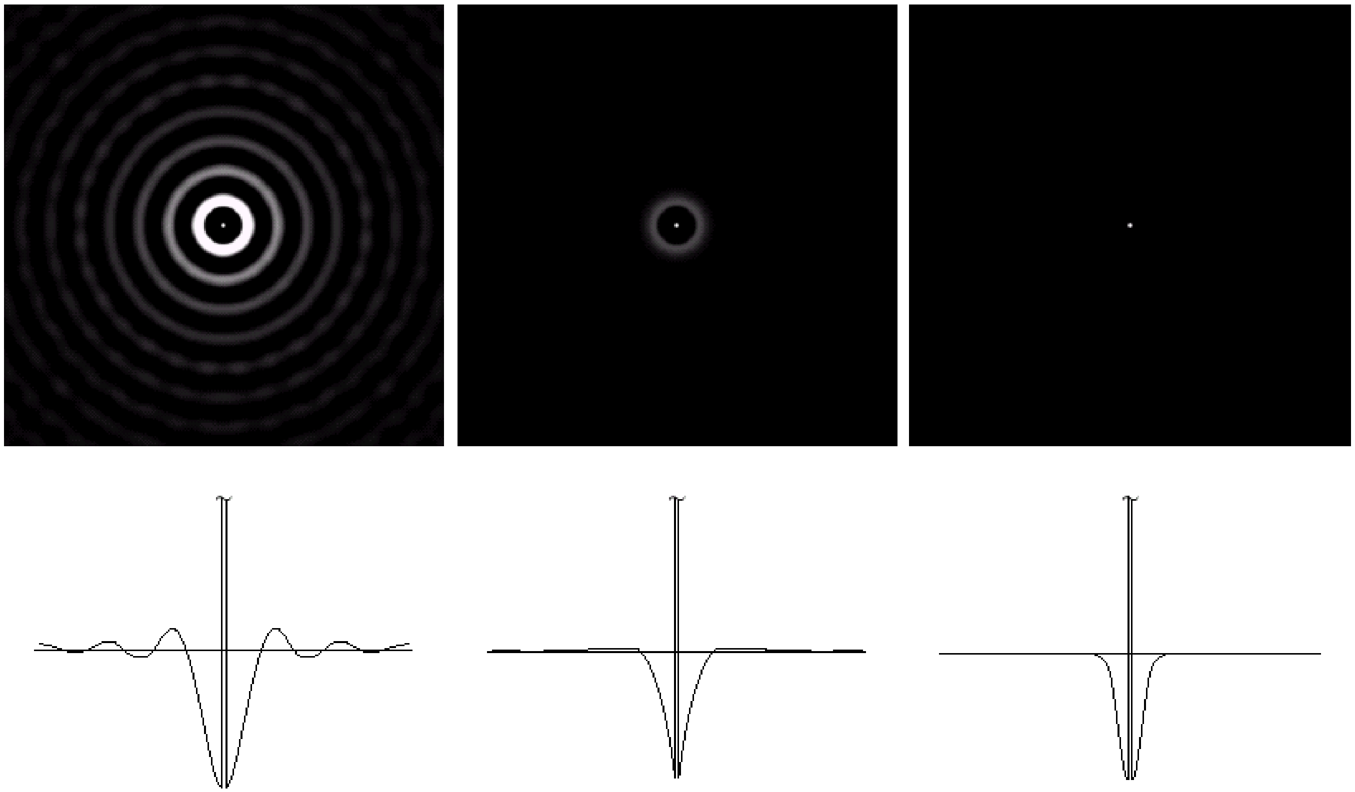
- Butterworth

$$H(u,v) = 1 - \frac{1}{1 + [D(u,v)/D_0]^{2n}} = \frac{1}{1 + [D_0/D(u,v)]^{2n}}$$

