SMOOTHING, RESTORATION

AND

ENHANCEMENT

Spatial filters for de-noising additive noise

- Skills similar to image enhancement
- Mean filters
- Order-statistics filters
- Adaptive filters

Courtesy: Jen-Chang Liu

Order-statistics filters

- Based on the ordering (ranking) of pixels
  - Suitable for unipolar or bipolar noise (salt and pepper noise)
- Median filters
- Max/min filters
- Midpoint filters
- Alpha-trimmed mean filters

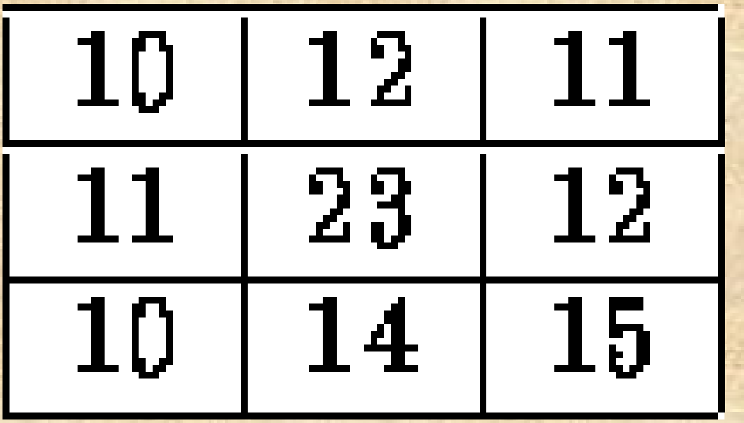
Smoothing - Spatial Domain

The simplest approach is *neighborhood averaging*, where each pixel is replaced by the average value of the pixels contained in some neighborhood about it.

The simplest case is probably to consider the group of pixels centered on the given pixel, and to replace the central pixel value by the un-weighted (for weighted - Gaussian function is commonly used) average of these (nine, in case of 3*3 neighborhood) pixels.

For example, the central pixel in Figure below is replaced by the value:

13 (the nearest integer to the average).



10	12	11
11	23	12
10	14	15

If any one of the pixels in the neighborhood has a faulty value due to noise, this fault will now be smeared over nine pixels as the image is smoothed. This tends to blur the image.

A better approach is to use a ***median filter***.

A similar neighborhood around the pixel under consideration is used, but this time the pixel value is replaced by the *median* pixel value in the neighborhood.

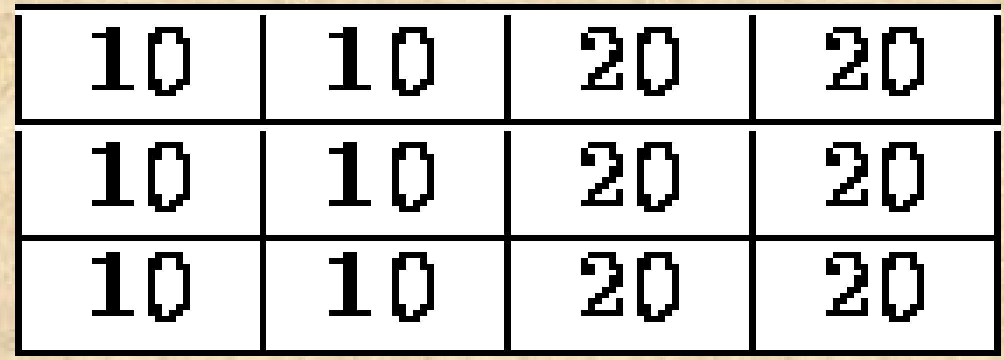
Thus, if we have a 3*3 neighborhood, we write the 9 pixel values in sorted order, and replace the central pixel by the fifth highest value. For example, again taking the data shown in Figure above, the central pixel is replaced by the value:

12

This approach has two advantages.

- Occasional spurious high or low values are not averaged in -- they are ignored

- The sharpness of edges is preserved. To see this, consider the pixel data shown in the next slide.



10	10	20	20
10	10	20	20
10	10	20	20

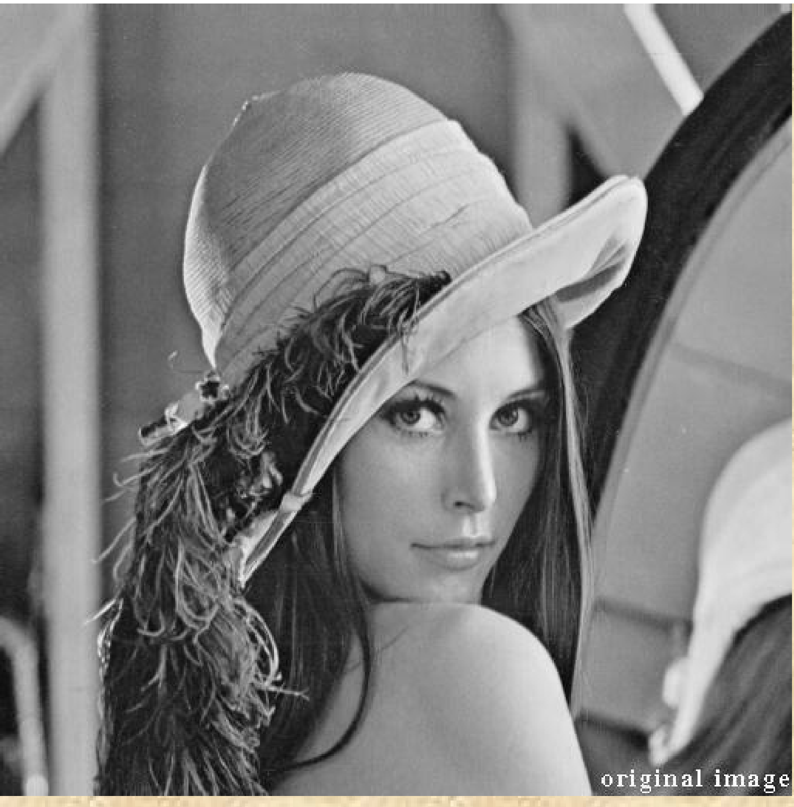
When the neighborhood covers the left-hand nine pixels, the median value is 10; when it covers the right hand ones, the median value is 20; thus the edge is preserved.

- If there are large amounts of noise in an image, more than one pass of median filtering may be useful to further reduce the noise.

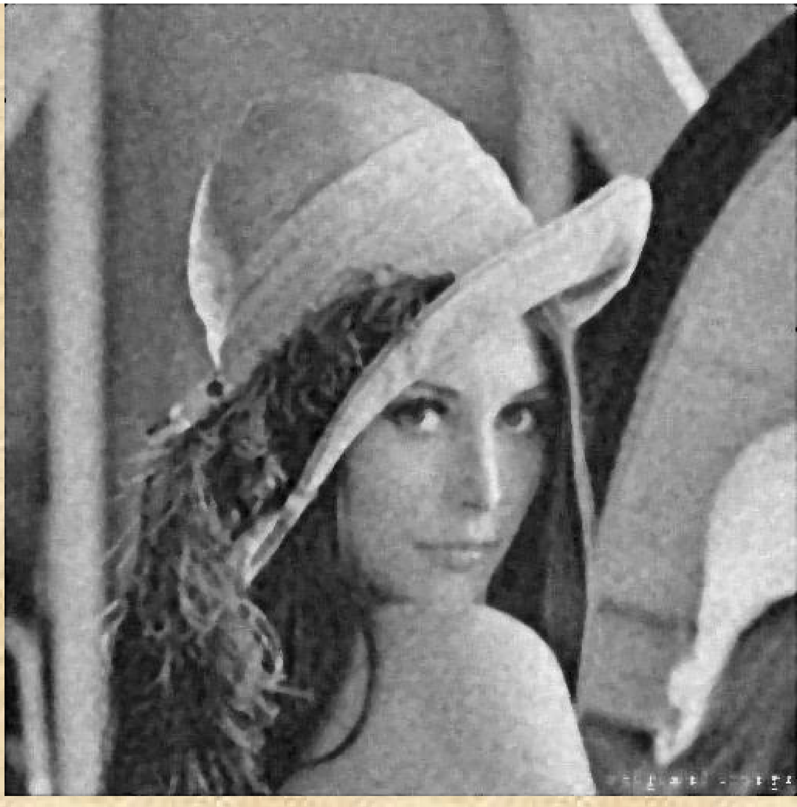
- A rather different real space technique for smoothing is to average multiple copies of the image.

- The idea is that over several images, the noise will tend to cancel itself out if it is independent from one image to the next.

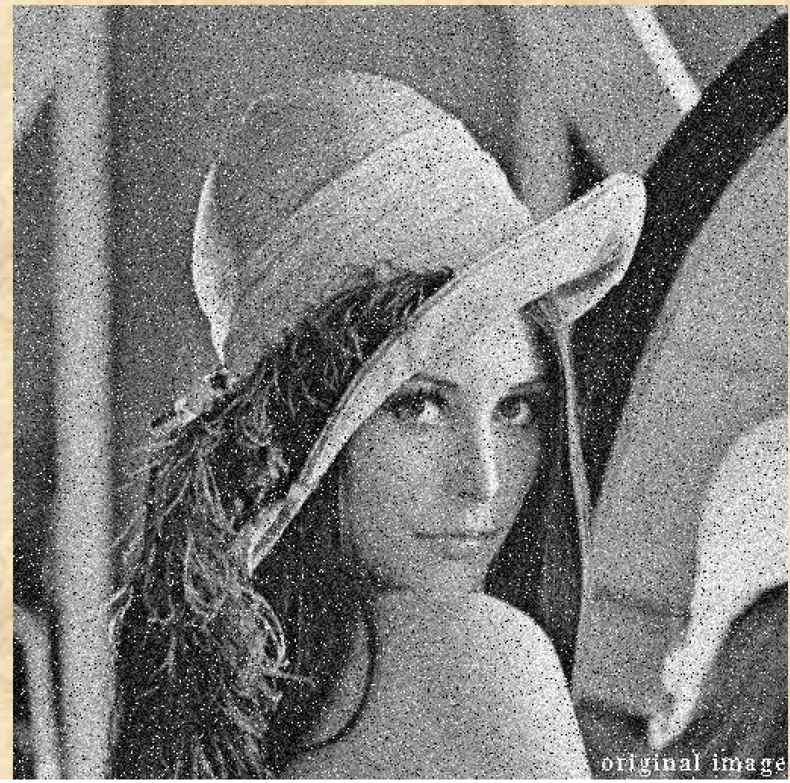
- Statistically, we expect the effects of noise to be reduced by a factor $n^{-1/2}$, if we use *n* images. One particular situation where this technique is of use, is in low lighting conditions.



