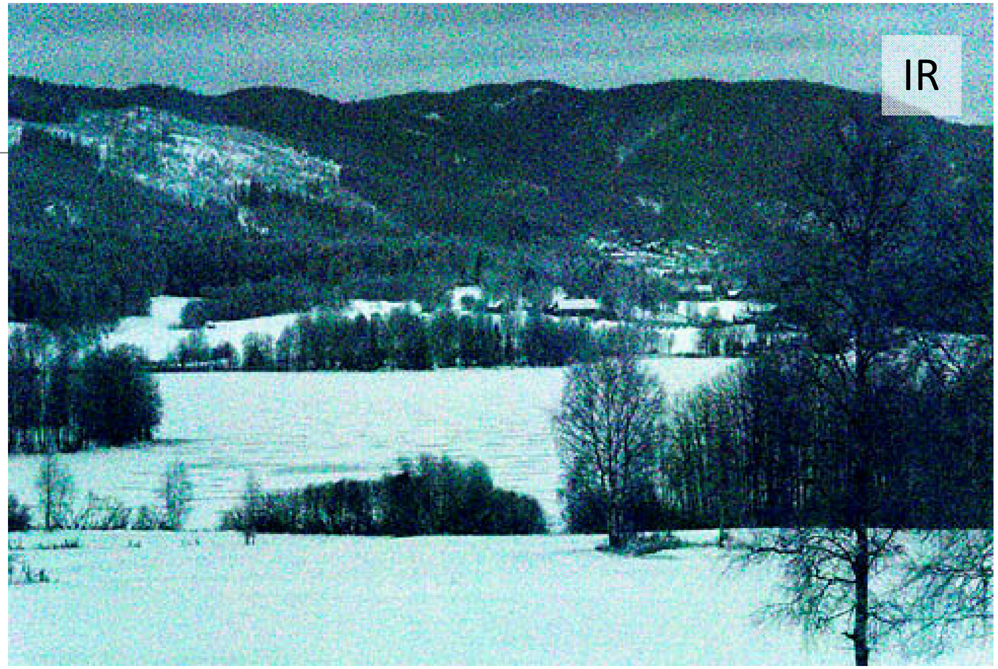
Video Signals

IMAGE RESTORATION

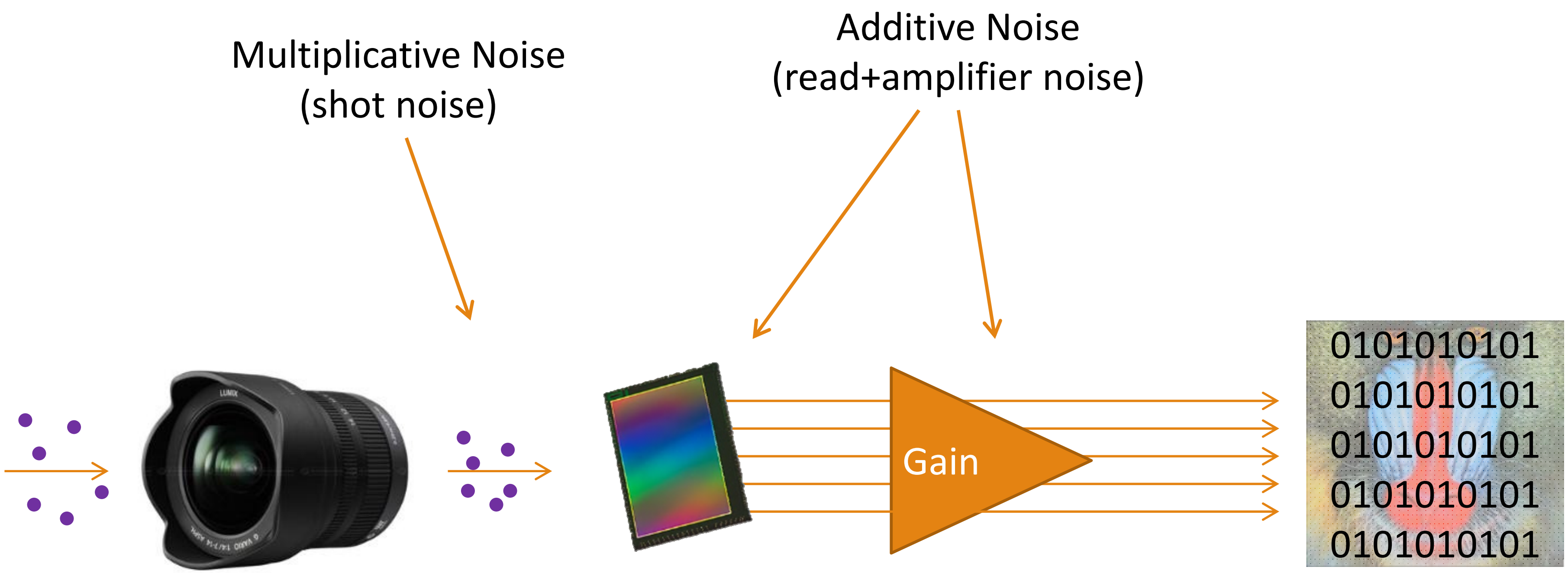

Indoor – low light


IR


US