- Gaussian

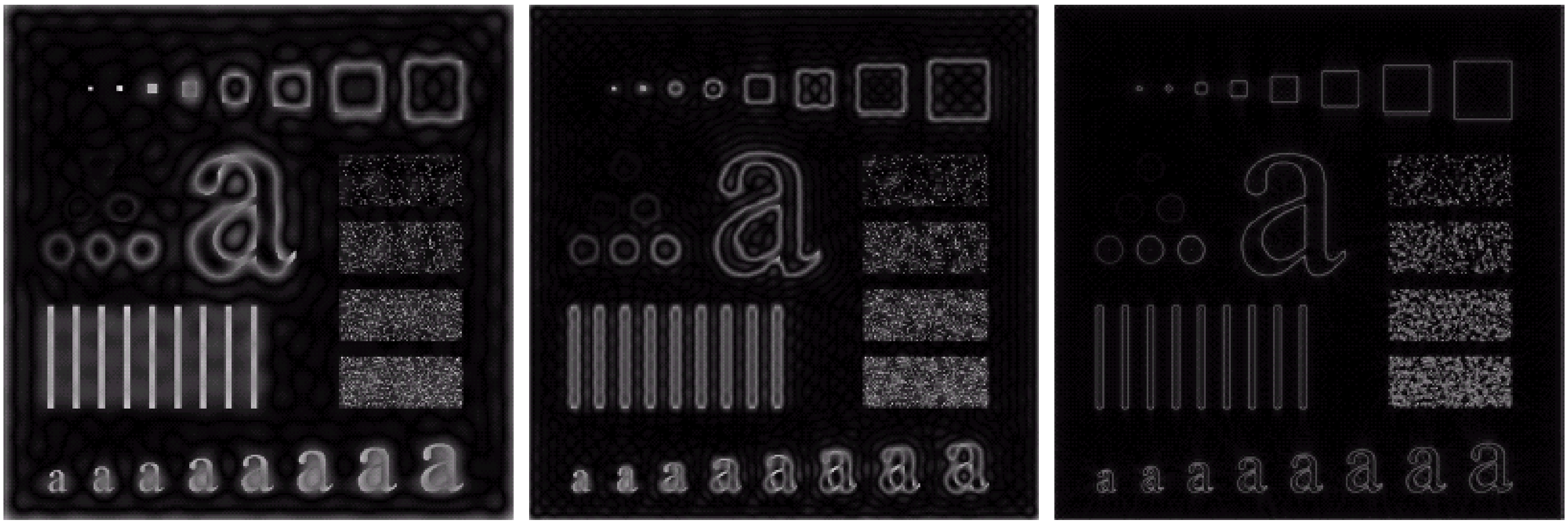
$$H(u,v) = 1 - e^{-D^2(u,v)/(2D_0^2)}$$

Highpass Filters in the Spatial Domain



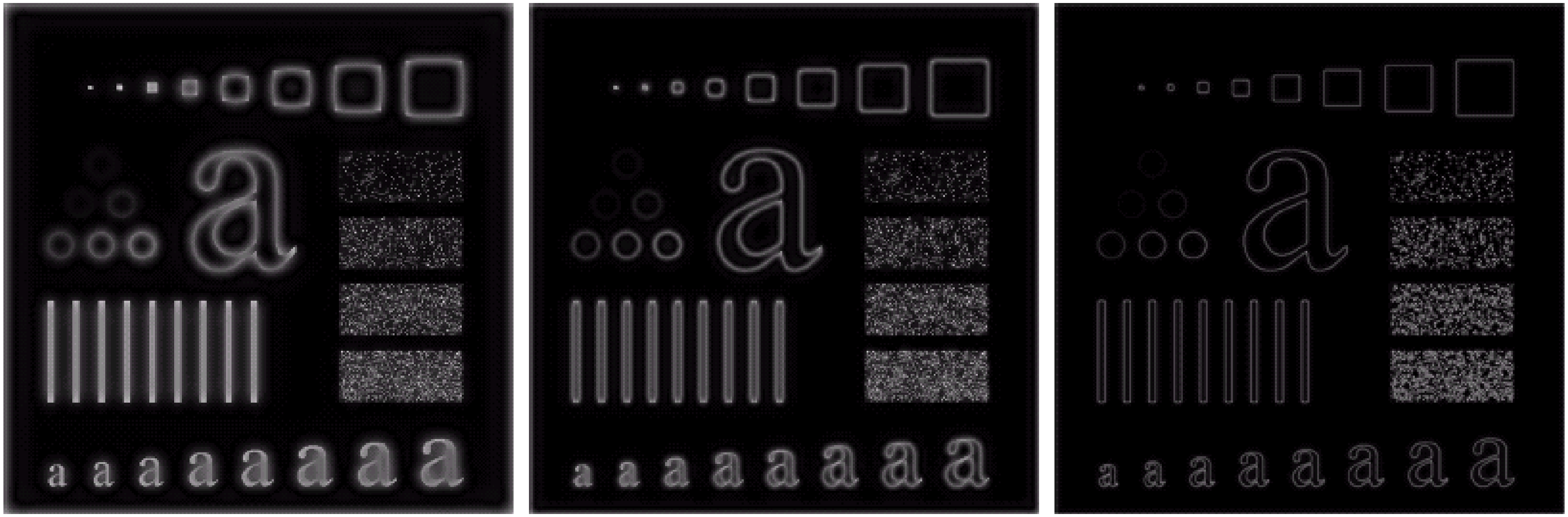
Left: ideal, center: Butterworth, right: Gaussian