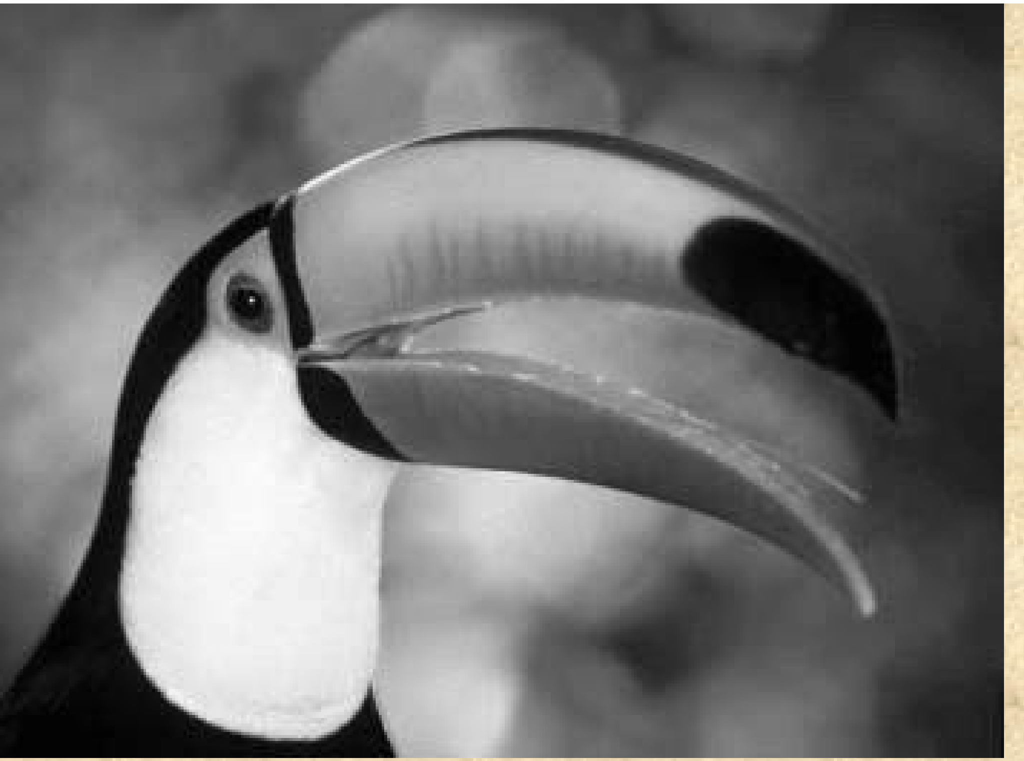
Original Image



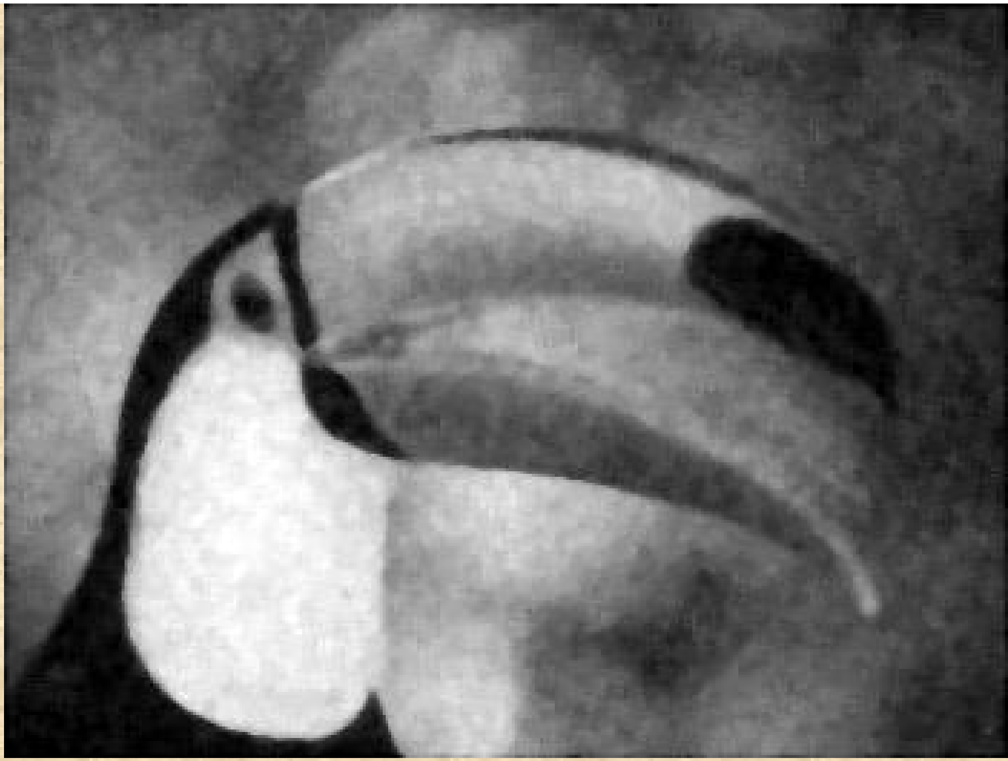
Median Filtered Image



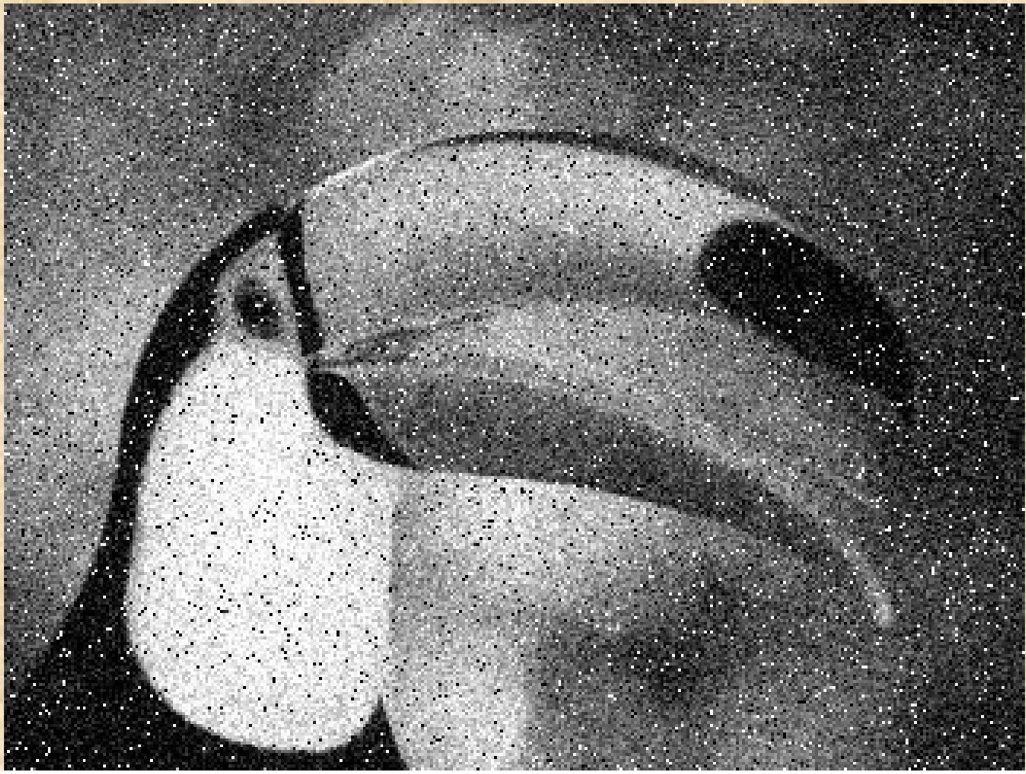
Noisy Image



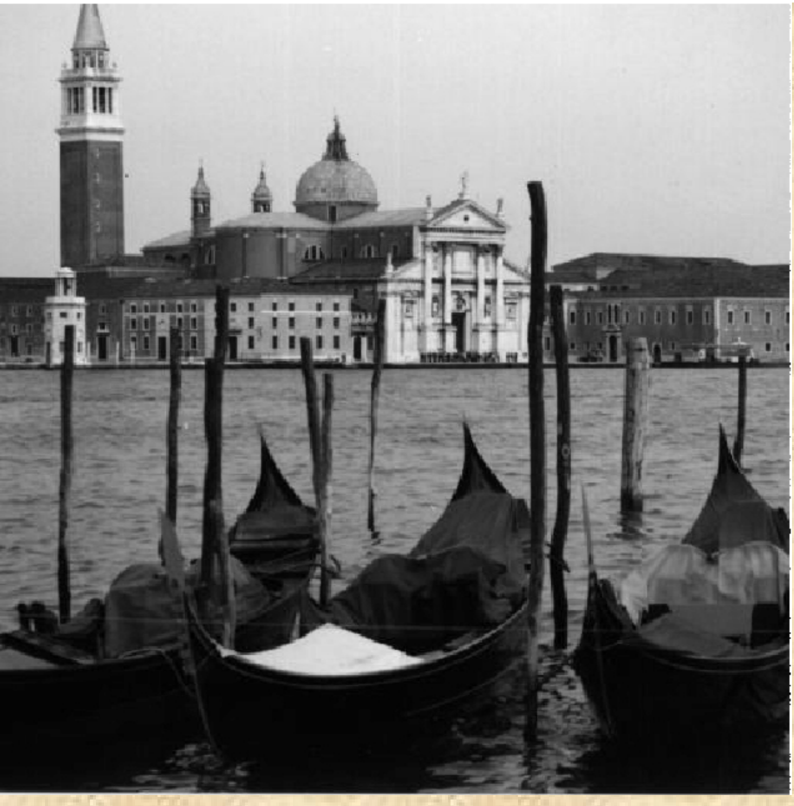
Original Image



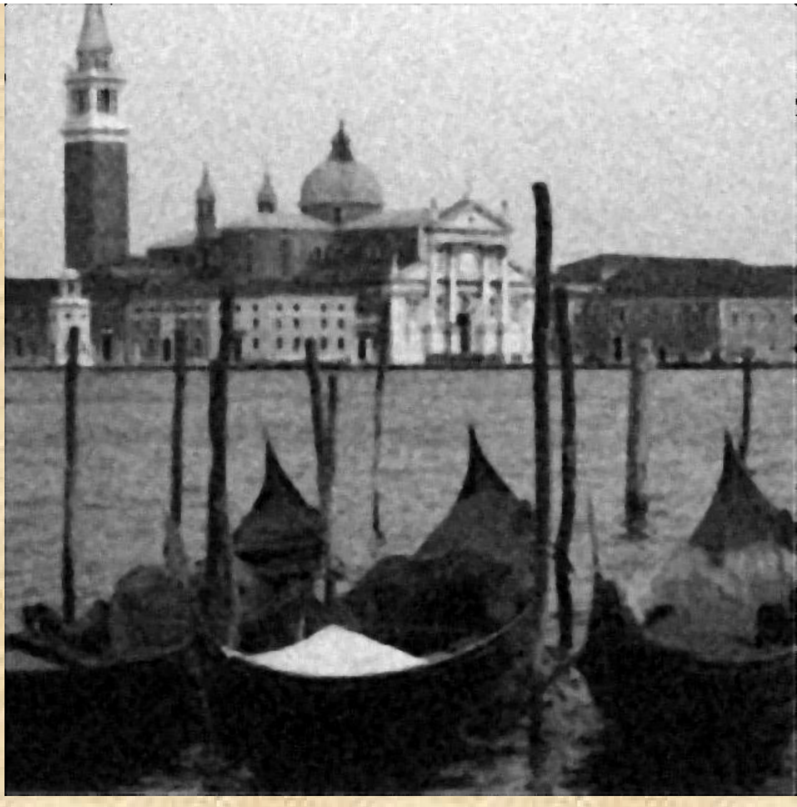
Median Filtered Image



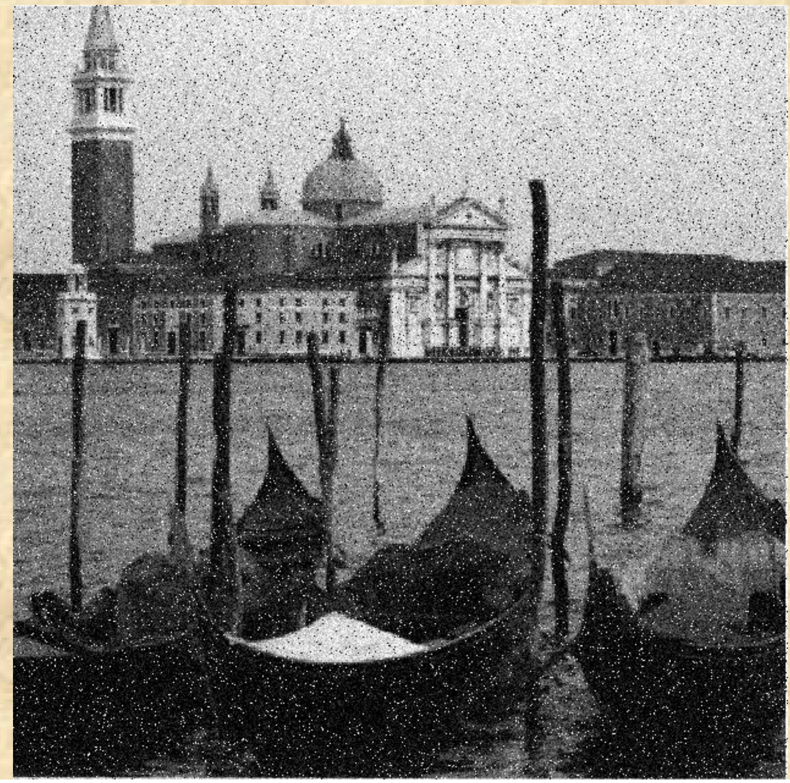
Noisy Image



Original Image

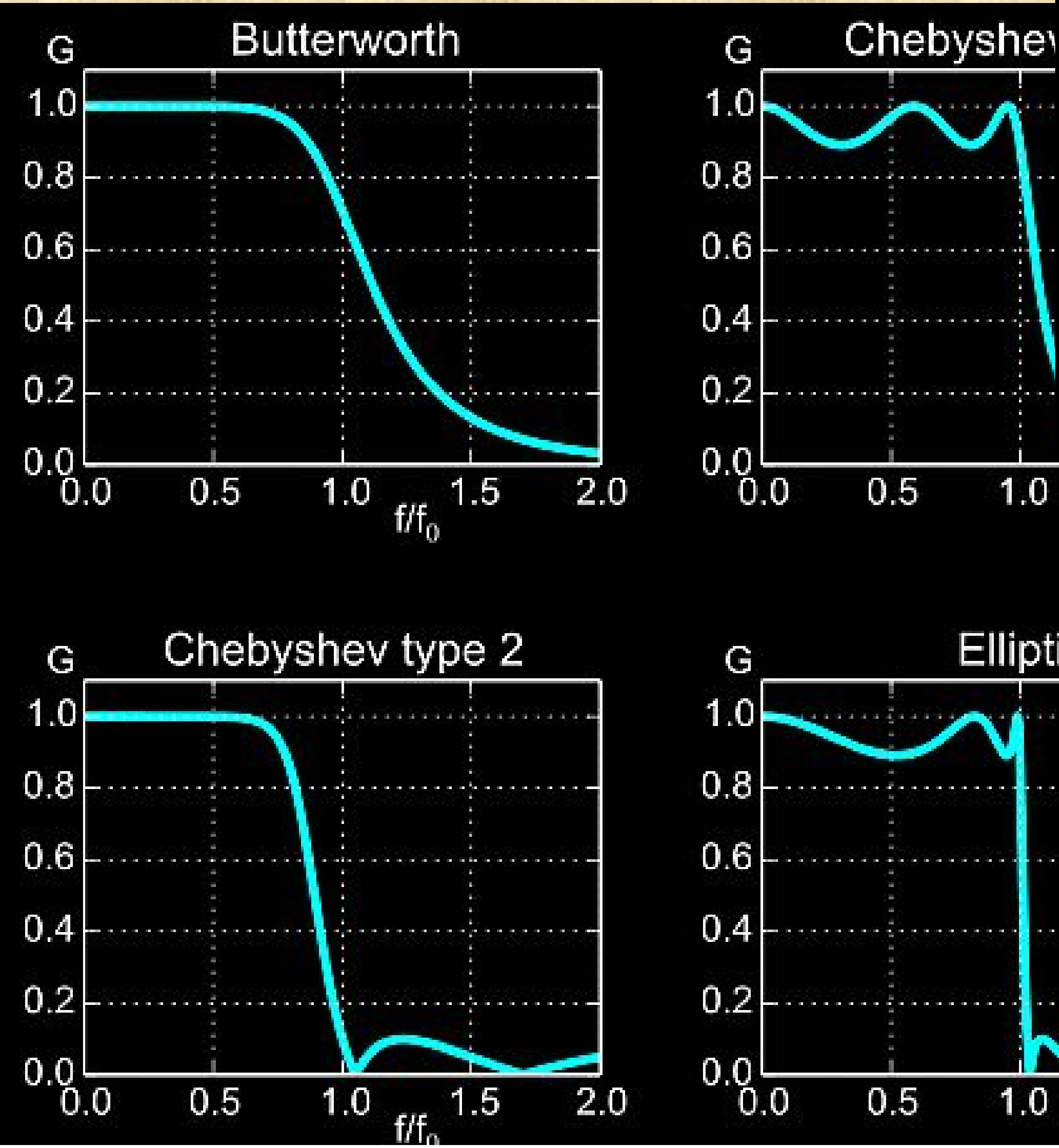


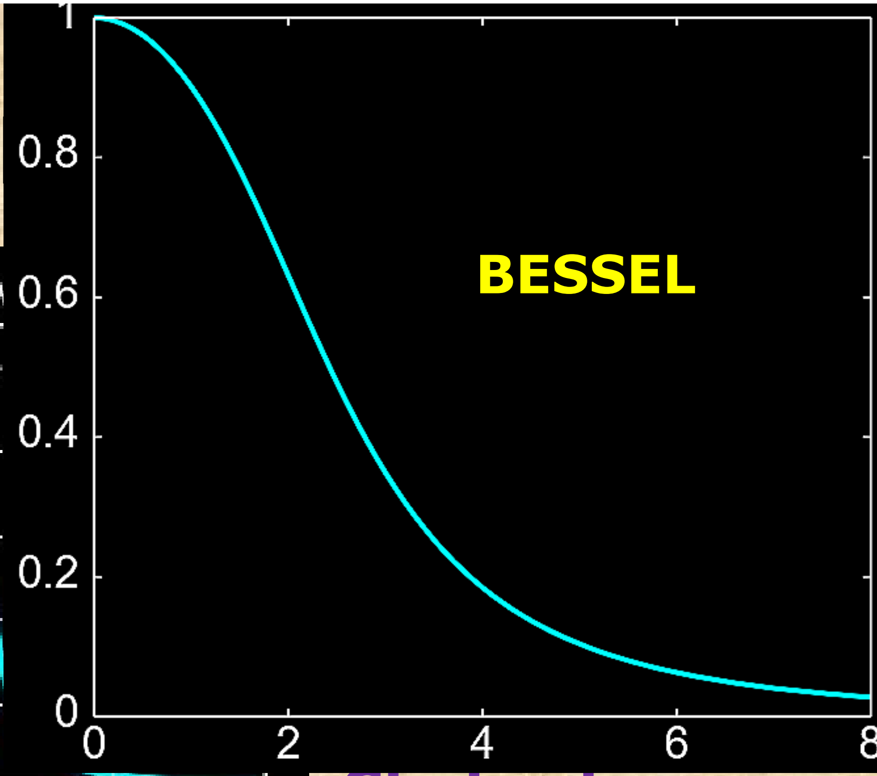
Median Filtered Image



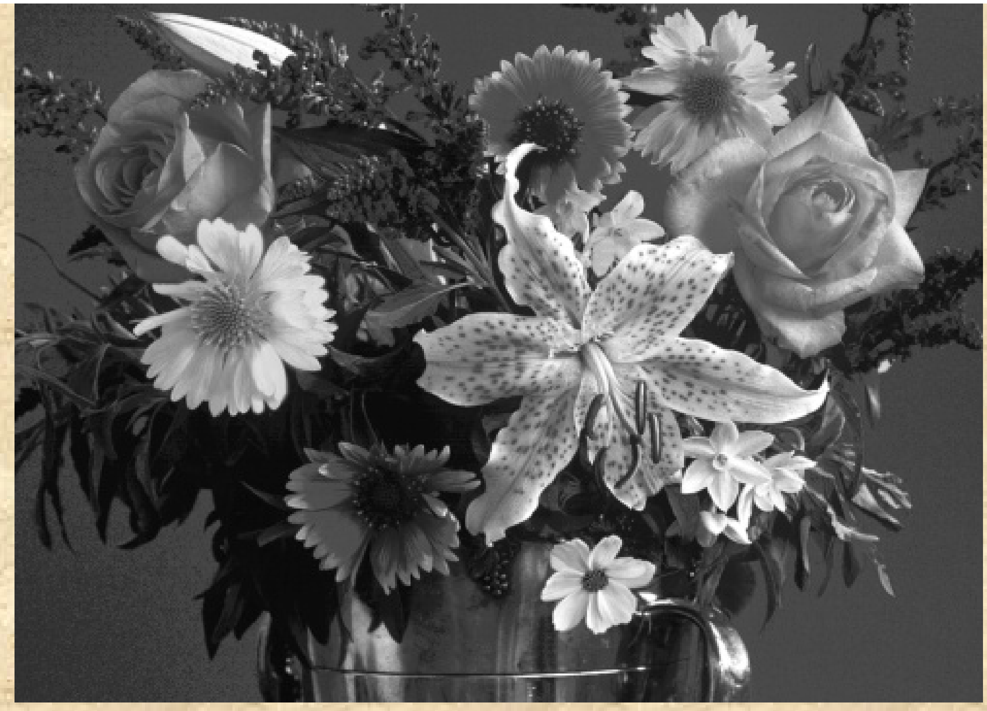
Noisy Image

Periodic noise

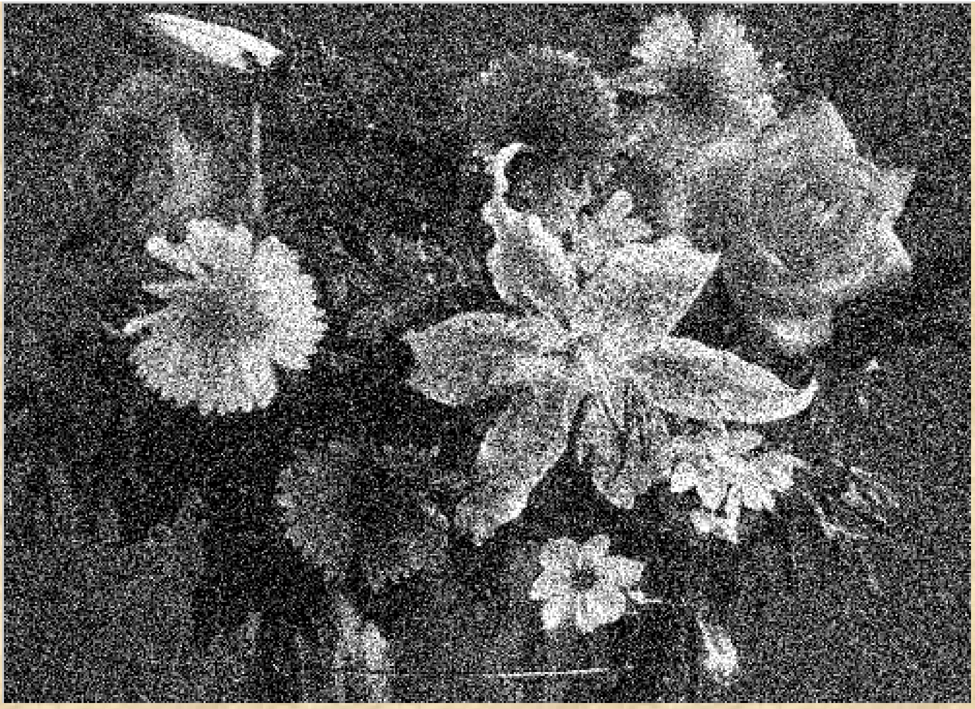




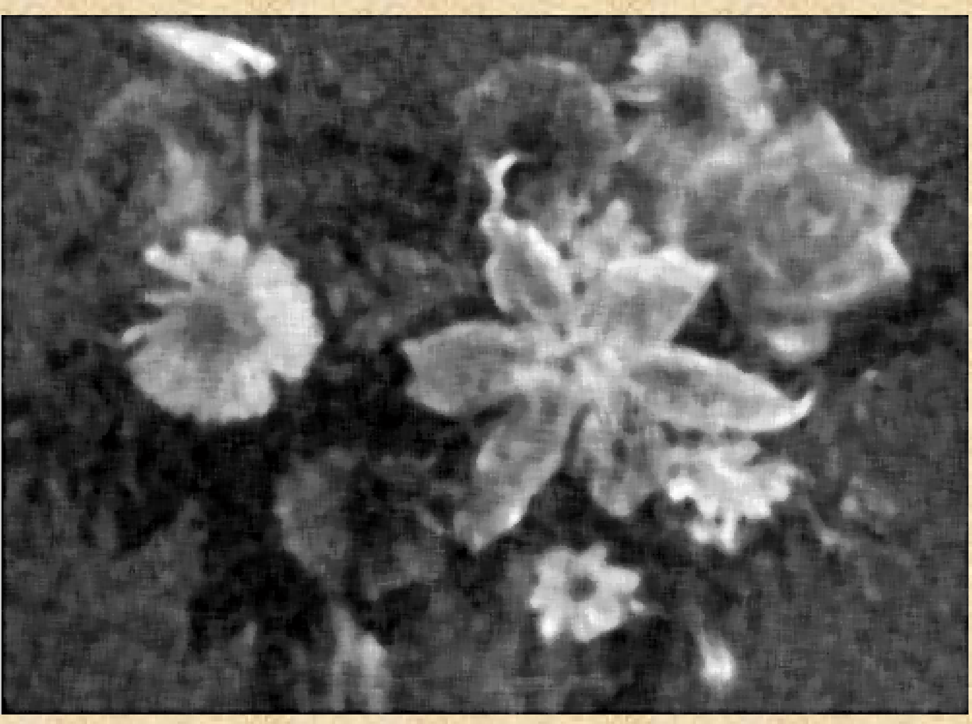
- Chebyshev
- Weiner
- Bessel
- Gaussian
- Bilateral, Box;
- Ring filters
- Gabor
- DWT, DCT