Can we (humans) denoise?

Sources of Noise



The photographic camera

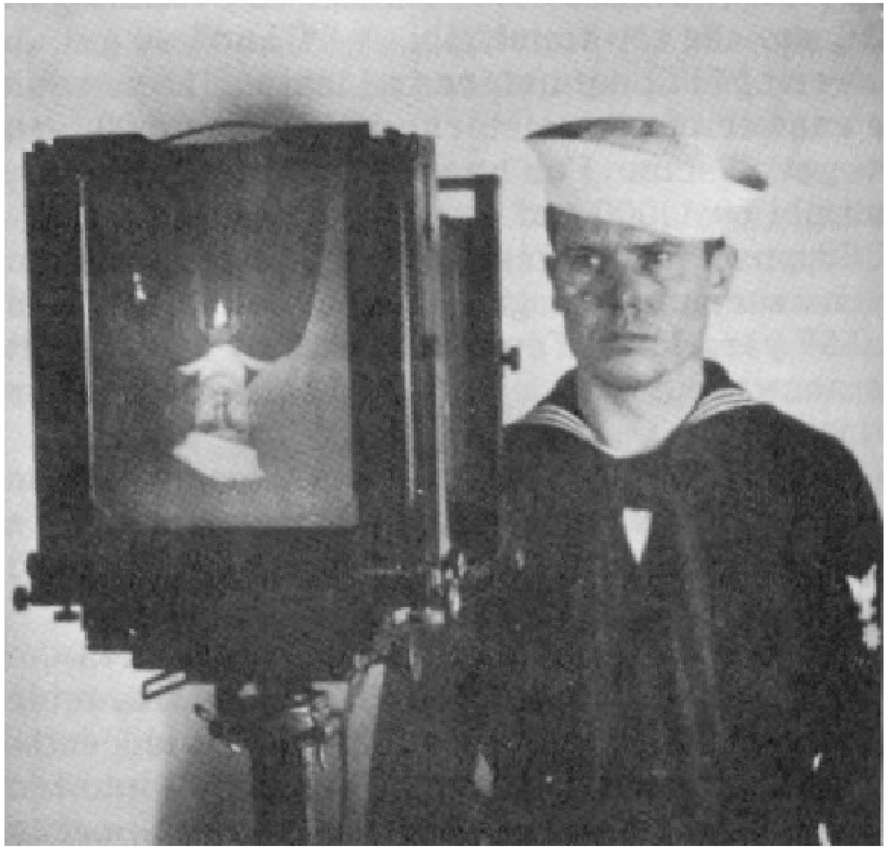
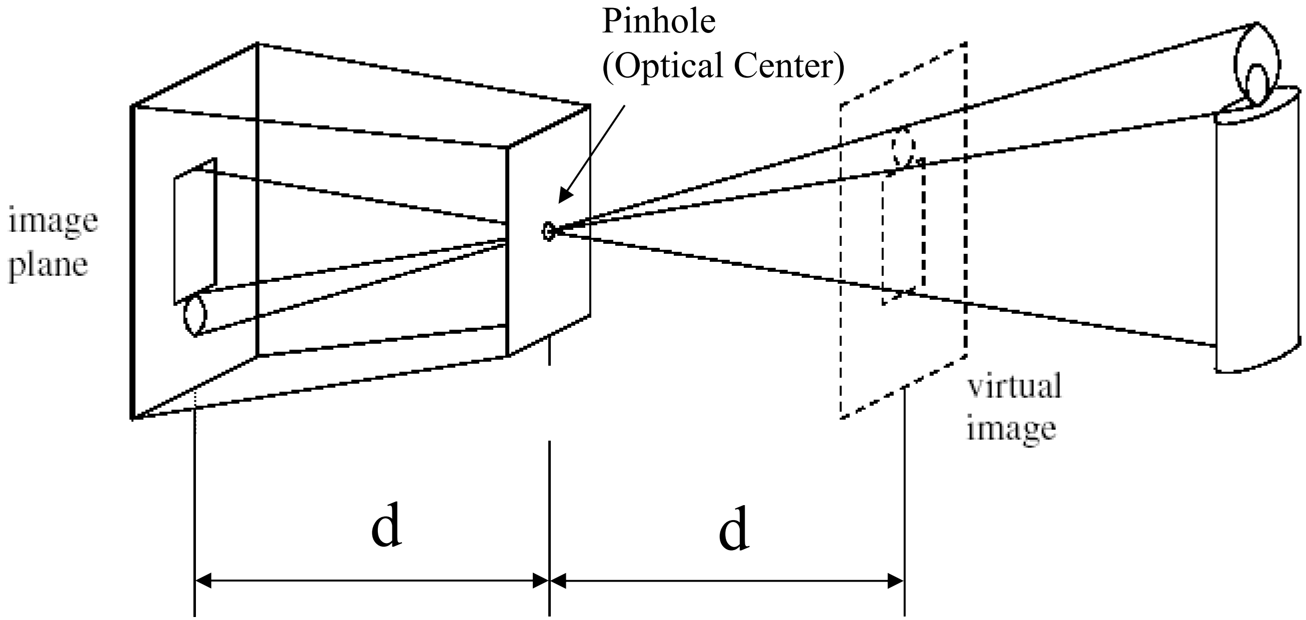