Ideal Highpass Filtering: Image Example



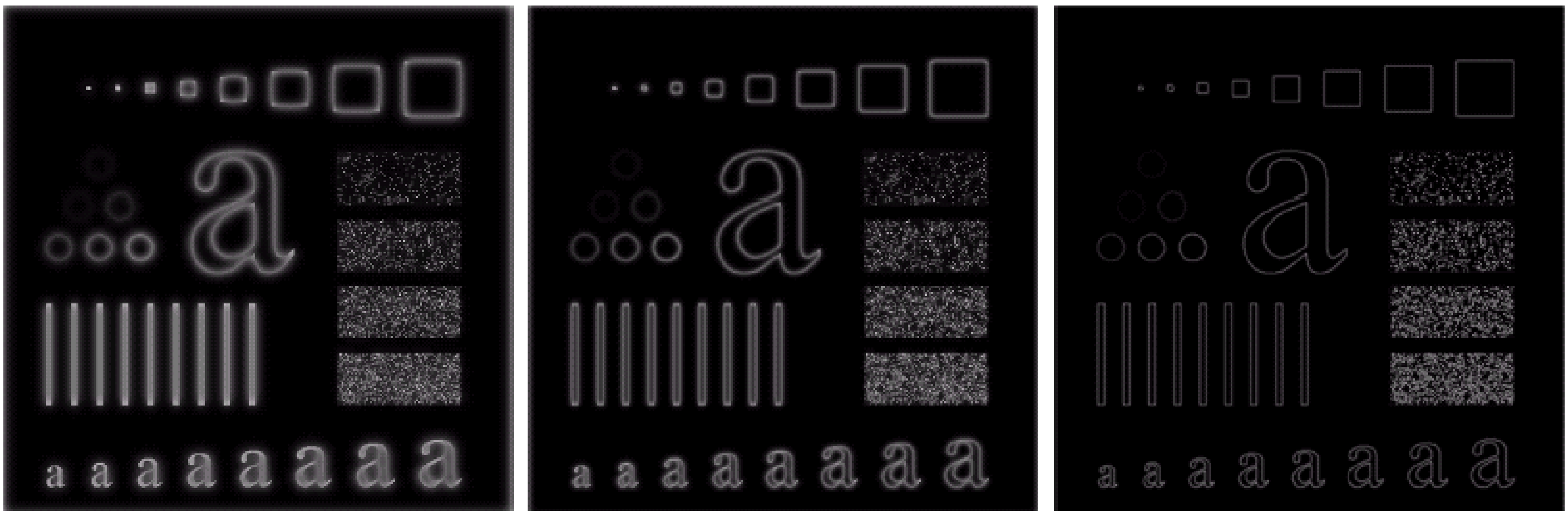
Left: ideal, center: Butterworth, right: Gaussian

Butterworth Highpass Filtering: Image Example



Left: ideal, center: Butterworth, right: Gaussian

Gaussian Highpass Filtering: Image Example



Left: ideal, center: Butterworth, right: Gaussian

Laplacian in the Frequency Domain

- Using the FT property we know that

$$\mathcal{F}\left\{\frac{\mathrm{d}f(x)}{\mathrm{d}x^n}\right\} = (ju)^n F(u)$$

- Then the Laplacian in the frequency domain is

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

- If we center the FT by multiplying the image by $(-1)^{x+y}$, then we must center the filter as well

$$H(u,v) = -[(u - M/2)^2 + (v - N/2)^2]$$

- The spatial representation can be obtained by taking the inverse FT

Modifications to Laplacian in the Frequency Domain

- We can further enhance the image by adding the original image to the fitlered version as in the spatial domain

$$H(u, v) = 1 + [(u - M/2)^2 + (v - N/2)^2]$$

- We can also use boosting to increase the average gray level of the image as in the spatial domain