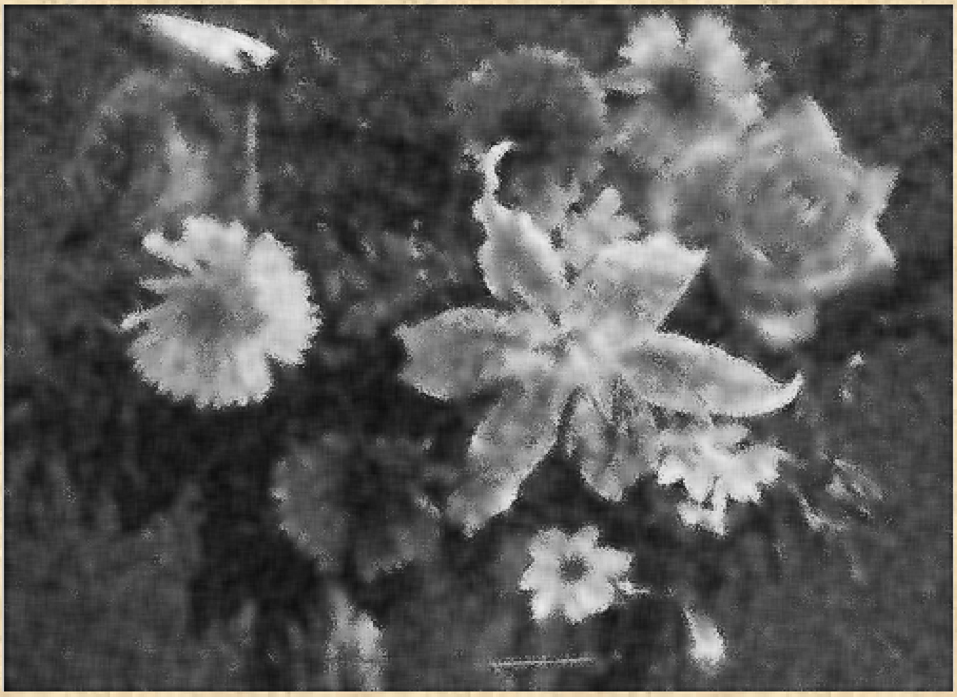
Original Image



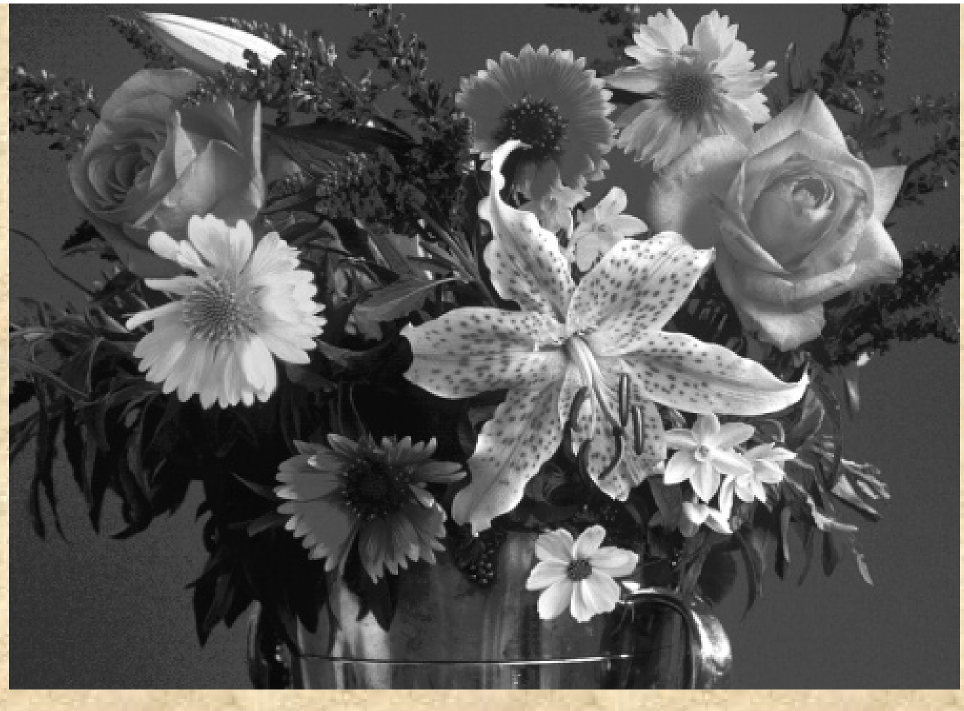
Noisy Image



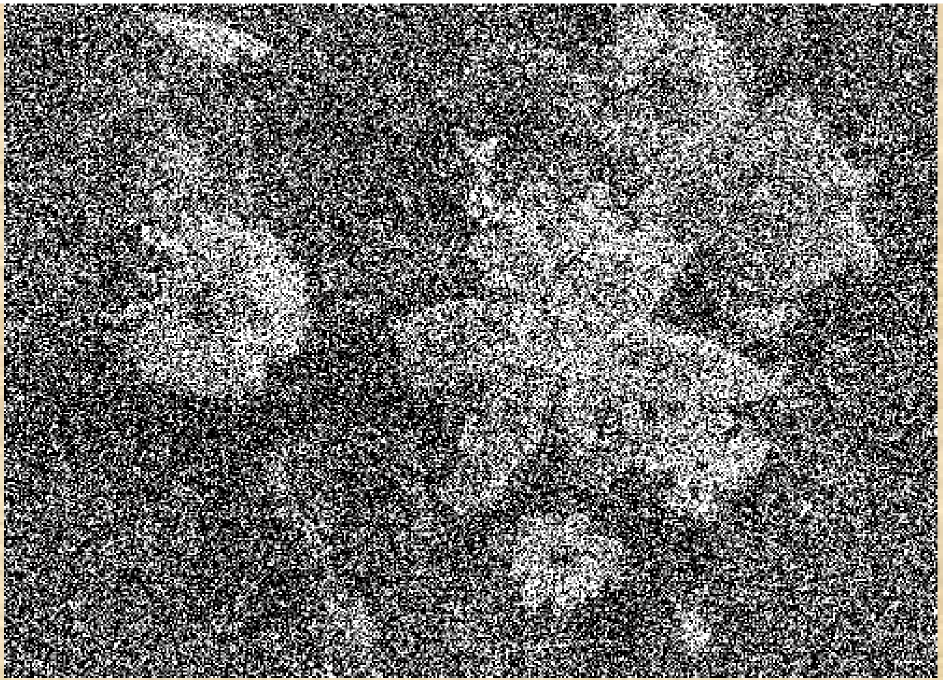
Median Filtered Image



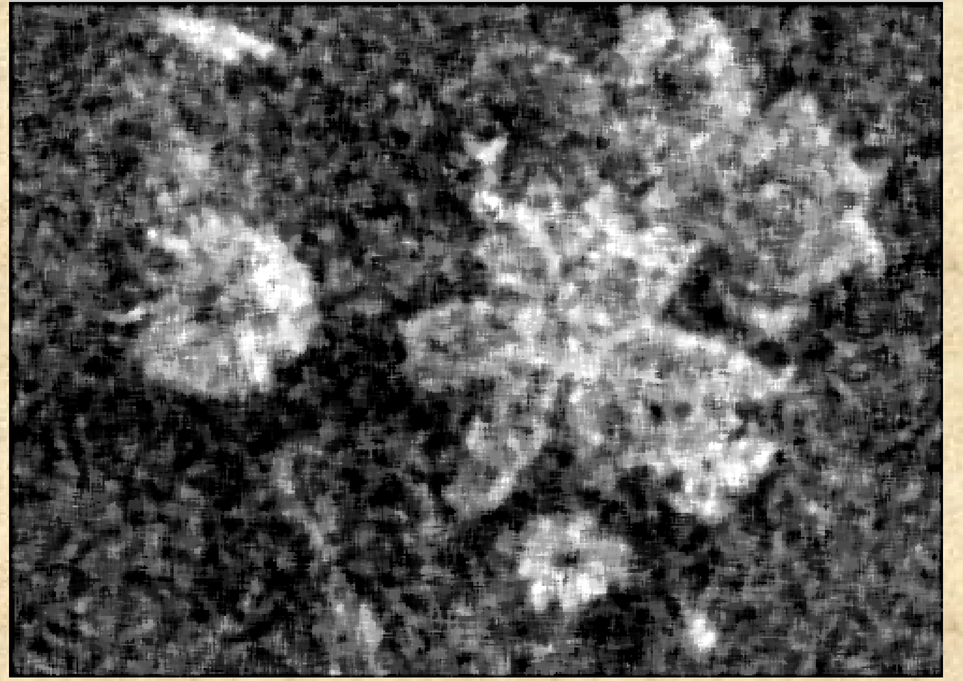
Filtered using Wiener Filter



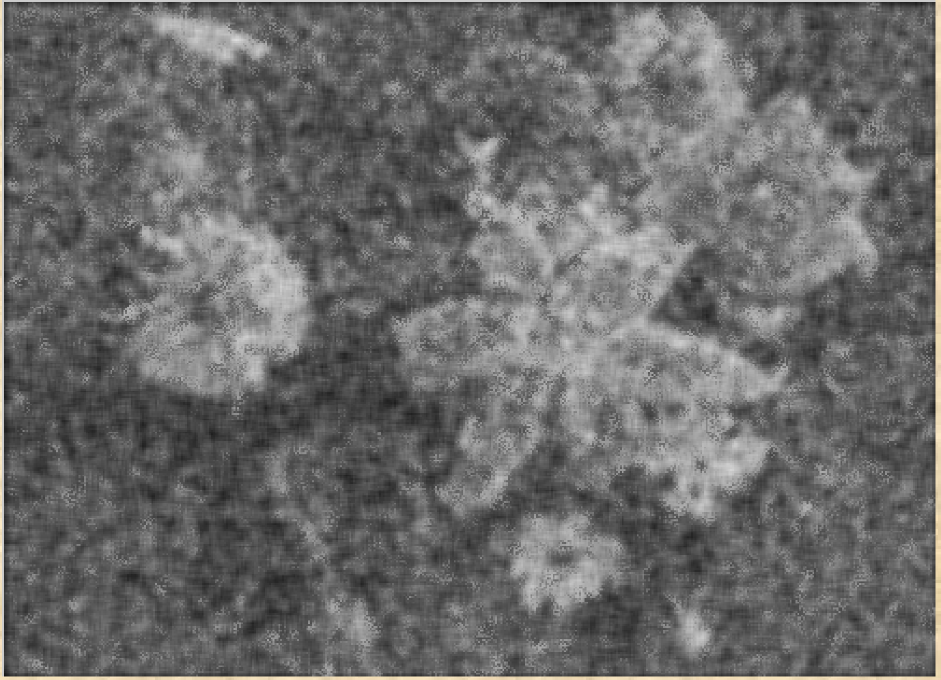
Original Image



Noisy Image



Median Filtered Image



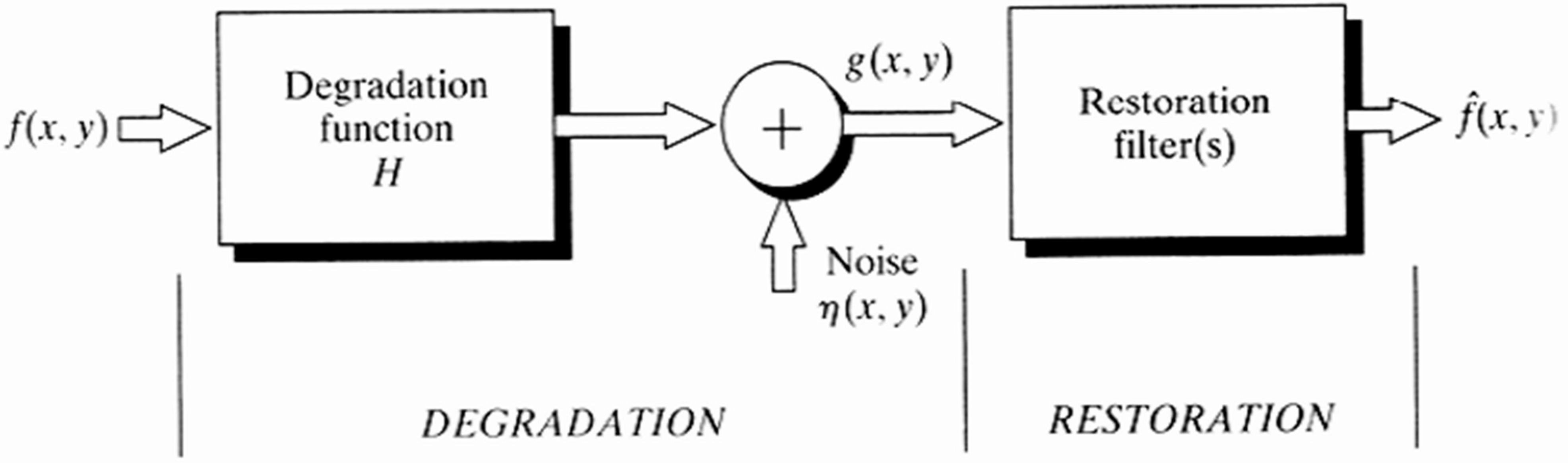
Filtered using Wiener Filter

MATHEMATICAL MODEL OF IMAGE DEGRADATION

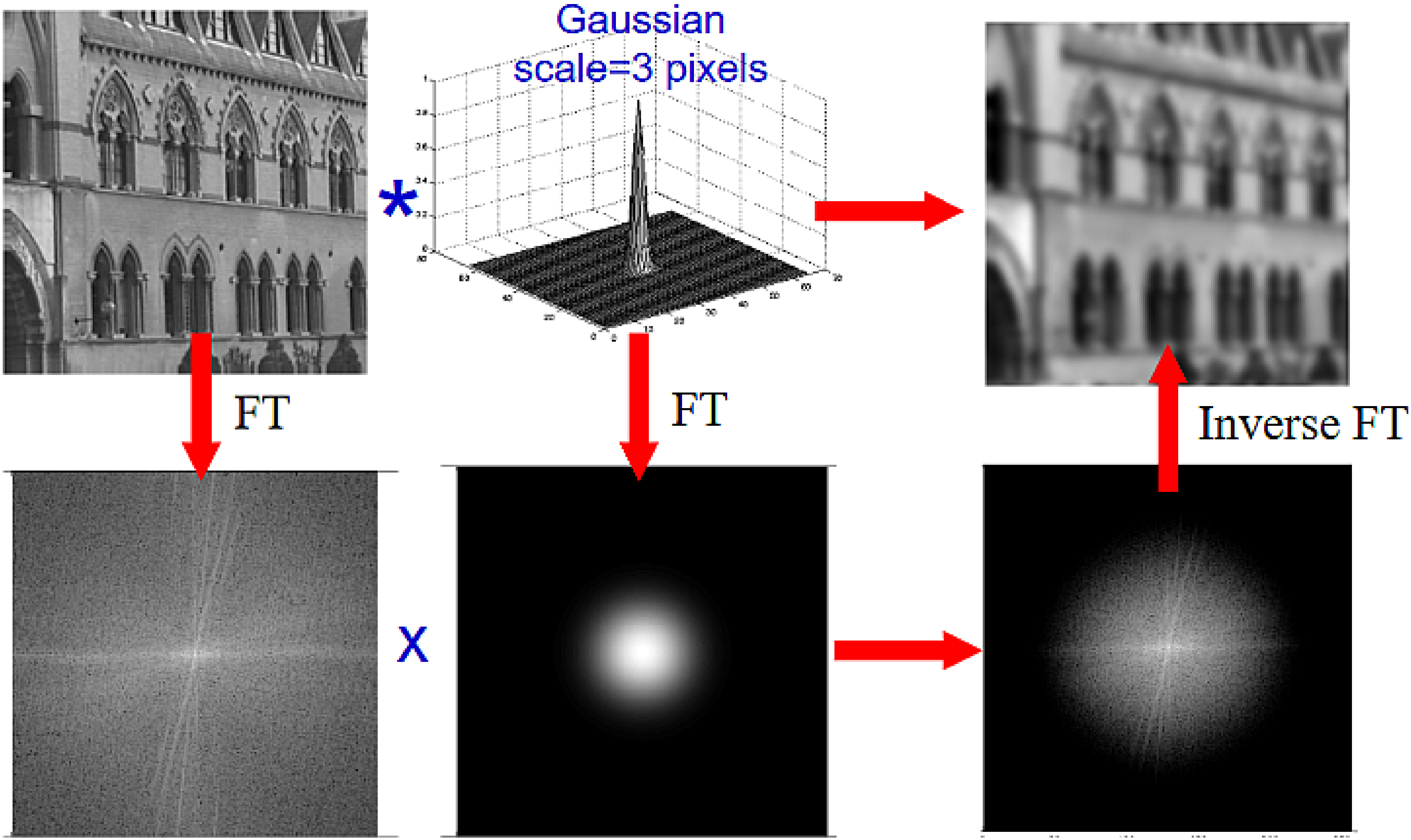
$$g(x,y) = H\{f(x,y)\} + n(x,y)$$

$$H(u,v).F(u,v) = G(u,v)$$

$$H_s(u,v) = G_s(u,v) \big/ F_s(u,v) \qquad \hat{F}(u,v) = G(u,v) \big/ H_s(u,v)$$



The challenge: loss of information and noise



Gaussian
scale=3 pixels

*

FT

FT

X

Inverse FT

Blurring acts as a low pass filter and attenuates higher spatial frequencies

$$H(u,v) = e^{-k(u^2+v^2)^{\frac{5}{6}}}$$

Obtain restoration as: $F(u,v) = H^{-1}(u,v)G(u,v)$

Minimize
(by Optimization): $\sum[g(x,y) - h(x,y) * f(x,y)]^2$

(Constrained Least Squares, Splines, TV-Reg)

Refs:
https://www.clear.rice.edu/elec431/projects95/lords/wolf.html;

https://www.ece.iastate.edu/~namrata/EE528_Spring07/ImageRestoration1.pdf

https://www.owlnet.rice.edu/~elec539/Projects99/BACH/proj2/inverse.html

https://www.robots.ox.ac.uk/~az/lectures/ia/lect3.pdf

Inverse filtering

- With the estimated degradation function H(u,v)

$$G(u,v)=F(u,v)H(u,v)+N(u,v)$$

Unknown noise

$$\Rightarrow \quad \hat{F}(u,v)=\frac{G(u,v)}{H(u,v)}=F(u,v)+\frac{N(u,v)}{H(u,v)}$$

Estimate of original image

Problem: 0 or small values

Sol: limit the frequency around the origin

Psuedo- Inverse Filter

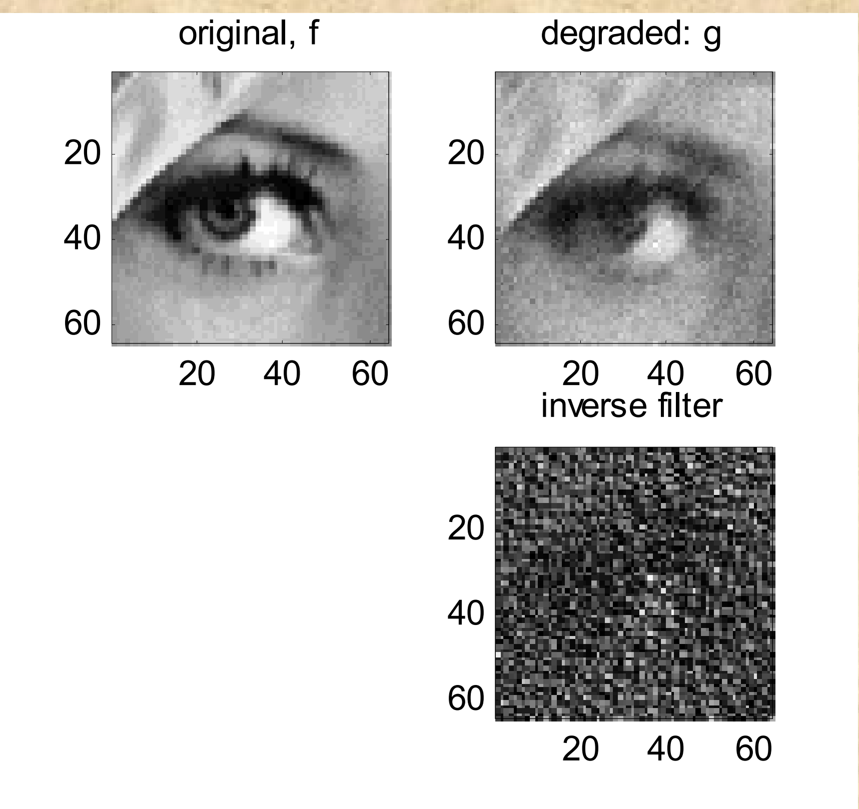
Given H(u,v), one may directly estimate the original image by

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

At (u,v) where H(u,v) ≈ 0, the noise N(u,v) term will be amplified!

$$\hat{F}(u,v) = G(u,v)/H(u,v)$$

$$= F(u,v) + \frac{N(u,v)}{H(u,v)}$$



WEINER FILTER

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right]G(u,v)$$

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right]G(u,v)$$

Where K is a specified constant

$$= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)$$

Where H(u,v)= Degradation function

H*(u,v)=complex conjugate of H(u,v)

$| H(u,v)|^2$=H* (u,v) H(u,v)

H (u, v) is the transform of the degradation function and

G (u, v) is the transform of the degraded image.