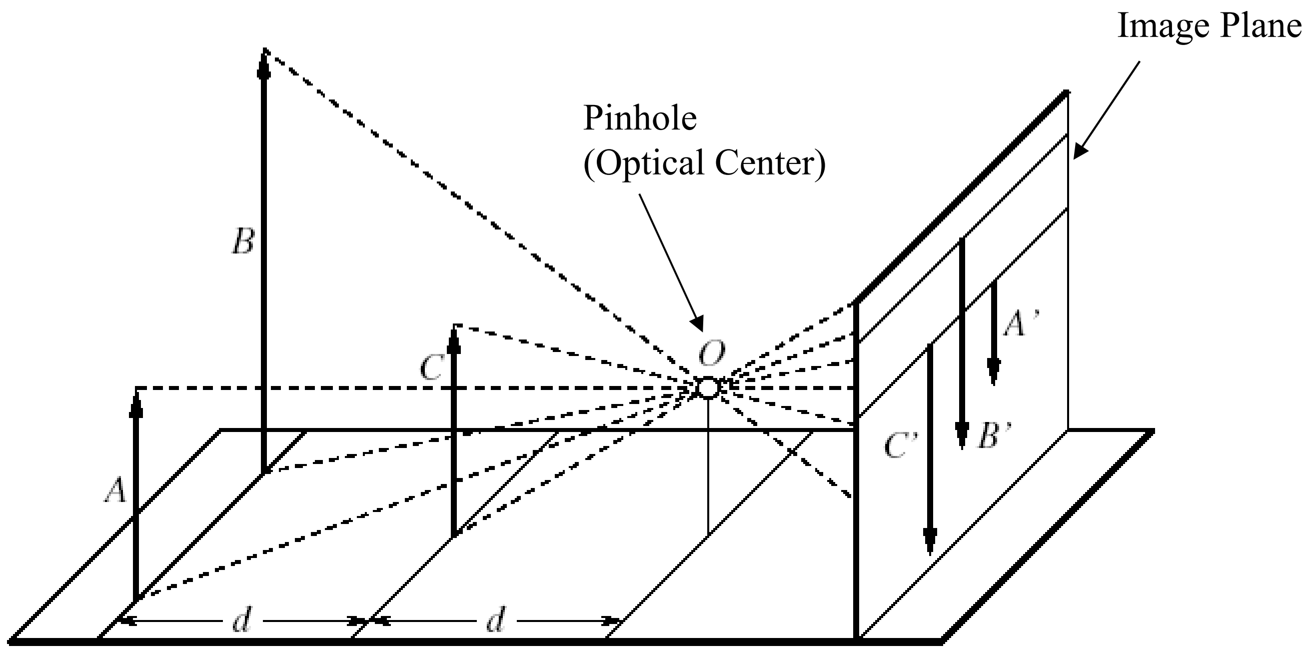
Image formation on the back-plate of a photographic camera