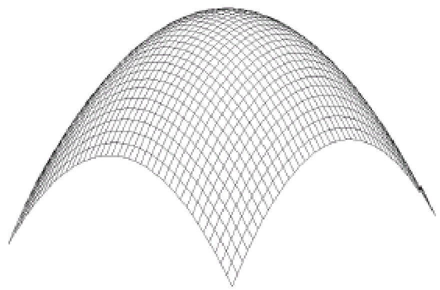
$$H(u, v) = 1 + [(u - M/2)^2 + (v - N/2)^2]$$

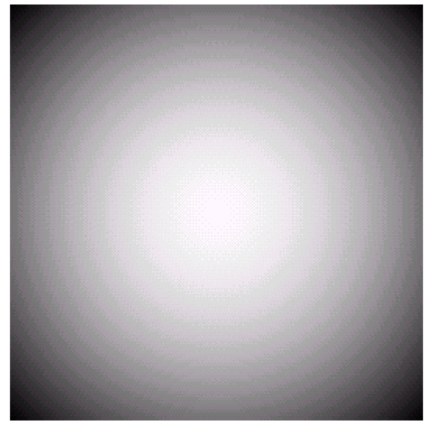
- We can emphasize the frequency components more when necessary

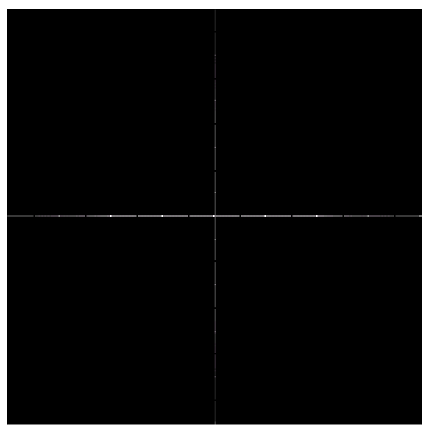
$$H(u, v) = 1 + a[(u - M/2)^2 + (v - N/2)^2]$$