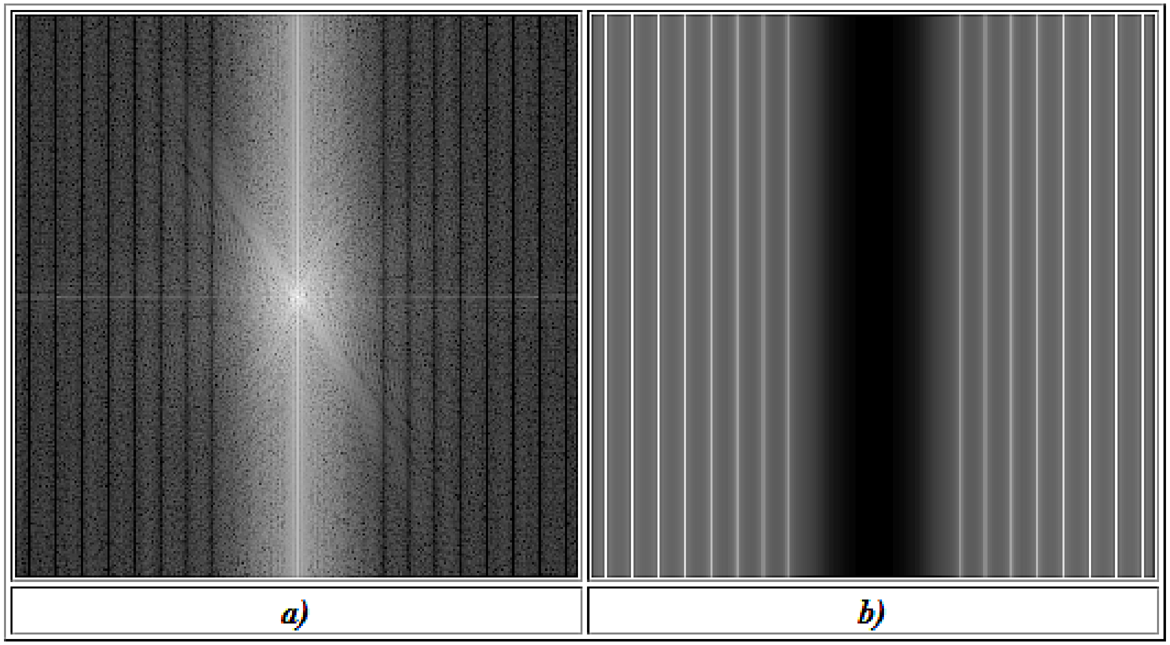
$S_f(u,v)=|F(u,v)|^2$= power spectrum of the underrated image

$S_n(u,v)=|N(u,v)|^2$= power spectrum of the noise

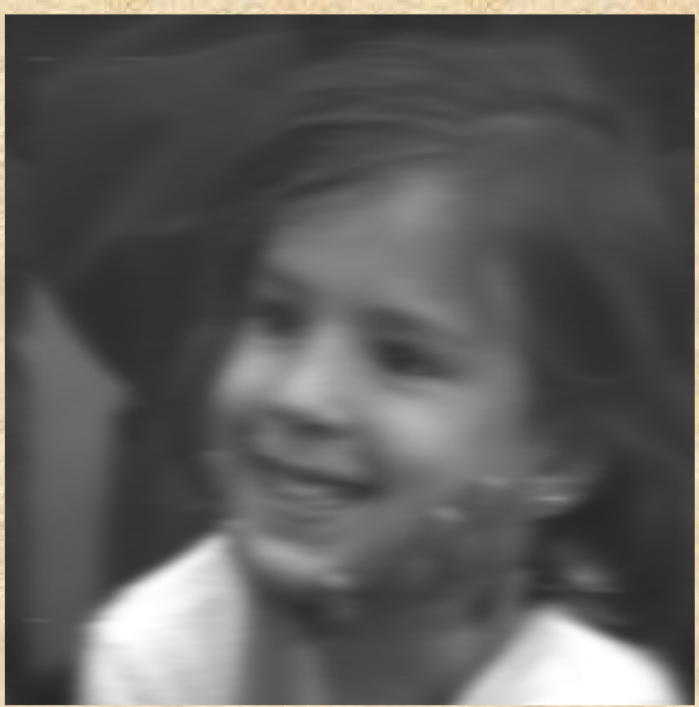
```
    P(u,v) = H(u,v) Q(u,v),
where
    P(u,v) is the degraded image,
    H(u,v) is the degradation transfer function, and
    Q(u,v) is the original image.
```

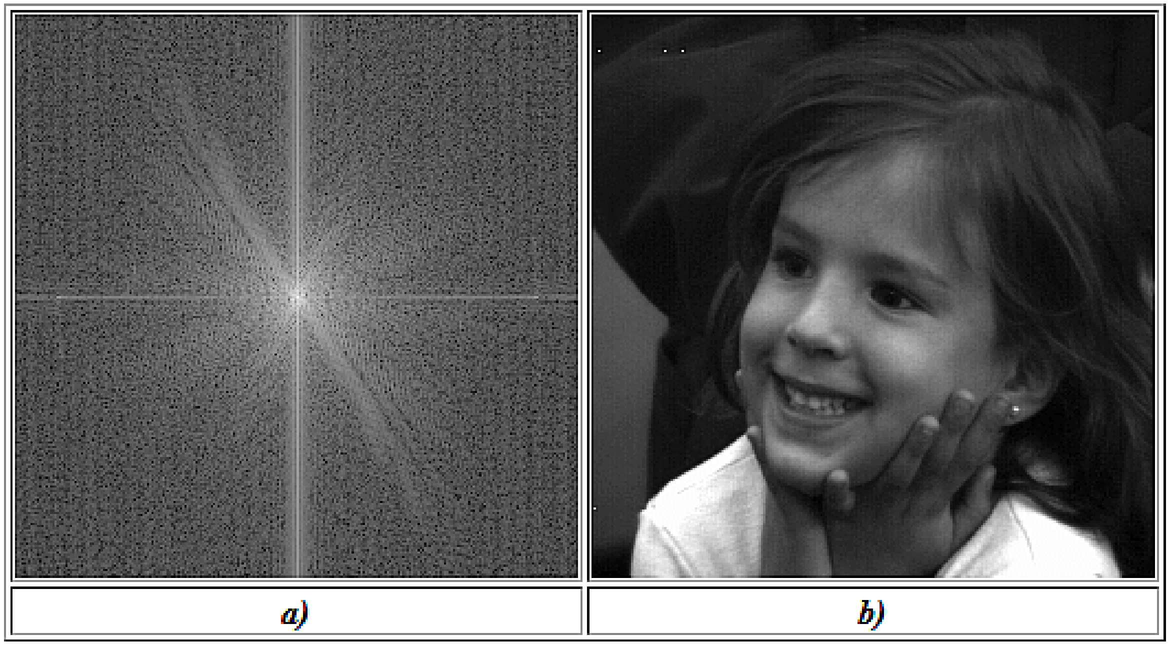
The inverse filtering process is then

```
    Q(u,v) = P(u,v) / H(u,v).
```



a)spectrum of the blurred image; b)spectrum of R(u) = 1/H(u)





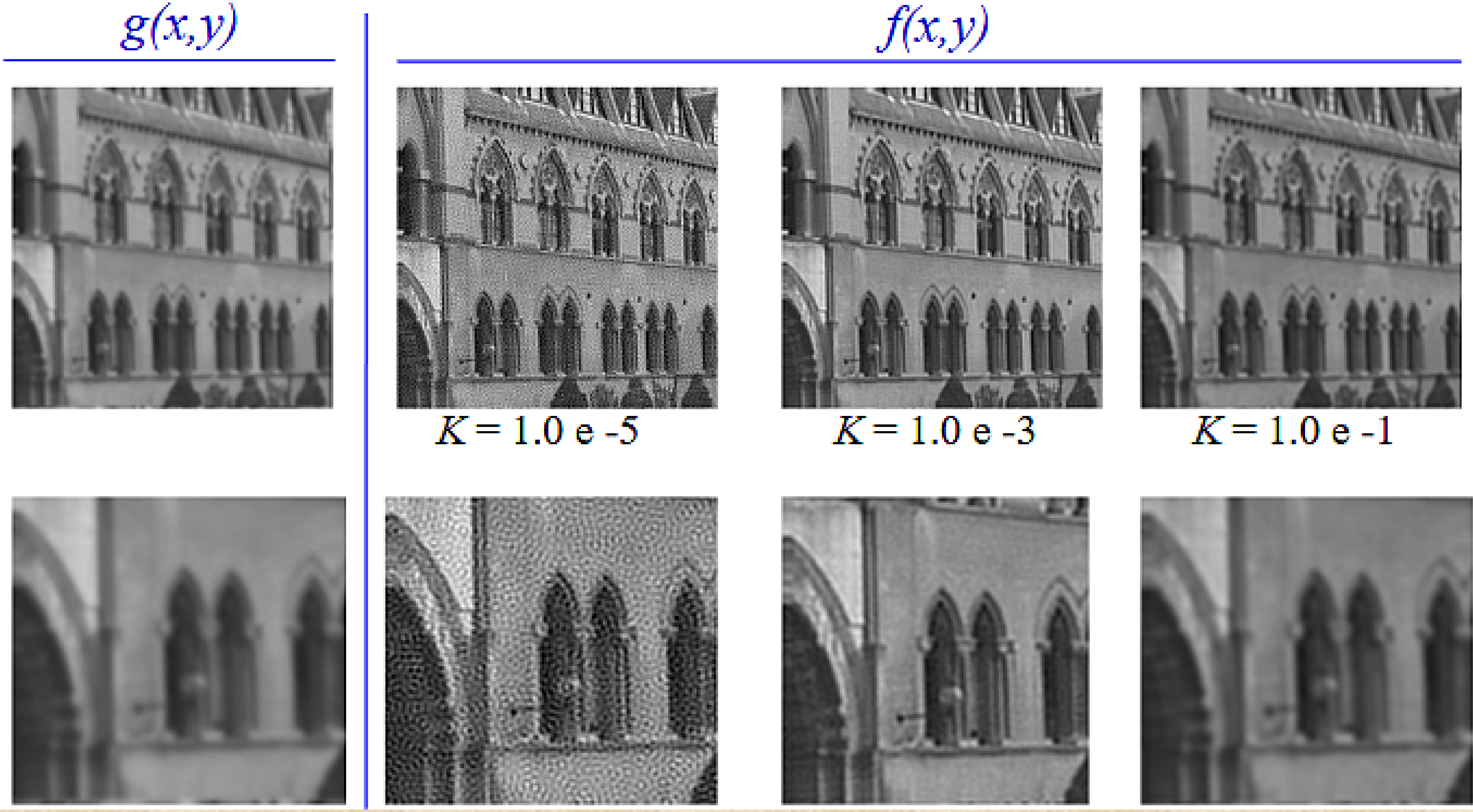
a)restored DFT; b)inverse DFT of restored DFT

Example 1: Focus deblurring with a Wiener filter

blur σ = 1.5 pixels
noise σ = 0.3 grey levels

$$\hat{F}(u,v) = W(u,v)\, G(u,v)$$

$$W(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + K(u,v)}$$



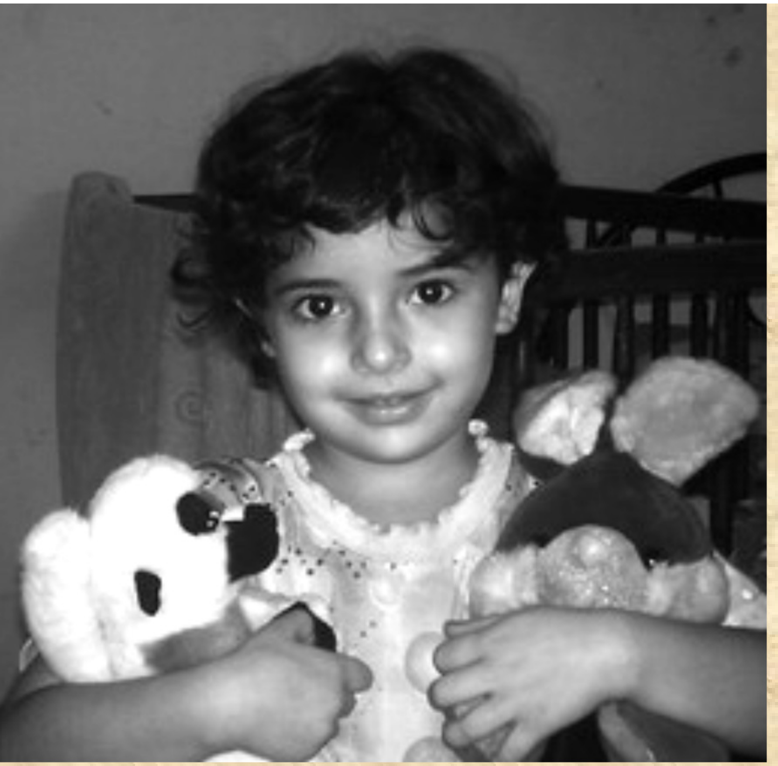
$g(x,y)$

$\hat{f}(x,y)$

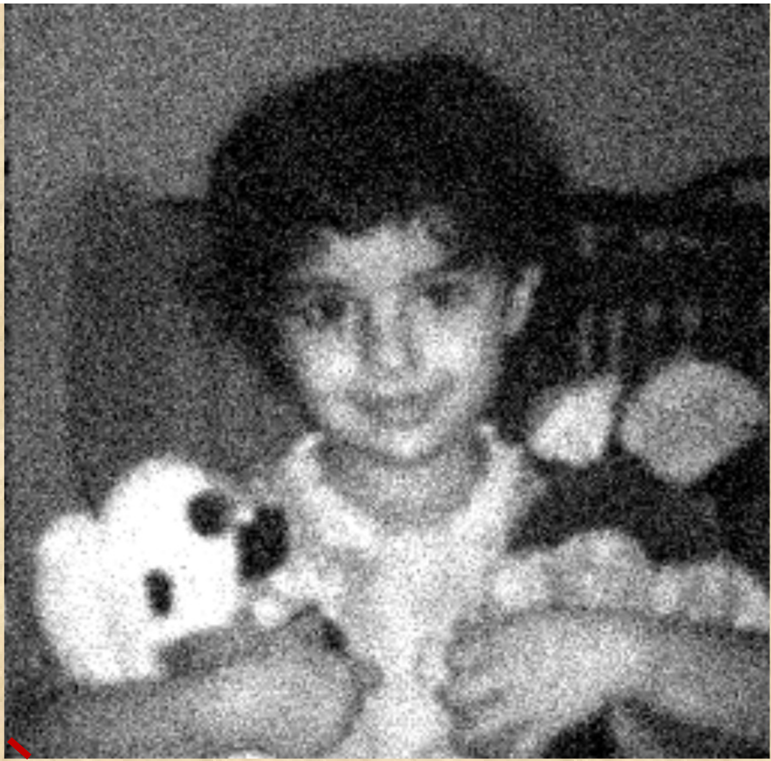
K = 1.0 e -5

K = 1.0 e -3

K = 1.0 e -1

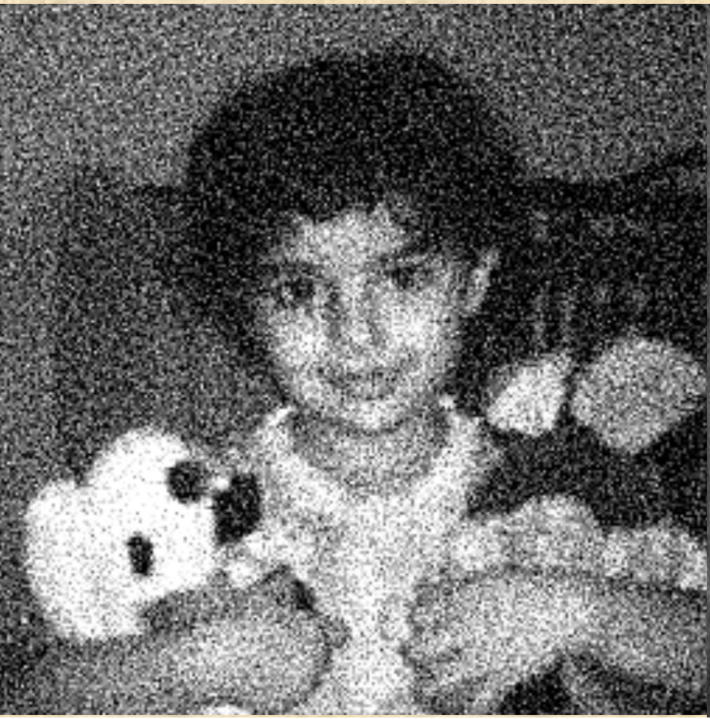


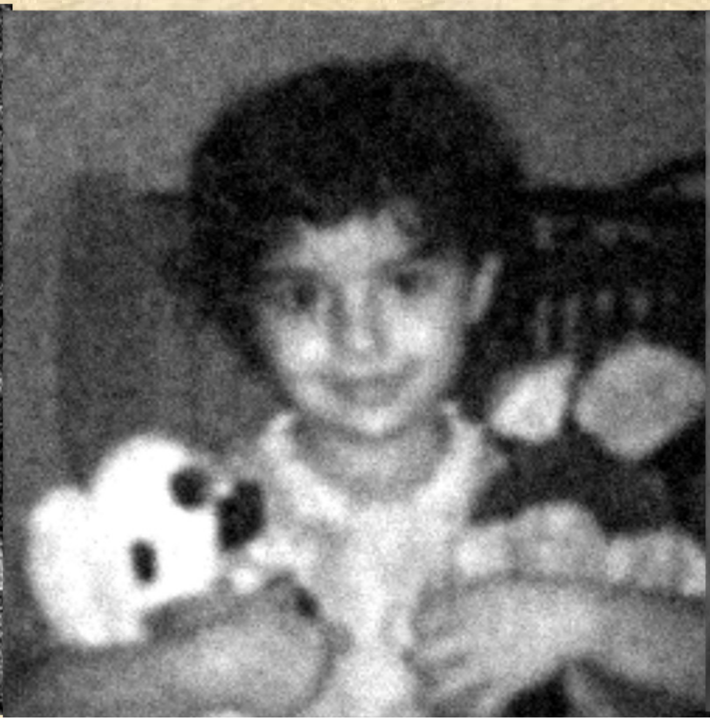
Al- Khwarizmi
Engineering
Journal,
Vol.3, No.1,
pp 48-62 (2007)