Pin-hole Camera - the Perspective Projection



The pinhole imaging model

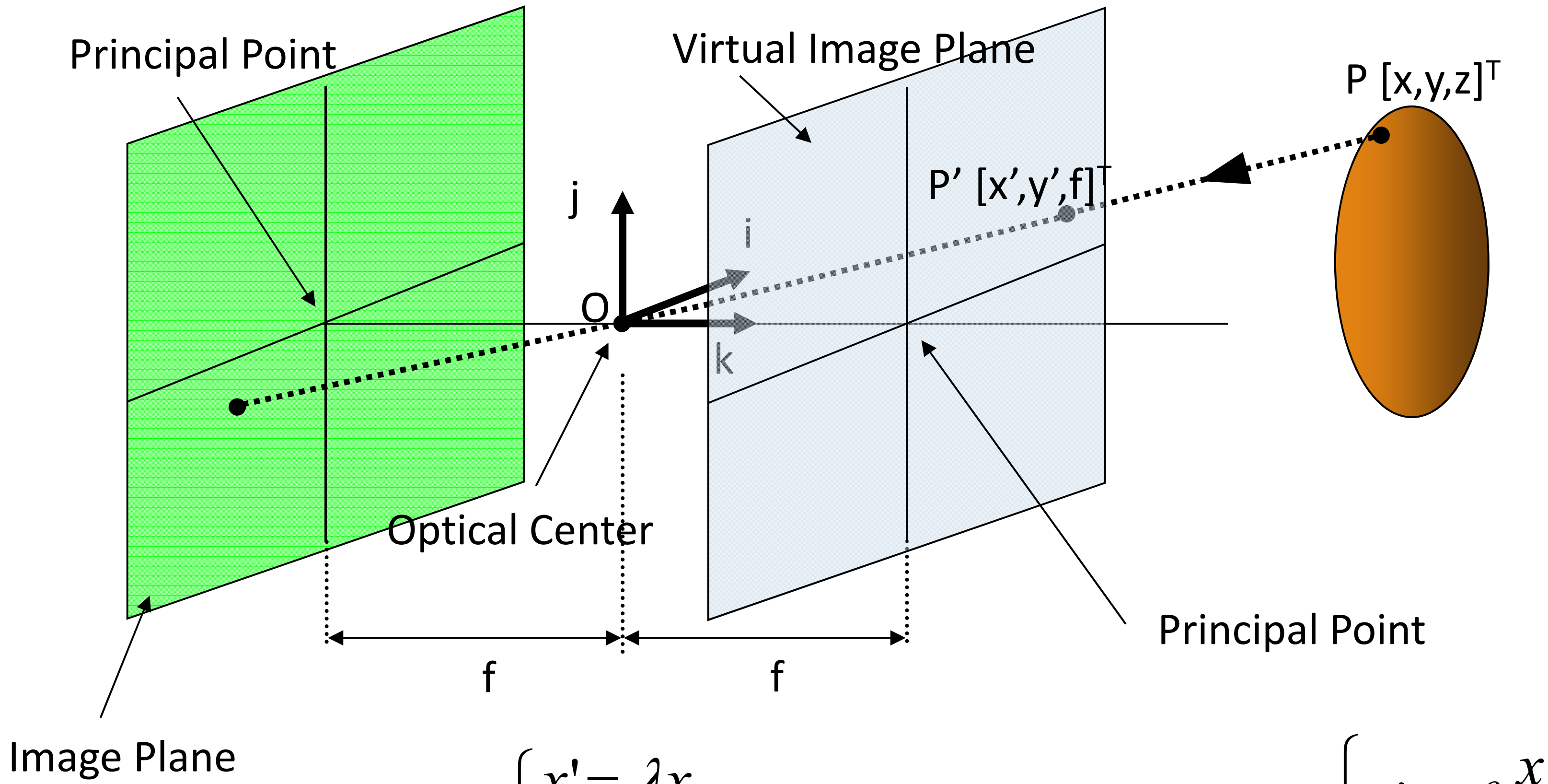
Perspective Projection



Far objects appear smaller than close ones

The distance *d* from the pinhole *O* to the plane containing *C* is half the distance from *O* to the plane containing *A* and *B*.

Basic Equations of Perspective Projection