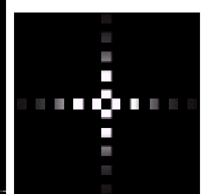
where $a > 1$

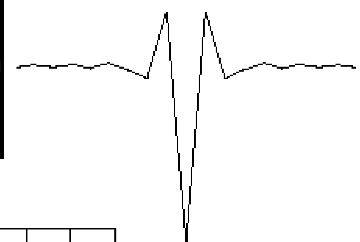
Laplacian Filter







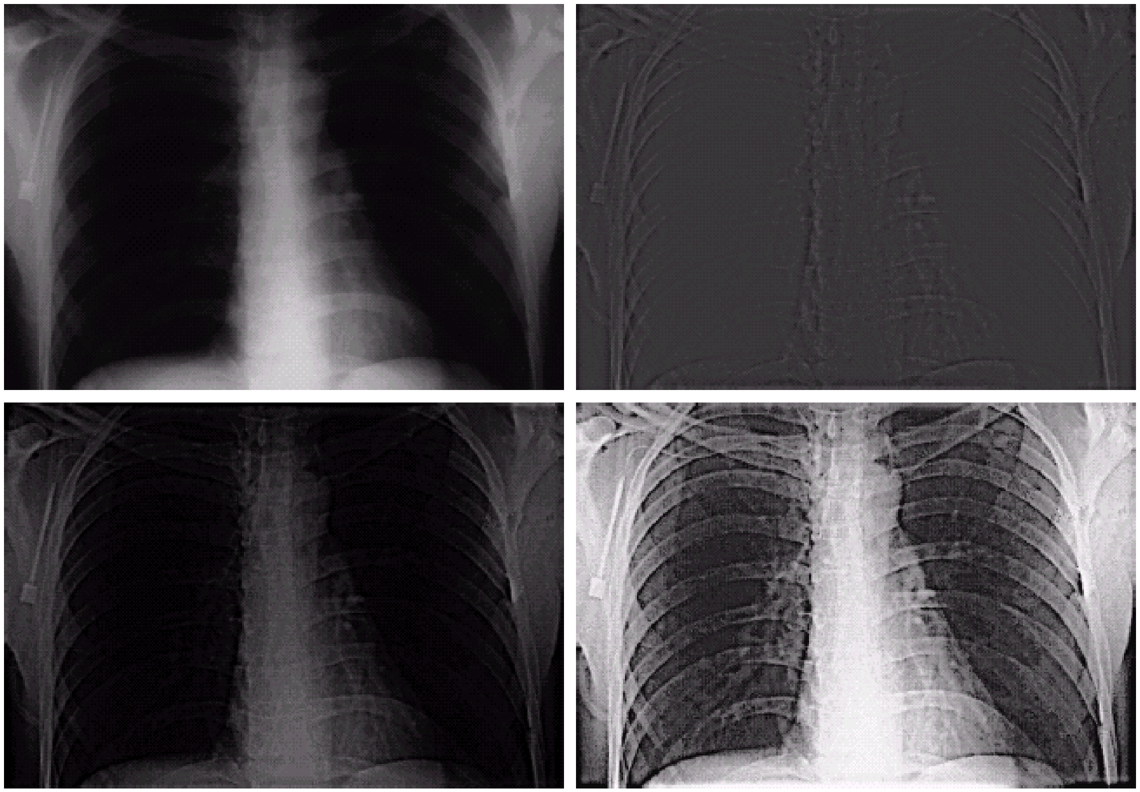




0	1	0
1	−4	1
0	1	0

Top-left: Freq. Domain, top-right, Freq. domain as an image, bottom-left: spatial domain, bottom-right: spatial domain zoomed, approximate used in previous chapter

High Frequency Emphasis: Image Example



Top-left: original, top-right: butterworth, bottom-left: high-freq. comp. emphasized, bottom-right: histogram equalized version of bottom-left

Homomorphic Filtering

- What if we want to enhance the high-freq components and low-frequency components in one combined filtering operation
- Is this possible: the answer is yes
- Let us start by modeling our image as a product of illumination and reflectance components
- Illimunation: our light source, usually low pass nature
- Reflectance: property of the object, has also high-pass nature
- The image is then modeled as

$$f(x,y) = i(x,y)r(x,y)$$

Homomorphic Filtering (cont.)

- We would like to process the illumination and reflectance components separately
- It is not possible to do that directly, since the model involves the product
- We use a trick and take the log of the image so that the two components are now added

$$z(x,y) = \ln[f(x,y)] = \ln[i(x,y)] + \ln[r(x,y)]$$

- We filter z instead of f and then take the exponential to recover the image

$$S(u,v) = H(u,v)Z(u,v) = H(u,v)F_i(u,v) + H(u,v)F_r(u,v)$$

where F_i and F_r represent the FT of log of illumination and reflectance components

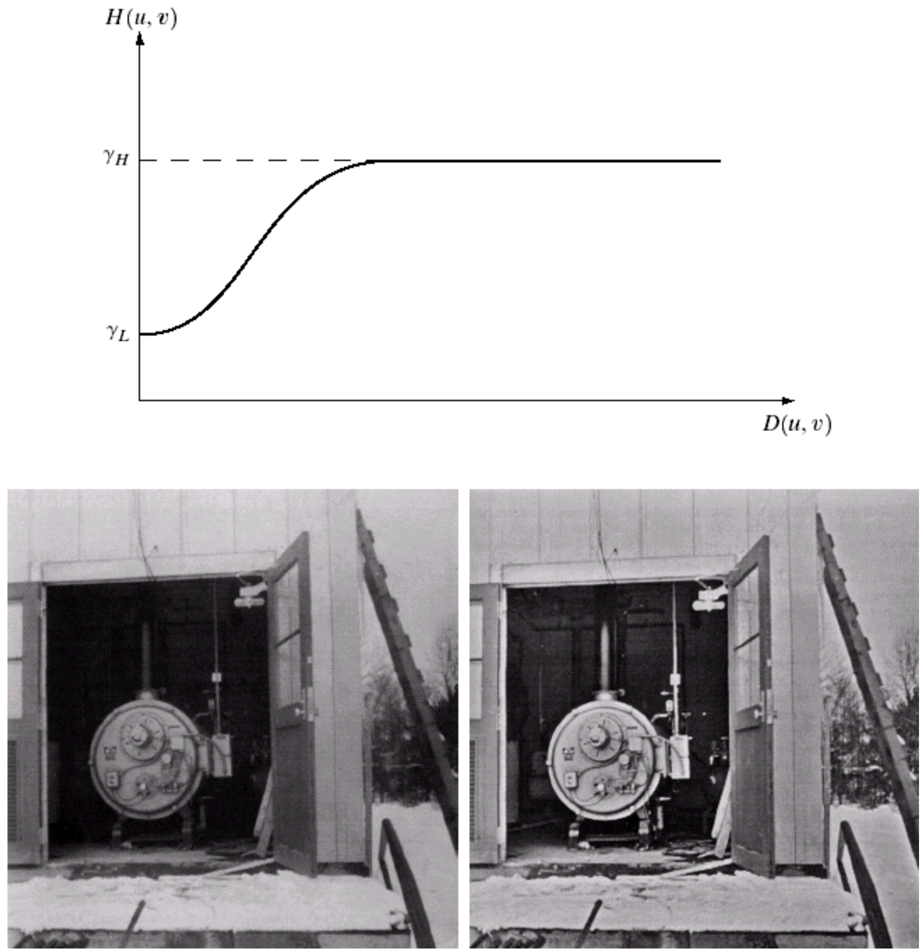
Homomorphic Filtering (cont.)