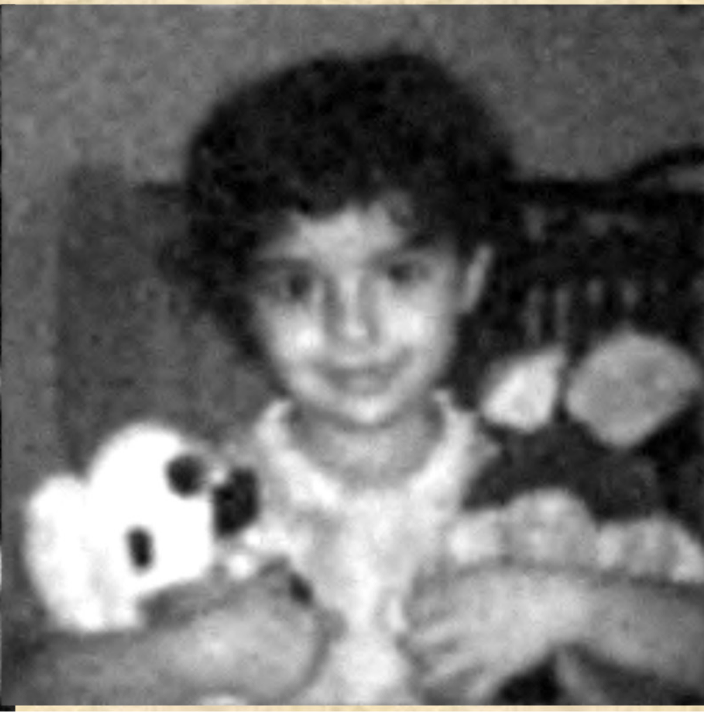


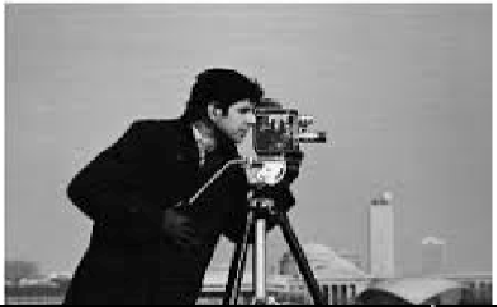
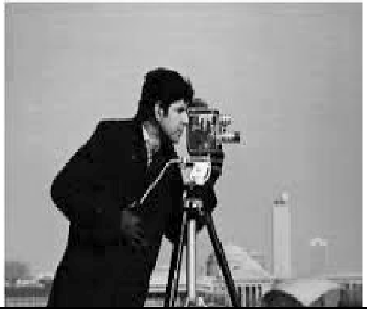
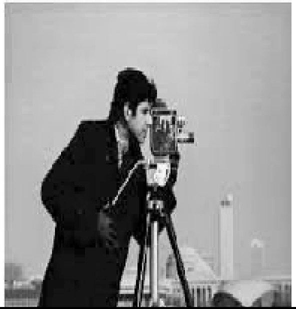
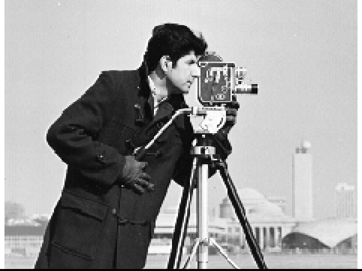
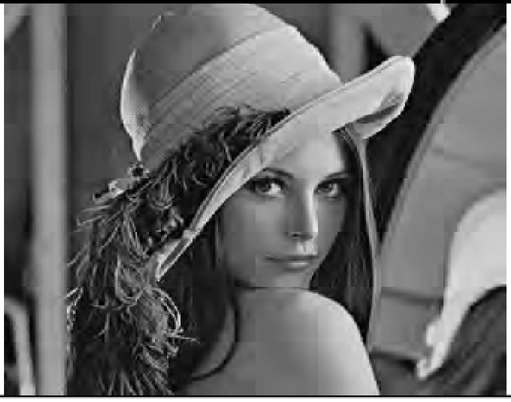
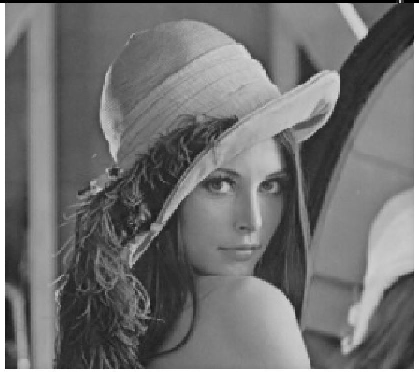
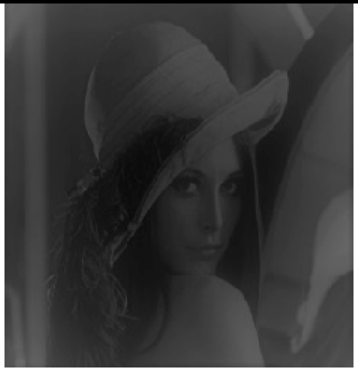
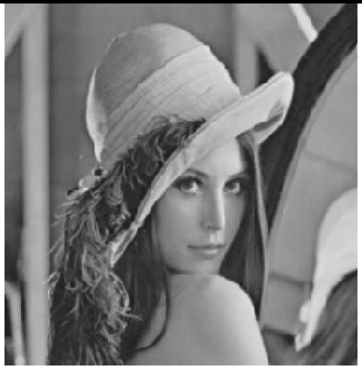
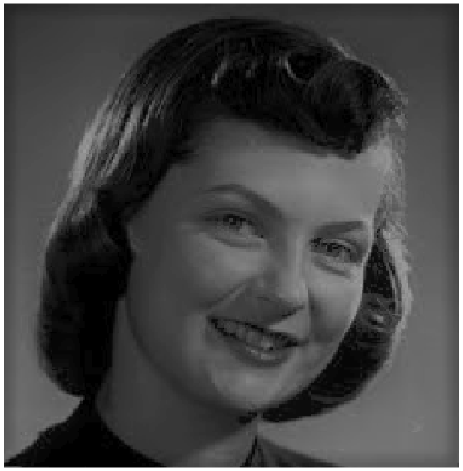
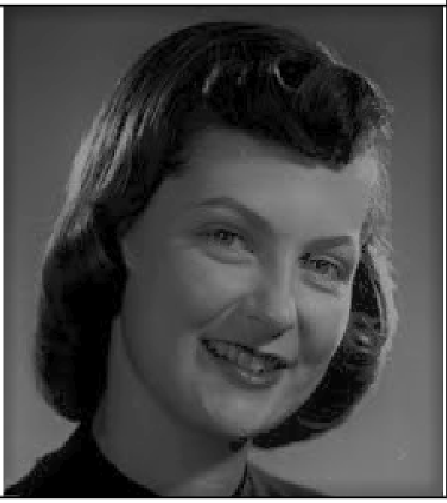
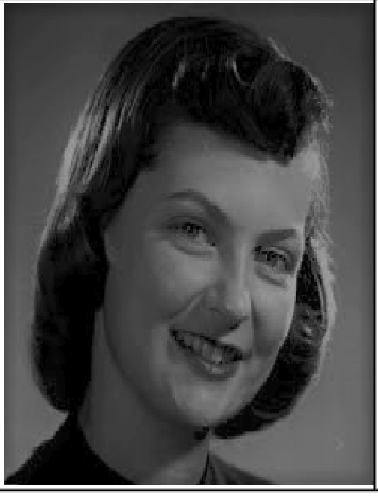
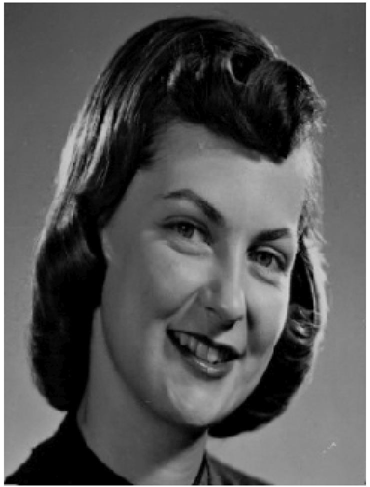
inverse filtering

Wiener filtering







Images		Original Images	Wiener filter method	Regularization Filter Method	Lucy-Richardson Filter Method
Blur Image Model	Image 1				
	Image 2				
	Image 3				

Original Images | Wiener filter method | Regularization Filter Method | Lucy-Richardson Filter Method

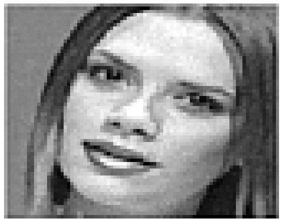
Fig. 1. (a)-(c) Blur Image Model showing Original image, Wiener filter, Regularization Filter and Lucy-Richardson Filter.

Other types of **degradations**, typically present in images/videos:

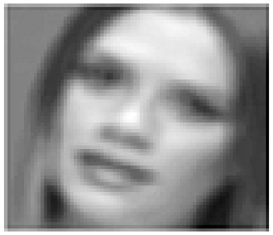
- **Low contrast**
- **Low illumination**
- **Optical distortions (more on this later)**
- **Blur : defocus, motion**
- **Scratches**
- **Low resolution**
- **Ghosts**
- **Fog/mist; atmospheric conditions**
- **Glare**
- **Degraded ink or Material (due to aging, lack of proper preservation etc.) being imaged**
- **Ice/water/dirt on lens**
- **Color Saturation, fading, diffusion,** ….
- …..

These could be spatially varying, or even exist as combinations.

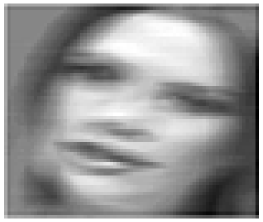
Degradations



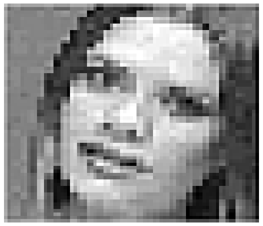
- original



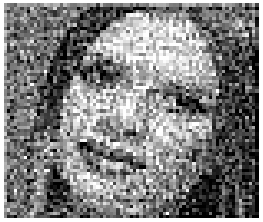
- optical blur



- motion blur



- spatial quantization (discrete pixels)

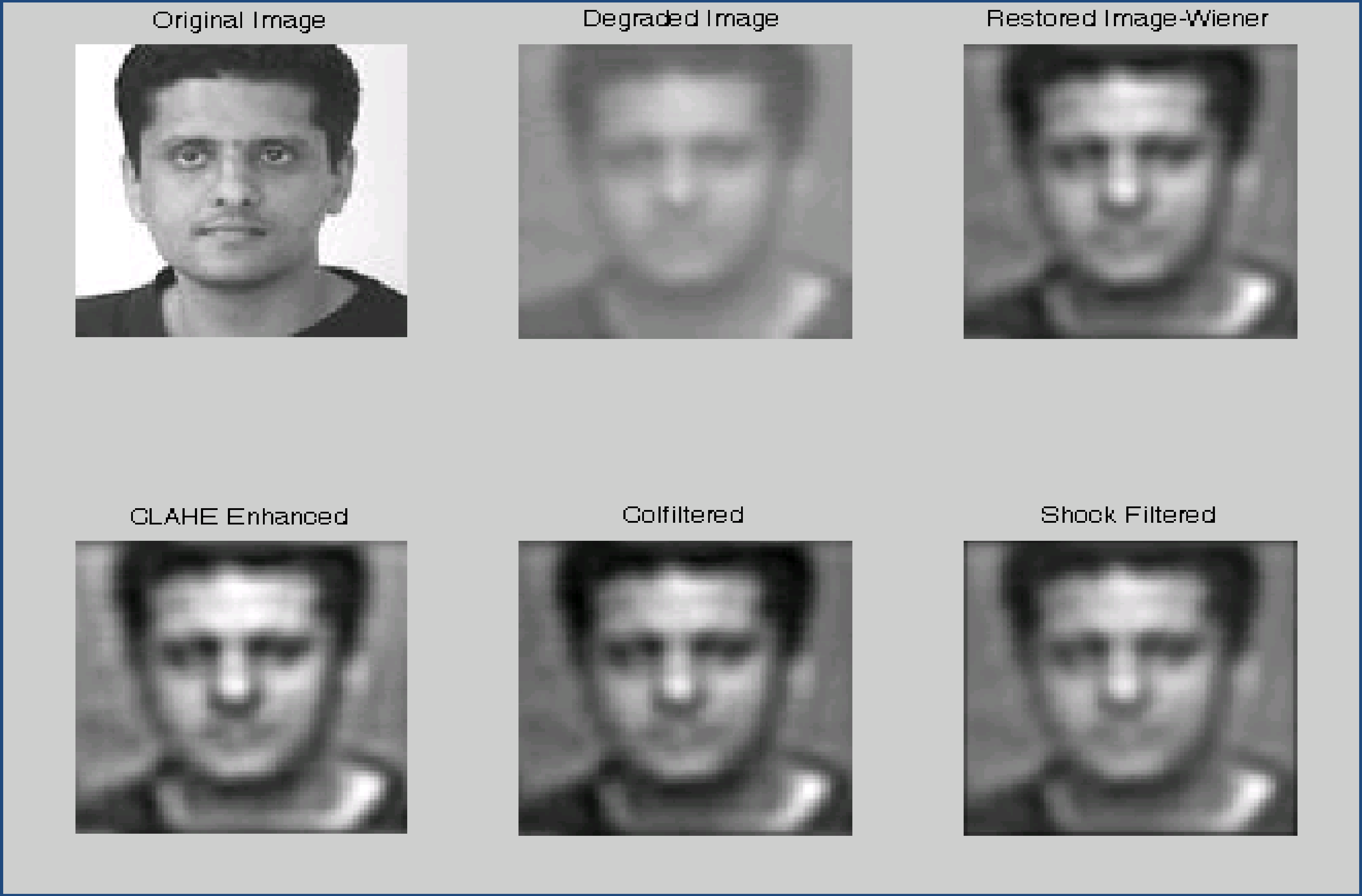


- additive intensity noise

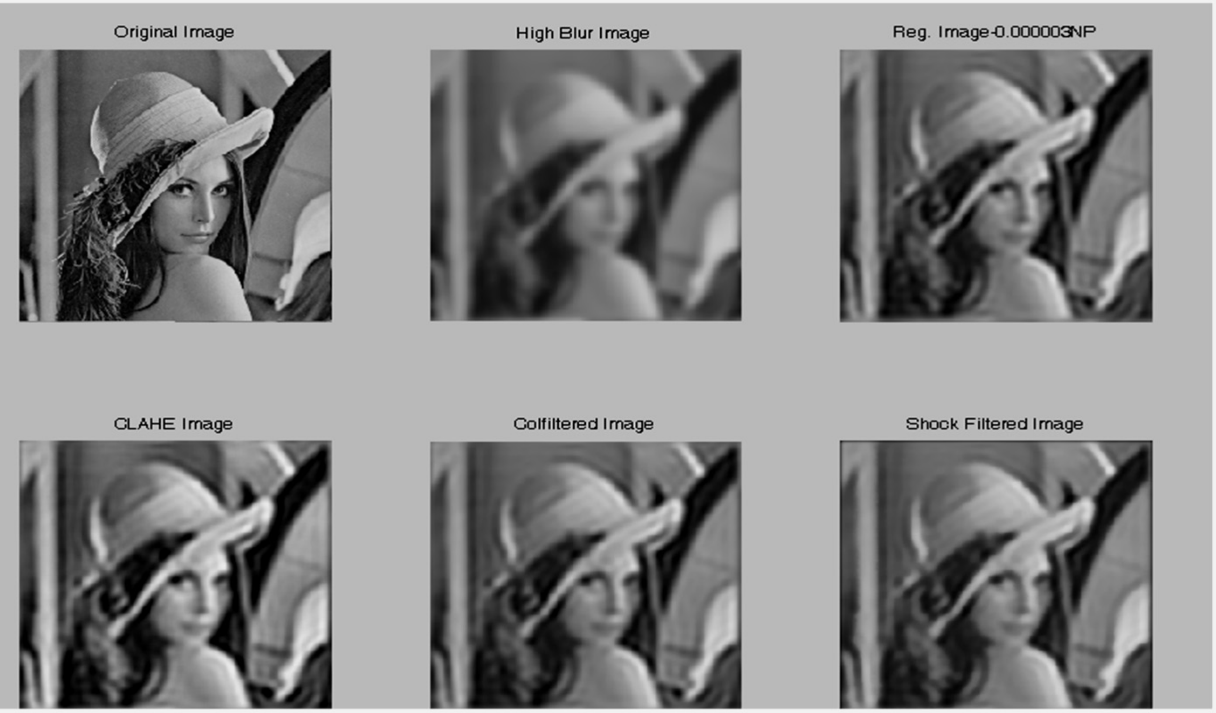
Latest (shallow) methods of Noise Removal use:

- Iterative and Adaptive Kalman Filtering
- Discrete Wavelet (multi-channel) transform
- SVD (PCA), ICA
- Fuzzy-based methods
- Optimization frameworks
- Non-linear ANNs
- Anisotropic diffusion (filtering)
- Bilateral & Homomorphic filtering
- Non-local means
- Particle Filtering
- Level Set Methods
- Basis Pursuit
- Graph-based approaches
- Stanford – DUDE
- Minimax Risk
- Manifold-based learning
- **CLAHE**
- Shock Filter
- DL based methods ??