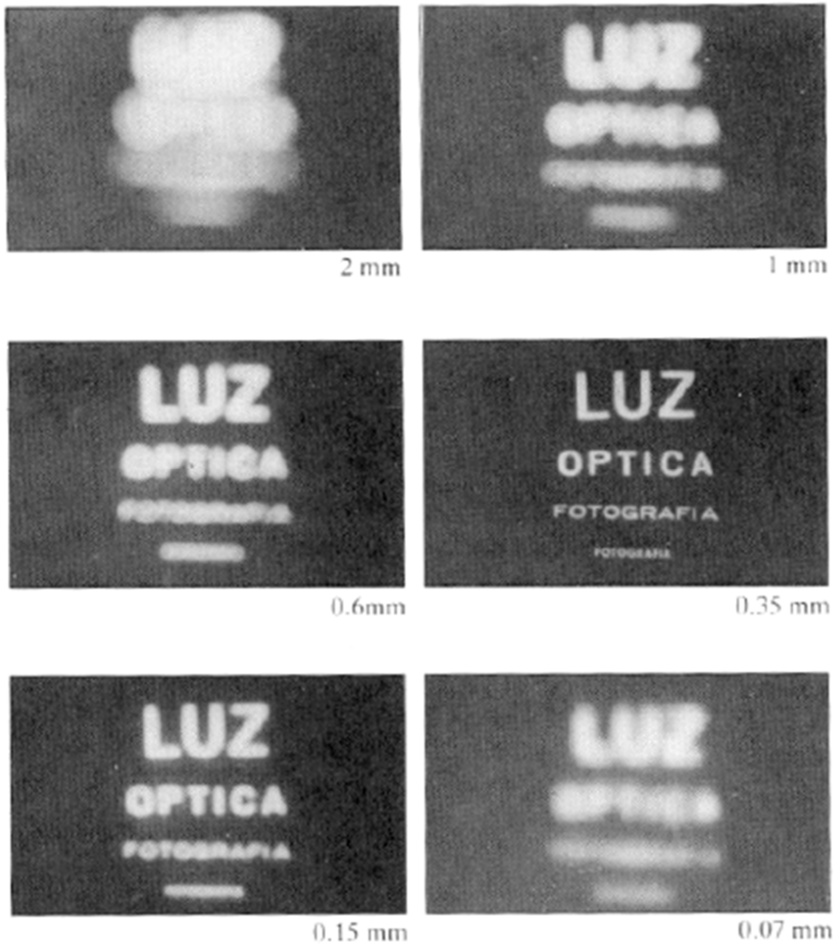
$$\begin{cases} x' = \lambda x \\ y' = \lambda y \\ f = \lambda z \end{cases} \Leftrightarrow \lambda = \frac{x'}{x} = \frac{y'}{y} = \frac{f}{z} \quad \Rightarrow \begin{cases} x' = f\dfrac{x}{z} \\ y' = f\dfrac{y}{z} \end{cases}$$

Pinhole Images

Exposure 4 seconds

Exposure 96 minutes

Why not a real pin-hole camera?