- The filtered image is the exponential of $s(x, y)$

$$g(x, y) = e^{s(x,y)} = e^{i'(x,y)} e^{r'(x,y)}$$

where $i'(x, y)$ is the inverse FT of $H(u, v)F_i(u, v)$, and $r'(x, y)$ is defined similarly

Homomorphic Filtering: Example



Top: filter, bottom-left: original, bottom-right: filtered

Implementation Issues: Properties of the 2-D FT

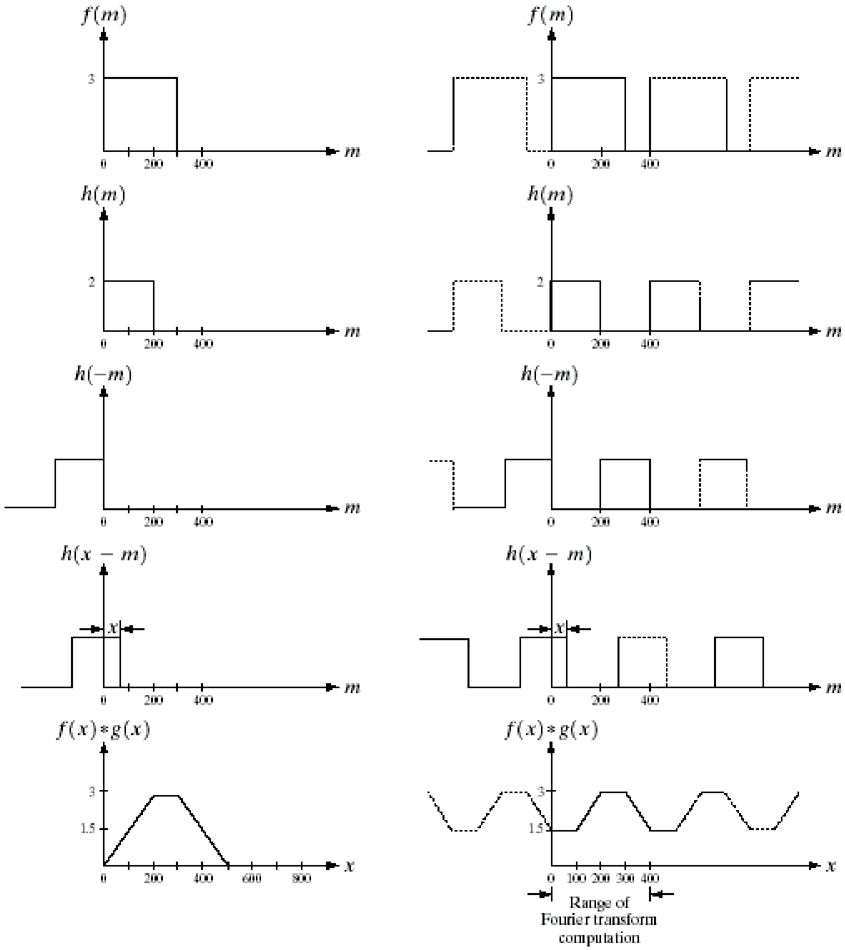
- Shifting property: $f(x - x_0, y - y_0) \leftrightarrow F(u, v)e^{-j2\pi(ux_0/M+vy_0/N}$
- Linearity: $af(x, y) + bg(x, y) \leftrightarrow aF(u, v) + bG(u, v)$
- Scaling: $f(ax, by) \leftrightarrow \frac{1}{|ab|}F(u/a, v/b)$
- Rotation (using polar coordinates): $f(r, \theta + \theta_0) \leftrightarrow F(\omega, \phi + \theta_0)$
- Periodicity of the discrete FT: $F(u, v) = F(u + Mk, v + Nl)$ where the image is of size $M \times N$, and k, l are any integers
- Symmetry: $F^*(u, v) = F(-u, -v)$
- Discrete 2-D FT is separable, we can use two consecutive 1-D FT to implement the 2-D FT