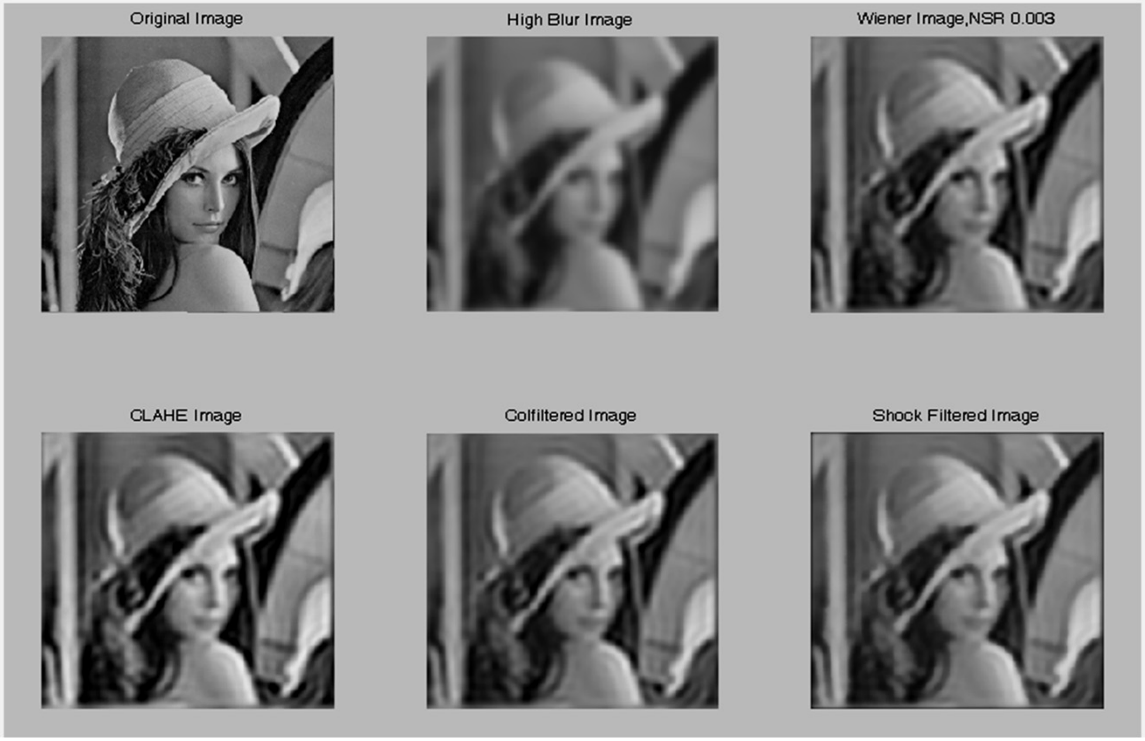
IMAGE RESTORATION (WIENER) AND ENHANCEMENT



LENA-IMAGE RESTORATION (REGULARIZATION) AND ENHANCEMENT

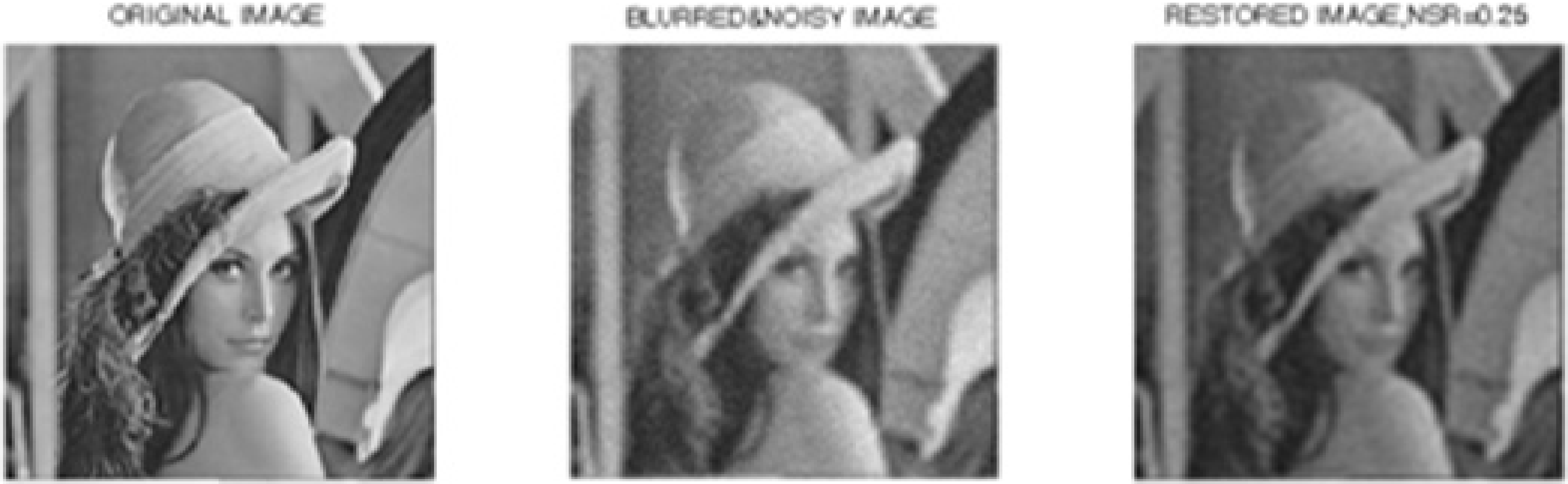


LENA-IMAGE RESTORATION (WIENER) AND ENHANCEMENT



ve histogram equalization

IMAGE RESTORATION-WIENER DECONVOLUTION

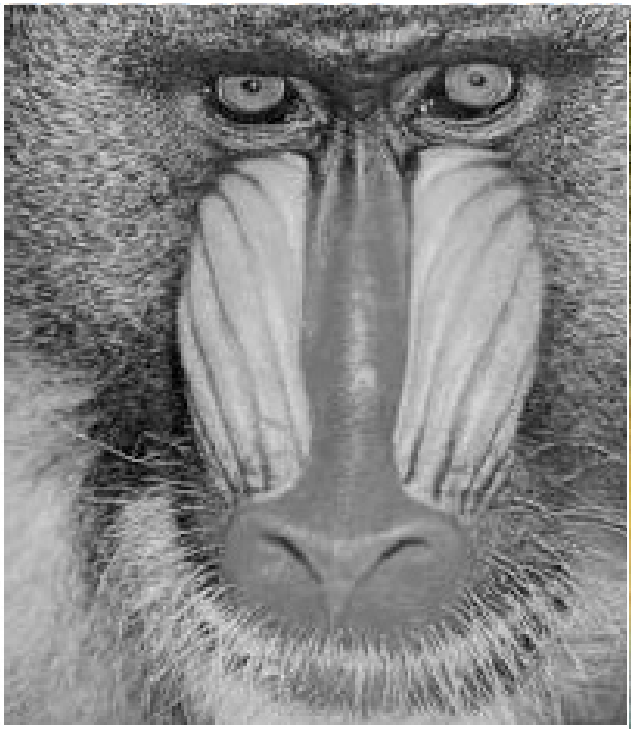


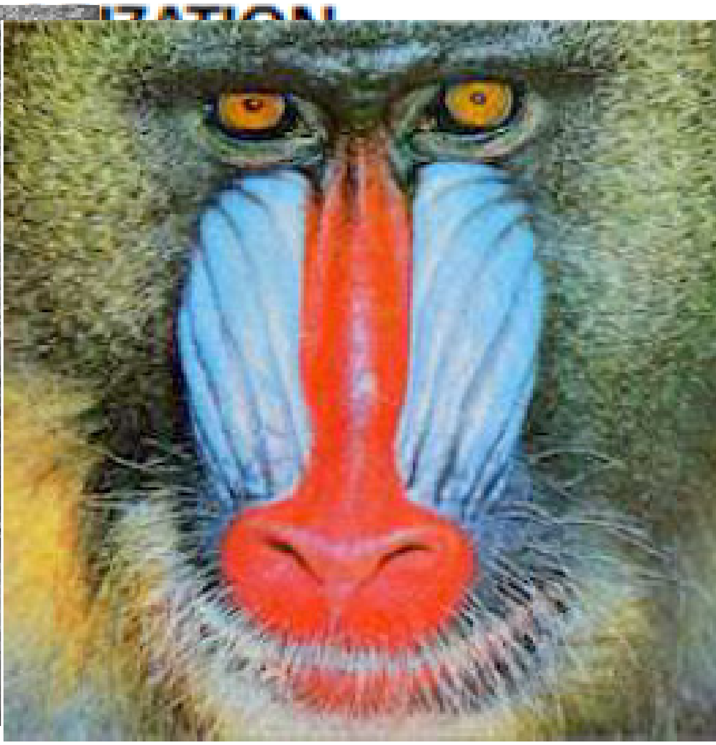
Standard Lena Image
PSNR = Infinity, MSE = Zero

Blurred Lena Image
PSNR = 23.2993, MSE = 304.1938

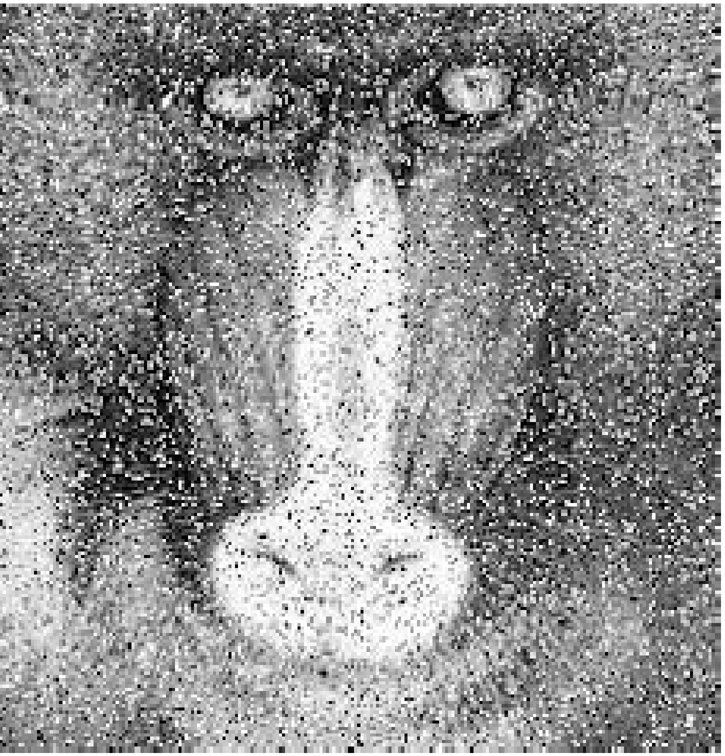
Restored Lena Image
PSNR = 19.1447, MSE = 791.7906