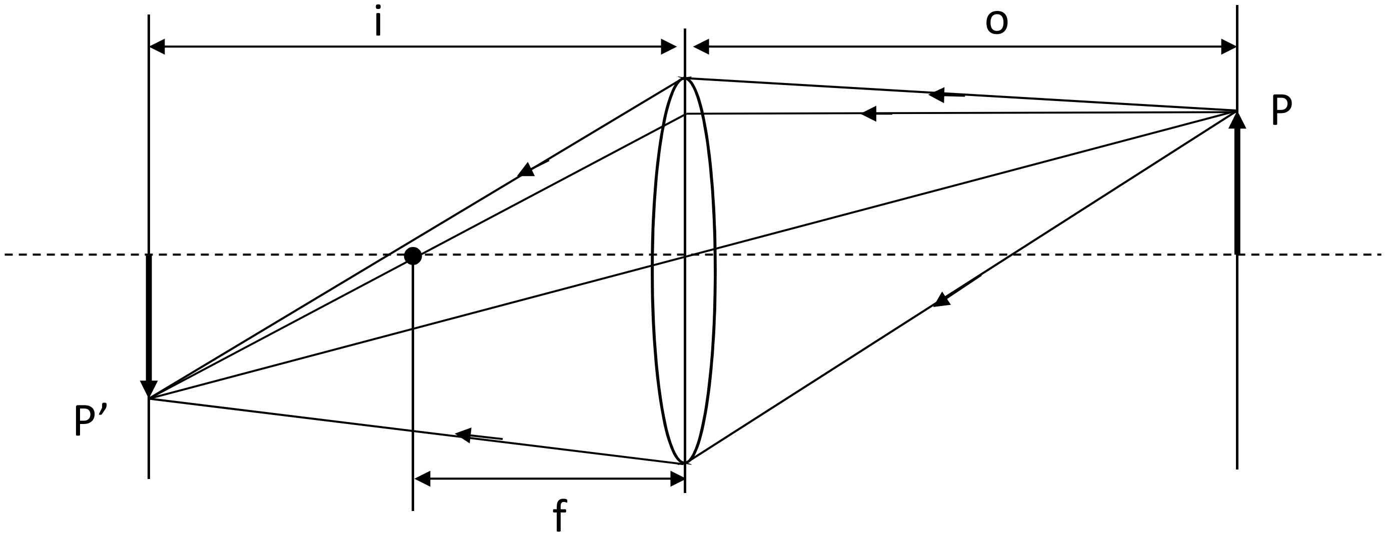


Images of some text obtained with shrinking pinholes:

- large pinholes give bright but fuzzy images;
- pinholes that are too small also give blurry images because of diffraction effects.

Image Formation using Lenses

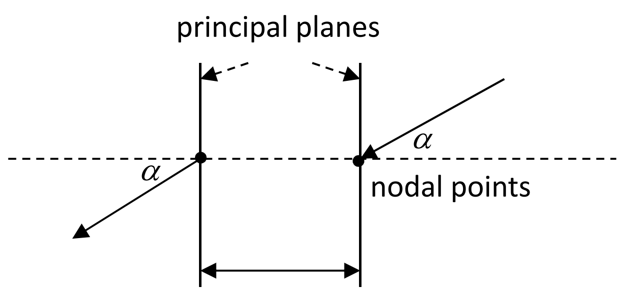
- Lenses are used to avoid problems with pinholes.
- Ideal Lens: Same projection as pinhole but gathers more light!



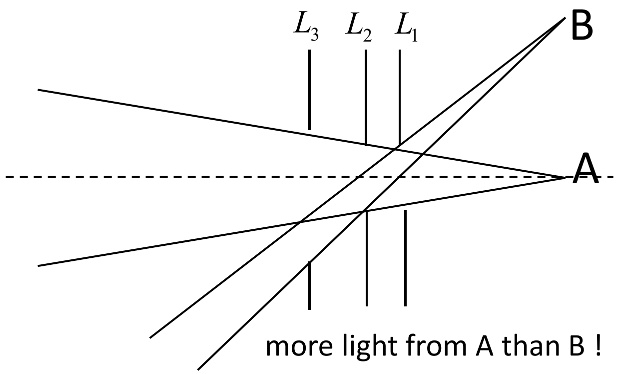
- Gaussian Thin Lens Formula: $$\frac{1}{i}+\frac{1}{o}=\frac{1}{f}$$
- f is the focal length of the lens – determines the lens's ability to refract light
- f different from the effective focal length f' discussed before!