$$F(u, v) = \mathcal{F}_{\mathrm{1D_y}}\{\mathcal{F}_{\mathrm{1D_x}}\{f(x, y)\}\}$$

Implementation Issues: Zero Padding

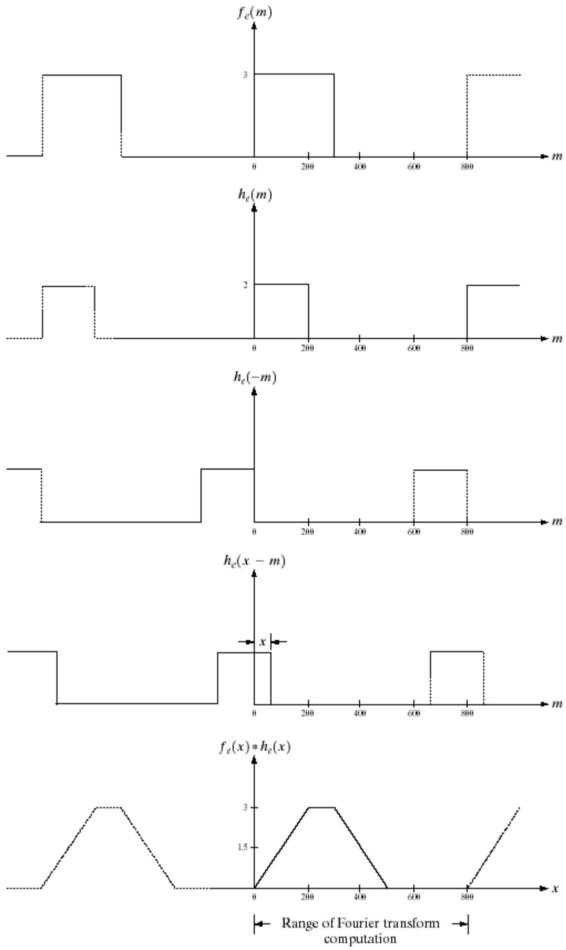
- Discrete FT of any image is periodic
- Therefore when performing the convolution, the parts that should normally be zero have values from adjacent periods causing errenous results
- Solution is to zero pad the signals so that when the convolution is computed there are no contributions from adjacent periods

Implementations Issues: Zero Padding (Cont.)