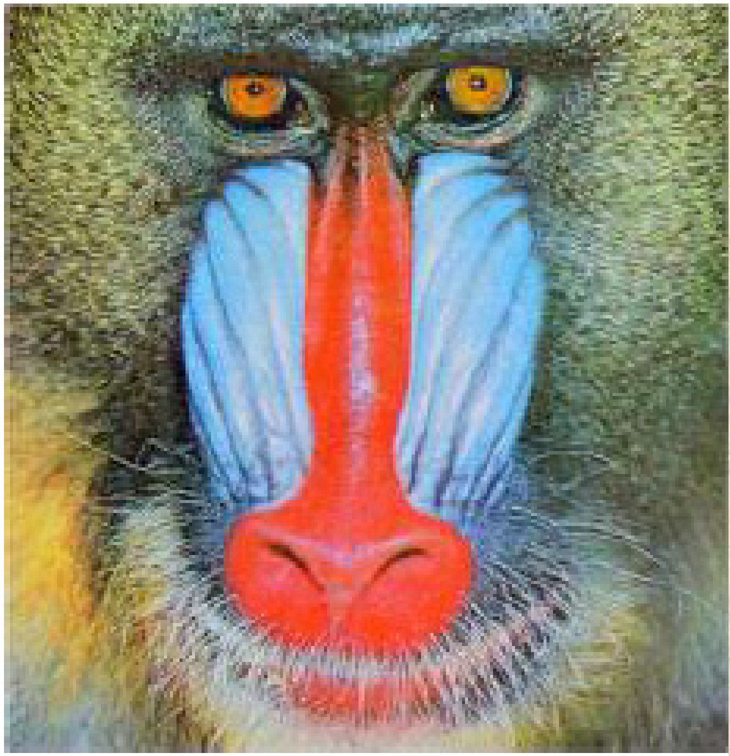
ZATION





PSNR=22.97
SSIM=0.6550

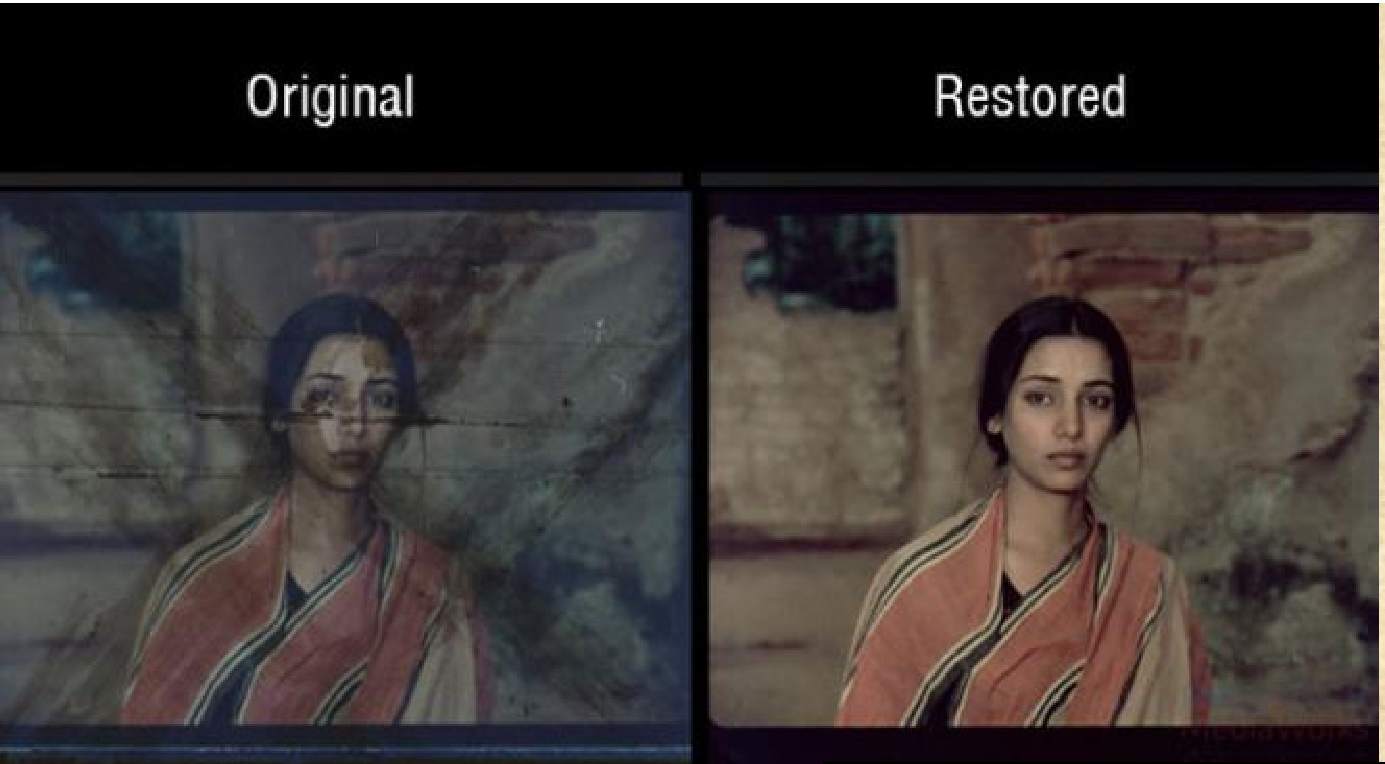


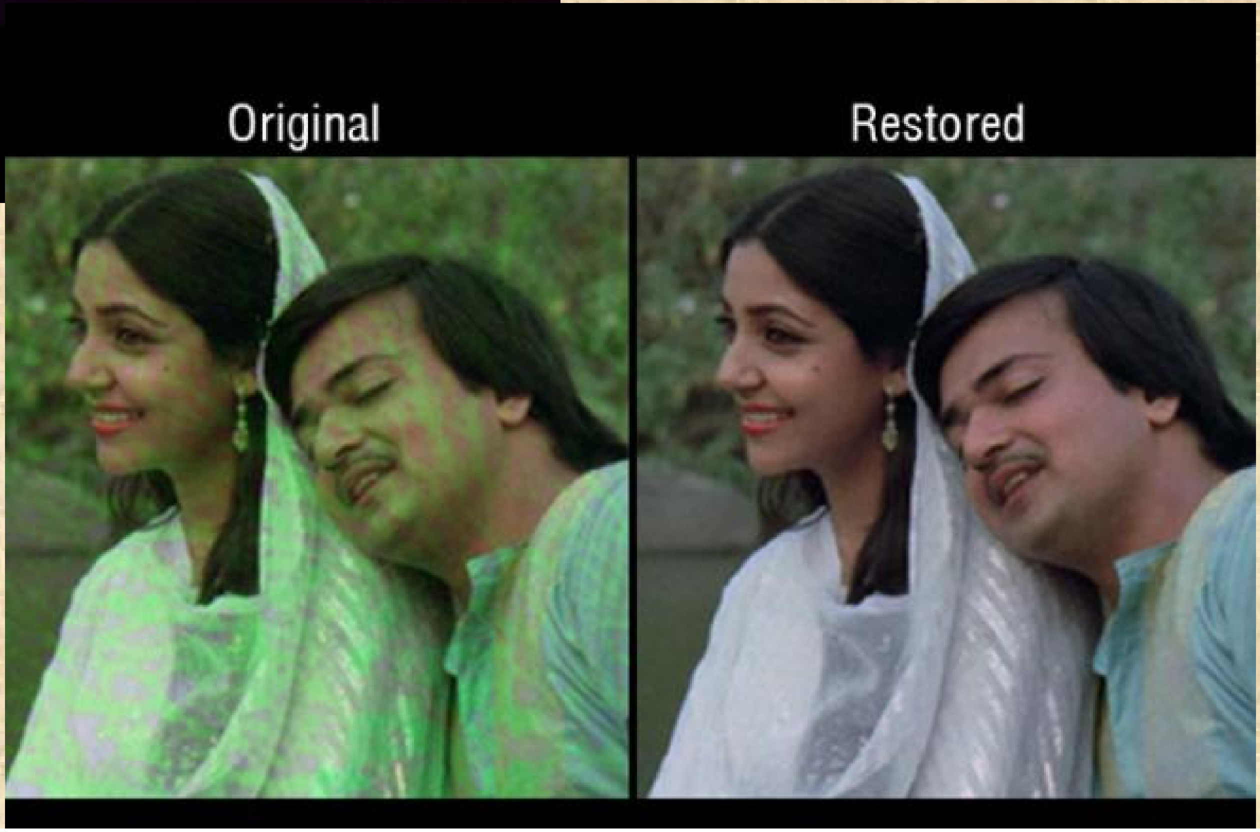


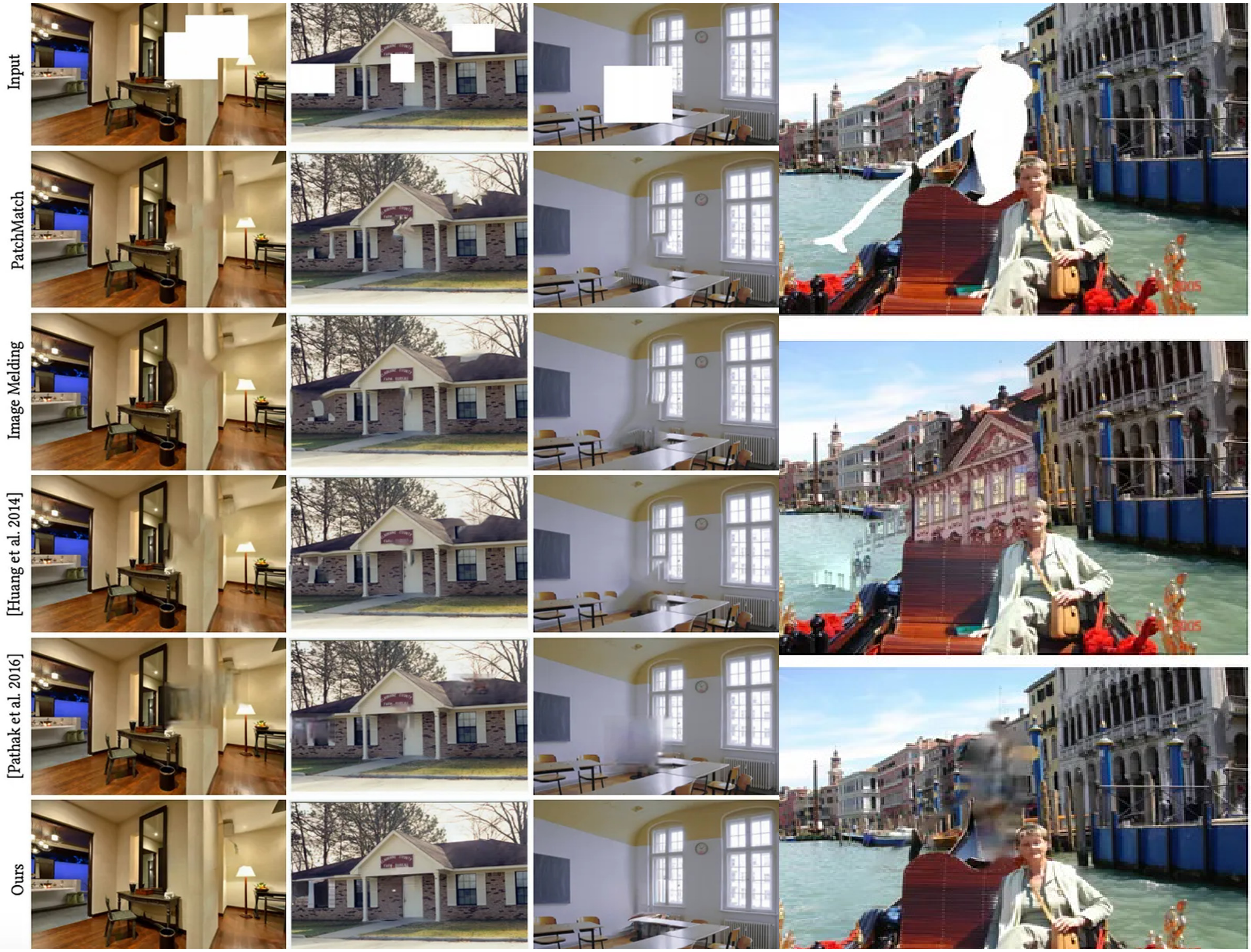
Weiner

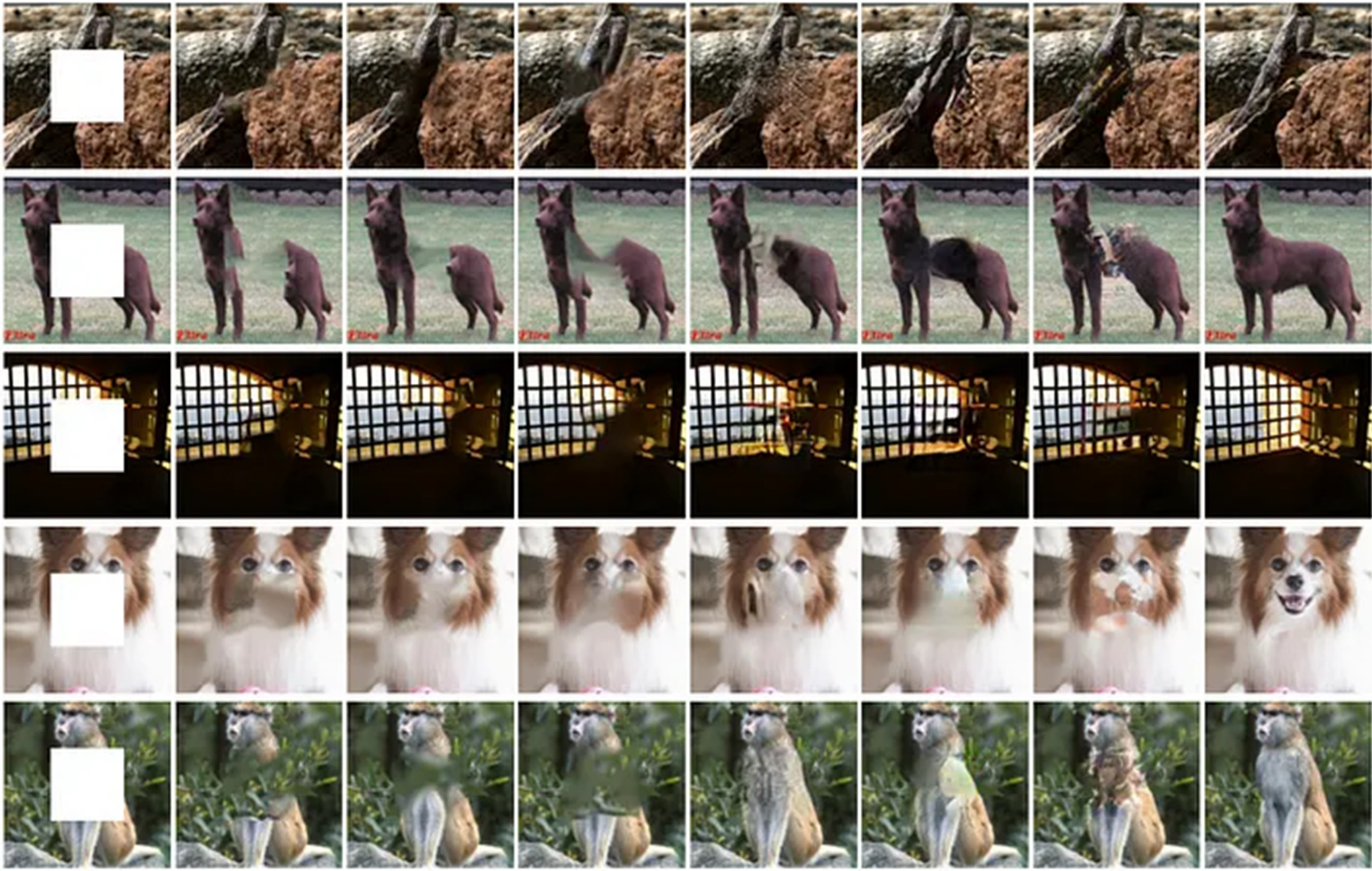
And

Inverse filter outputs


Original
Restored

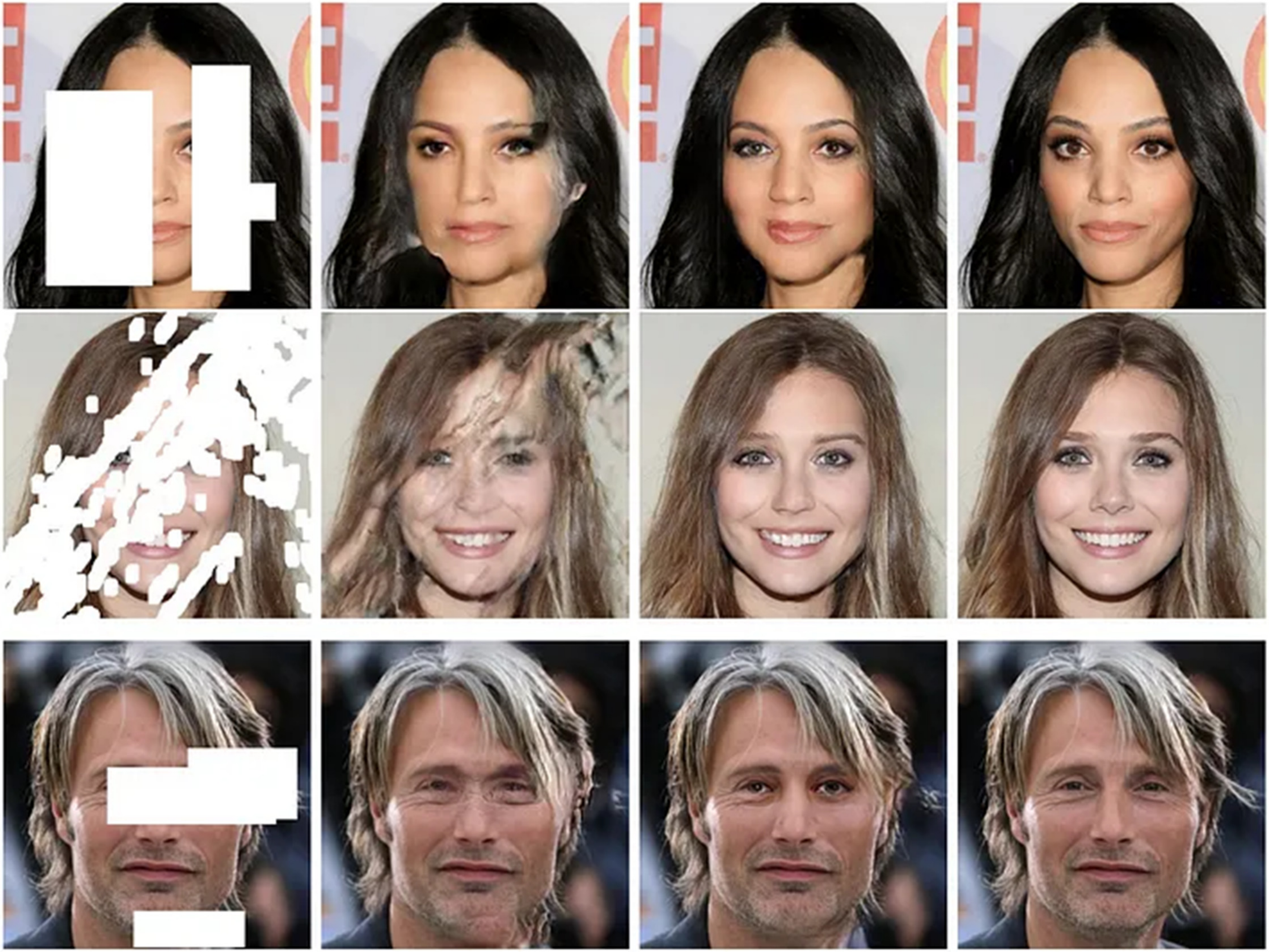

Original
Restored


Input
PatchMatch
Image Melding
[Huang et al. 2014]
[Pathak et al. 2016]
Ours



(a) Input (b) PM (c) IM (d) SC (e) CE (f) Ours (CM) (g) Ours (h) GT

(e) Oı



(a) Input (b) GntIpt (c) PConv(Ours) (d) Ground Truth

Gu
Brv
Co

nd

18:

Image Enhancement

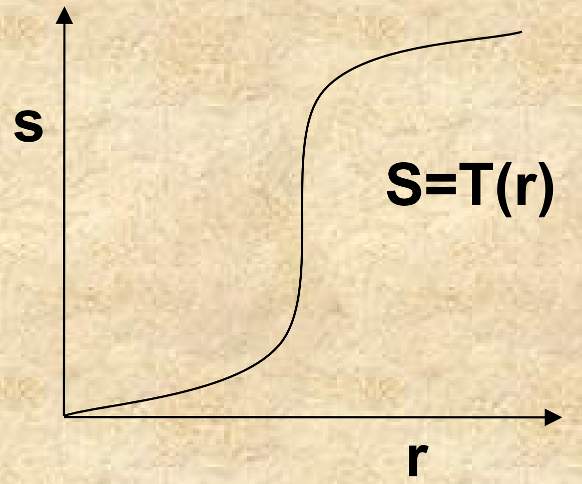
CONTRAST STRETCHING

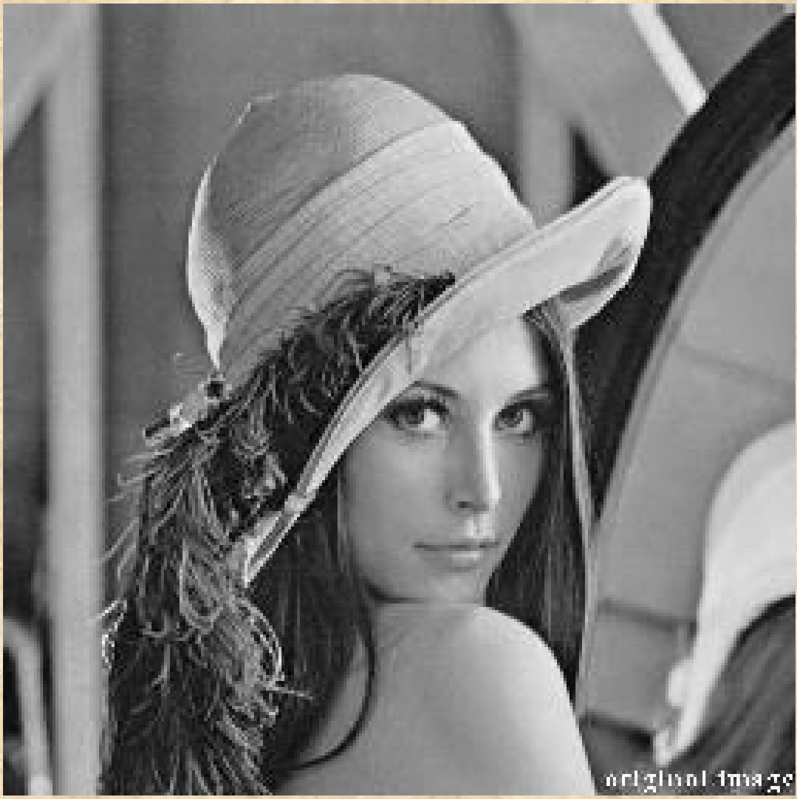
This is a pixel-based operation, where a given gray level $r \in [0, L]$ mapped into a gray level $s \in [0, L]$ according to a transformation function:

$$s = T(r)$$

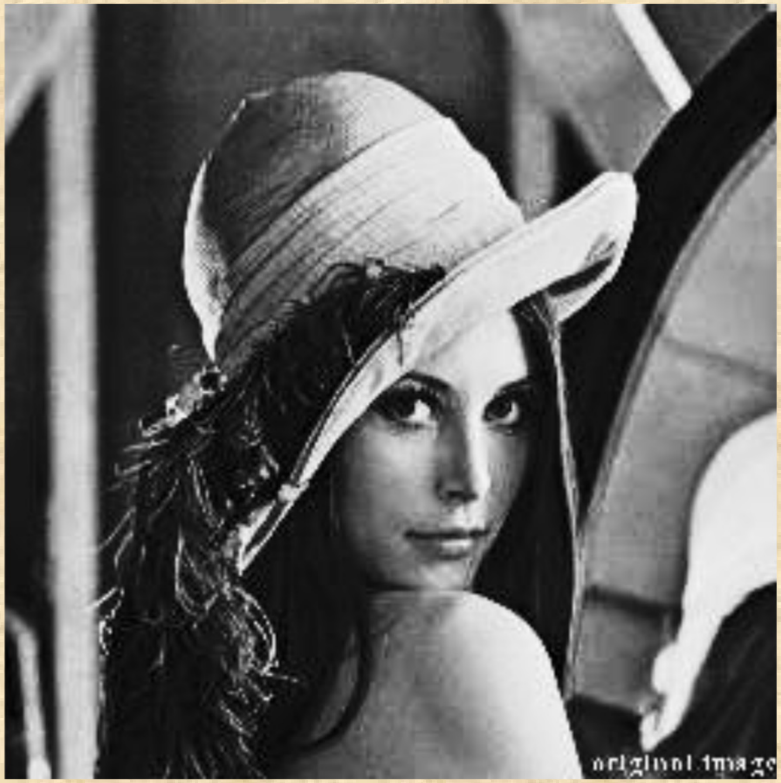
This process is mainly used to enhance done to handle low-contrast images occurring due to poor or non-uniform lighting conditions or due to non-linearity or small dynamic range of the imaging sensor.

Following examples shows some typical contrast stretching transformations.

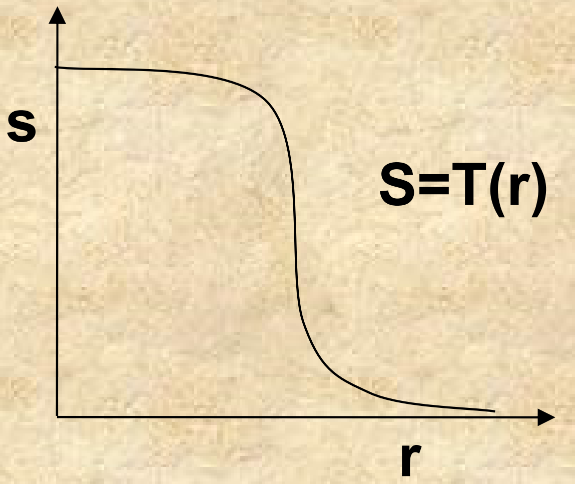


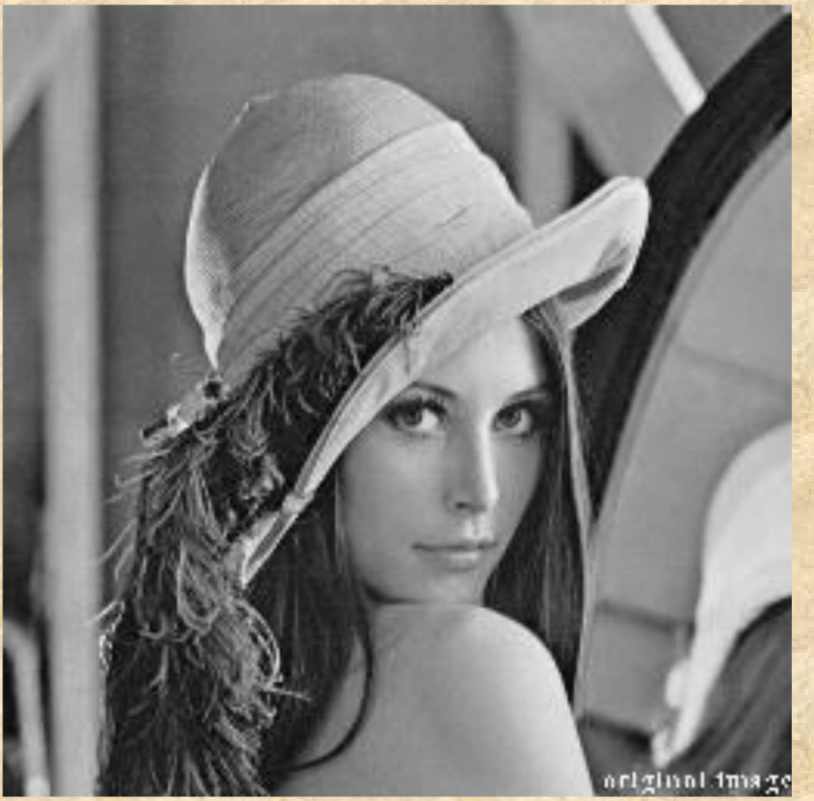


Original image

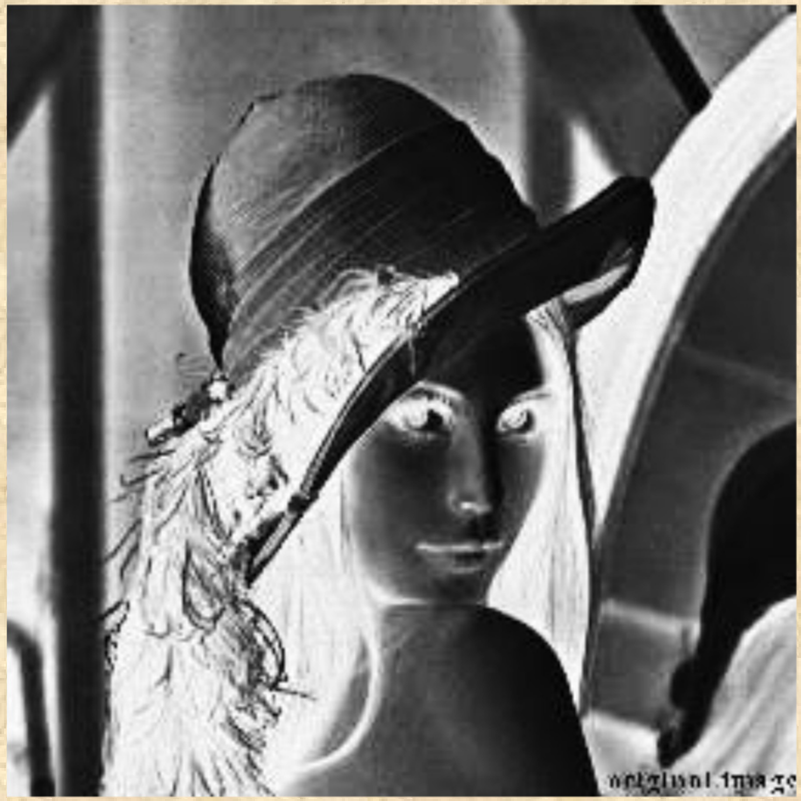


After contrast stretching

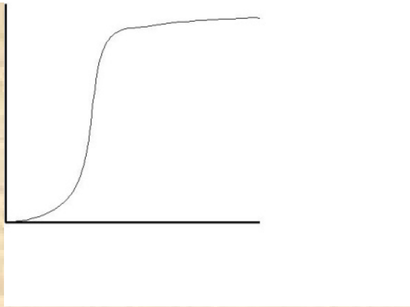


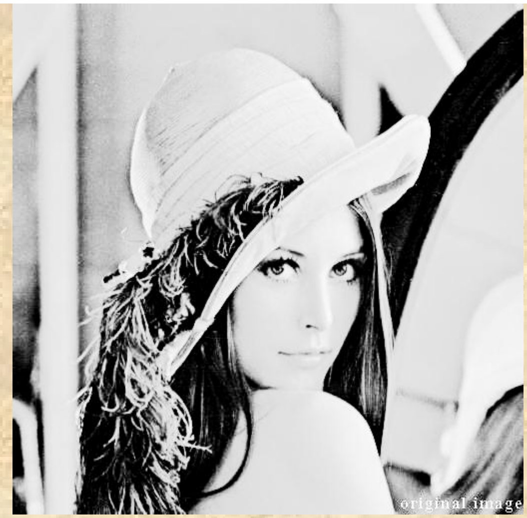


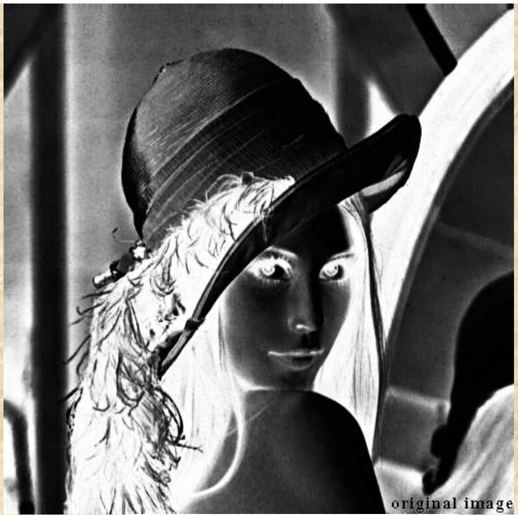
Original image

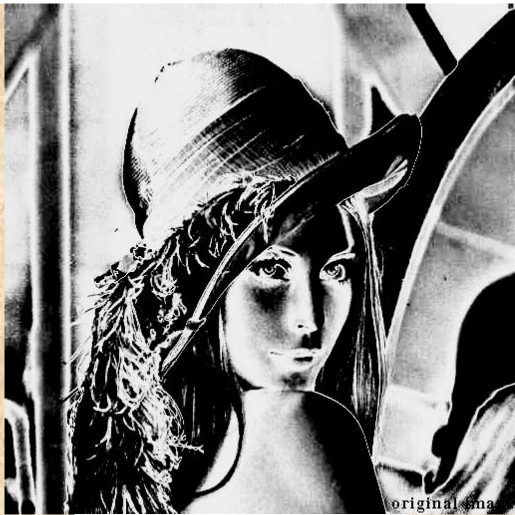


After contrast stretching

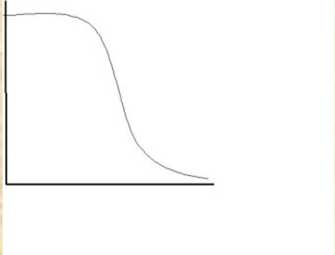


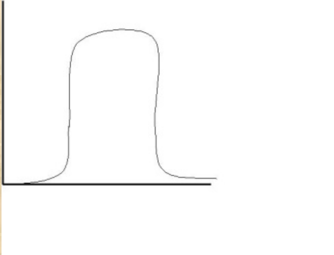

original image

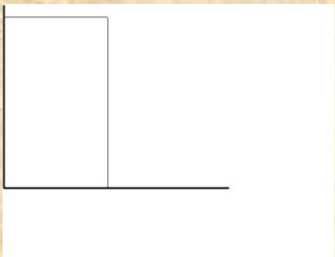

original image

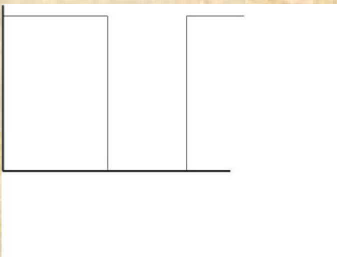

original ima

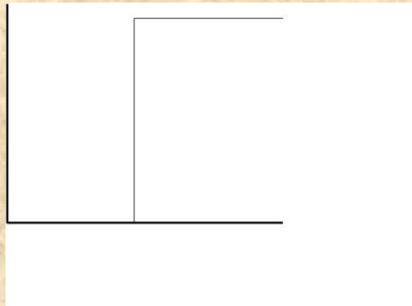
original image

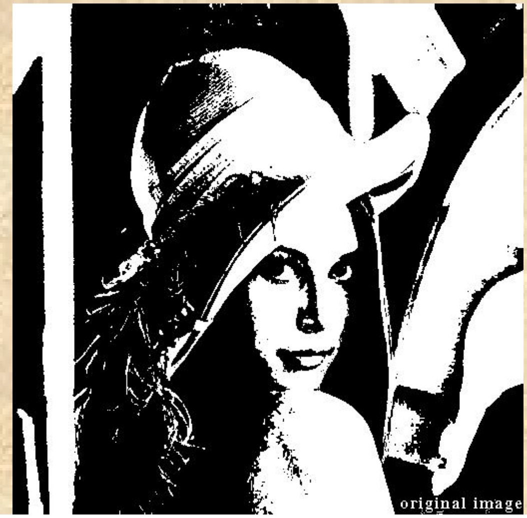


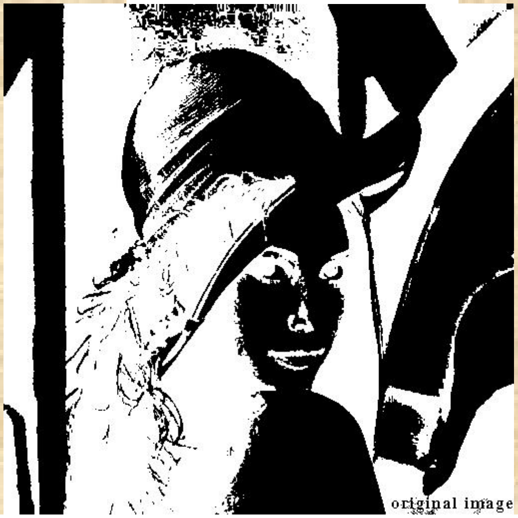


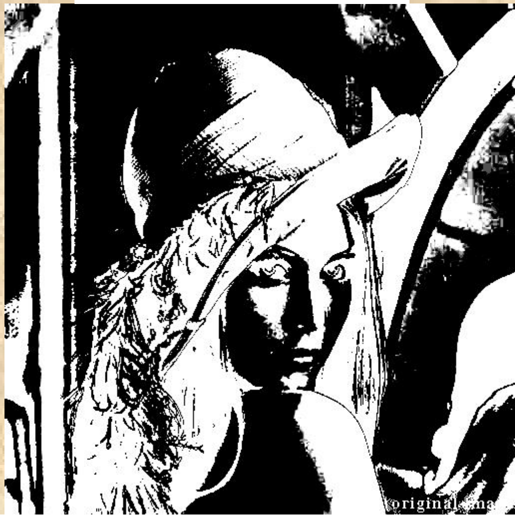


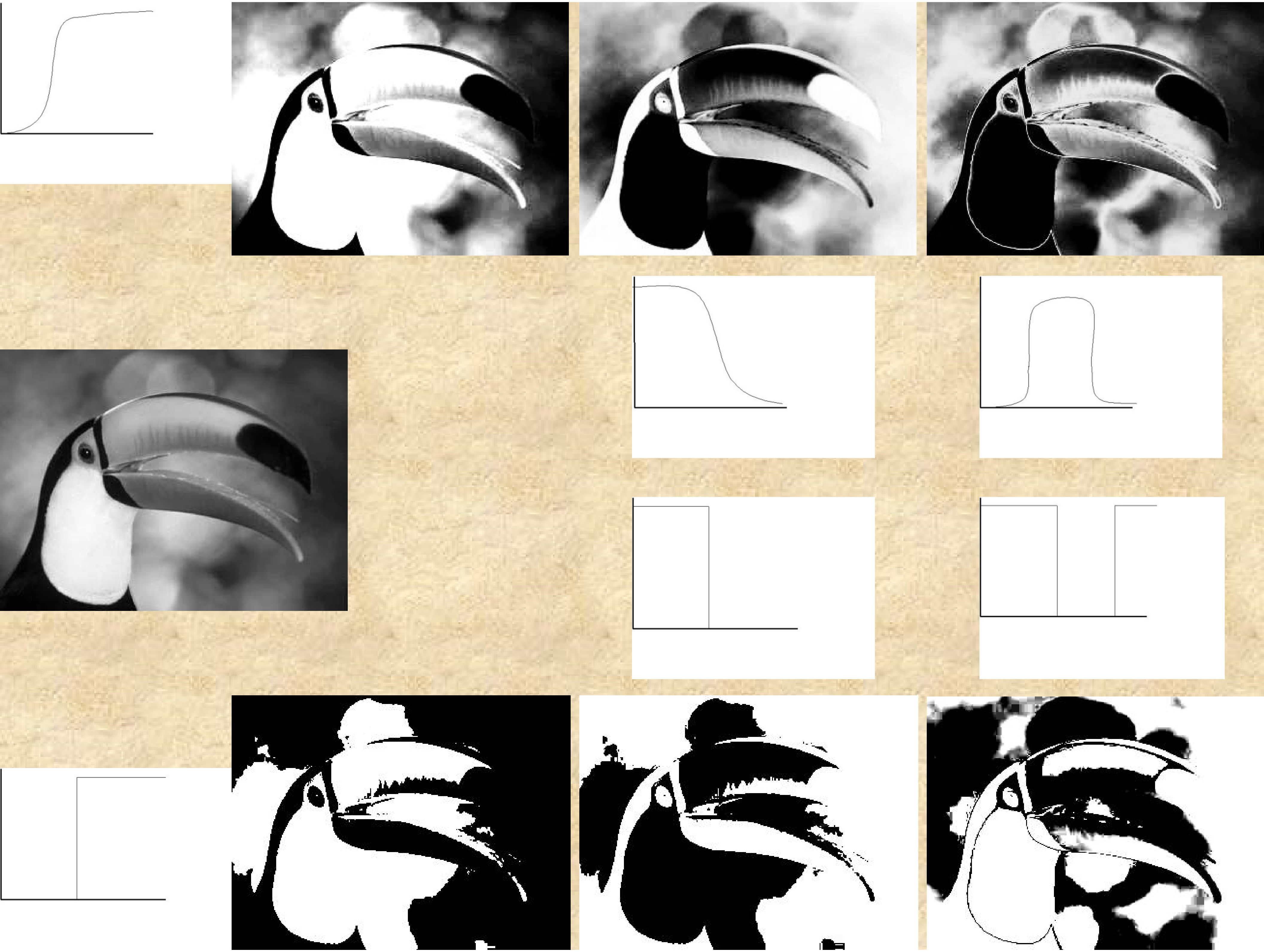


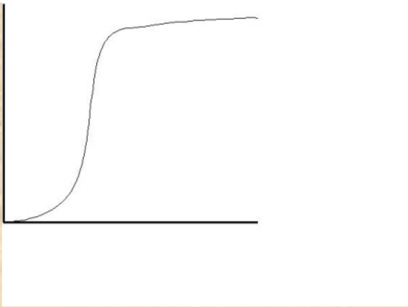


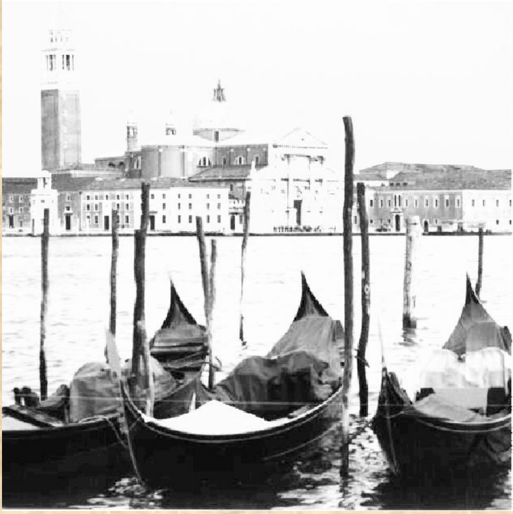

original image

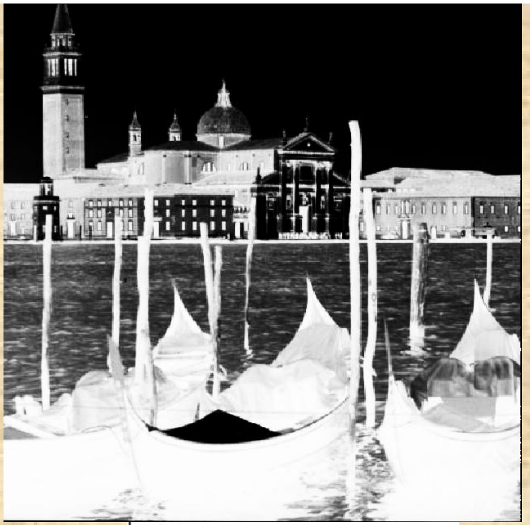

original image

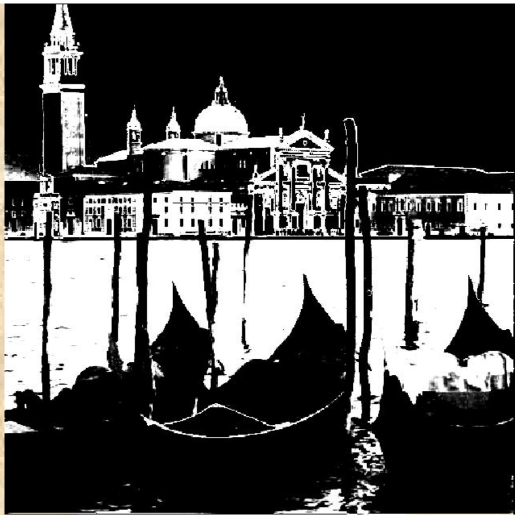

original ima

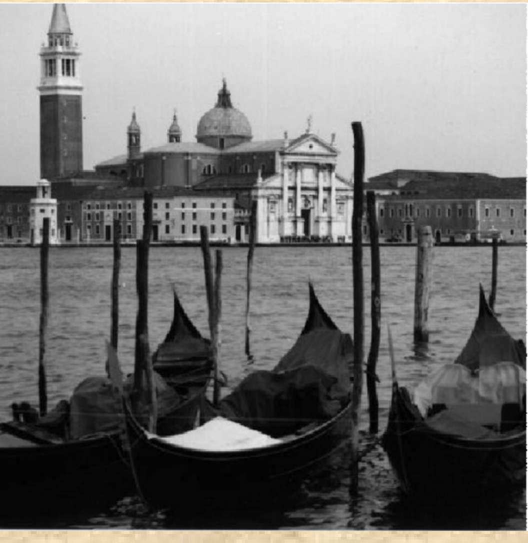


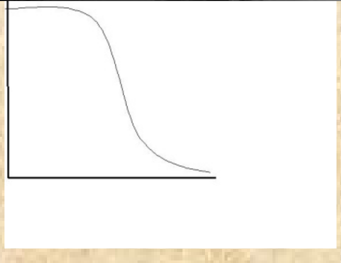


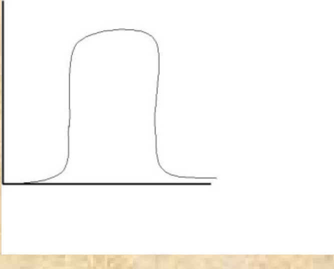


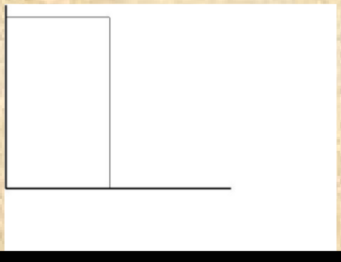


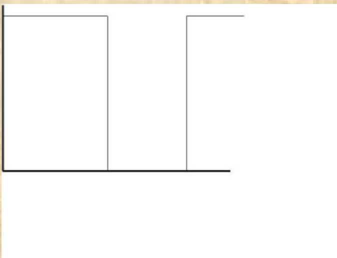


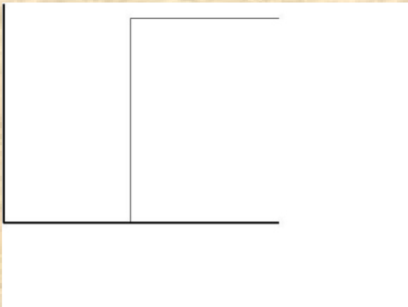


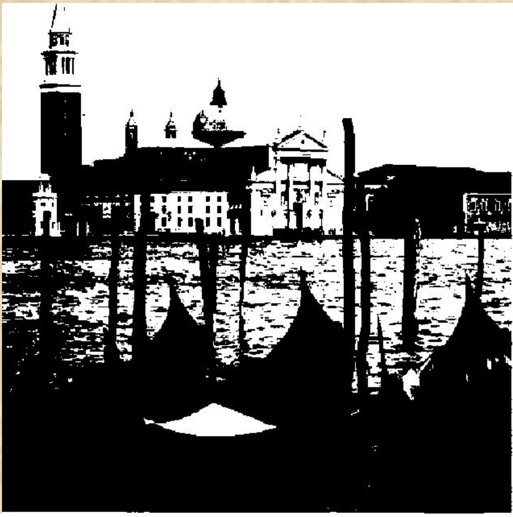


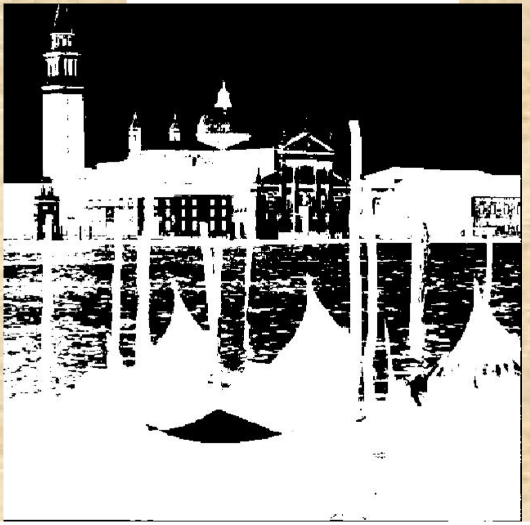


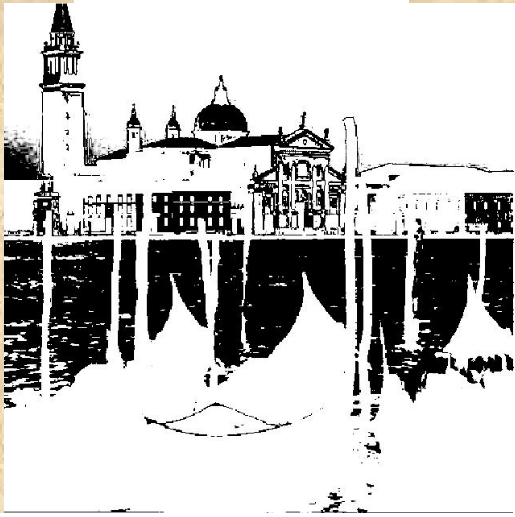












How to find a s = T(r), which depends on the image data and hence produces a global transform/enhancement of the image, and not simply a local transformation of the pixel ??

Let us look at Histogram Equalization next**.**