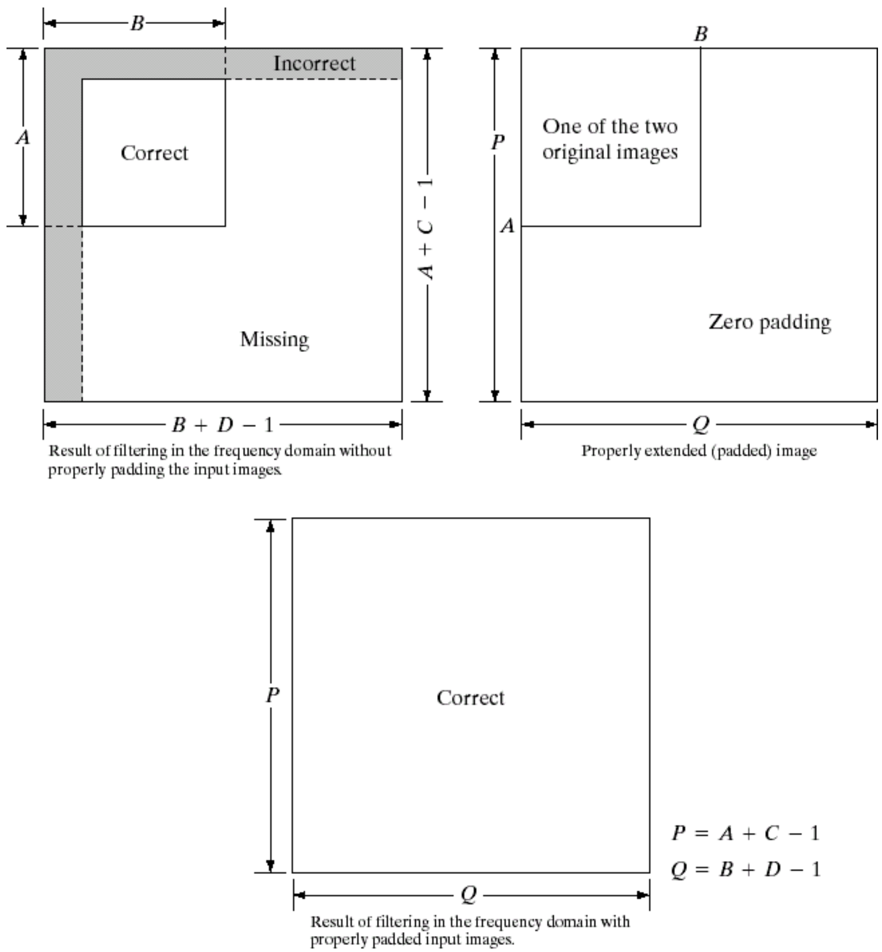
Left: correct convolution, right: convolution with periodicity involved producing errenous results

Implementations Issues: Zero Padding (Cont.)



Convolution with periodicity involved producing correct results due to zero padding

Implementations Issues: Zero Padding in 2-D



Zero-padding in 2-D

Correlation of Images

- Let us define the following function (correlation) of two images

$$f(x,y) \circ g(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) h(x+m, y+n)$$

  where the images are real valued

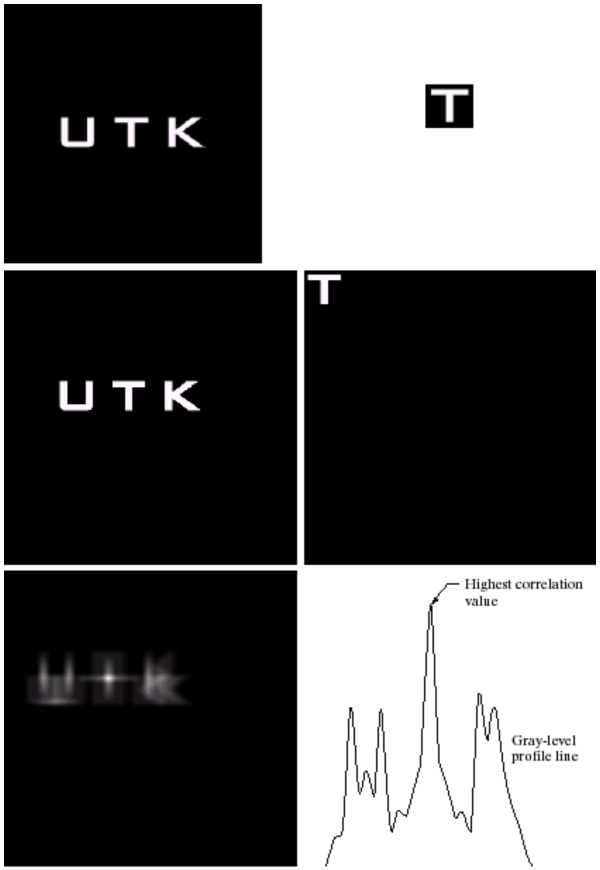
- Similar to the convolution, we need to zero pad to obtain the correct correlation values

- We have the following property analogus to the convolution property

$$f(x,y) \circ g(x,y) \leftrightarrow F*(u,v)G(u,v)$$

- Correlation is mostly used for matching purposes, since the correlation will have a large value when the two functions are close and a small value otherwise

Correlation of Images: Example



Top-left: original image, top right: template of the object to be found, middle-left: original with zero padding, middle-right: original with zero padding, bottom-left: correlation function, bottom-right profile of one row where best match occurs