Common Lens Related Issues - Summary

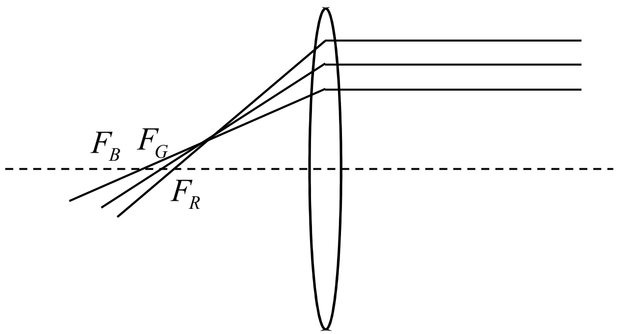
Compound (Thick) Lens



Vignetting



Chromatic Abberation



Lens has different refractive indices for different wavelengths.

Radial and Tangential Distortion

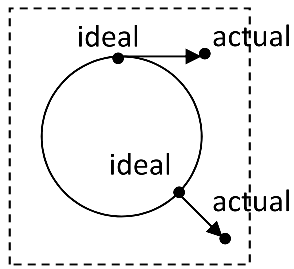
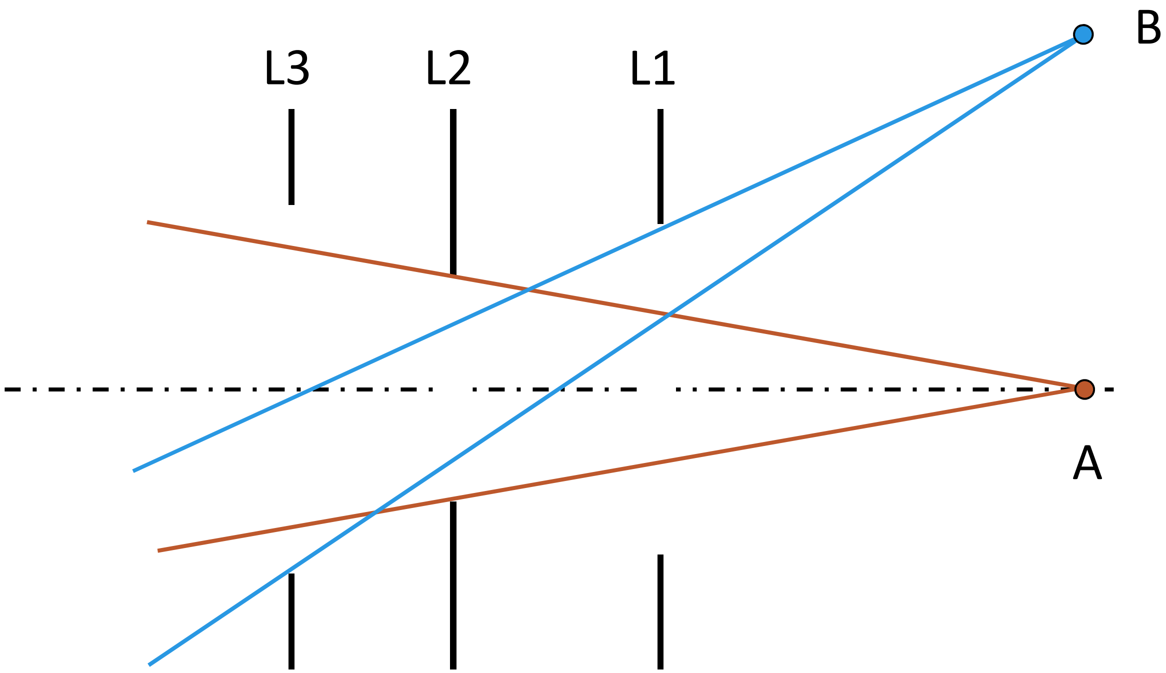


image plane

Lens Glare

- Stray interreflections of light within the optical lens system.
- Happens when very bright sources are present in the scene.

Vignetting



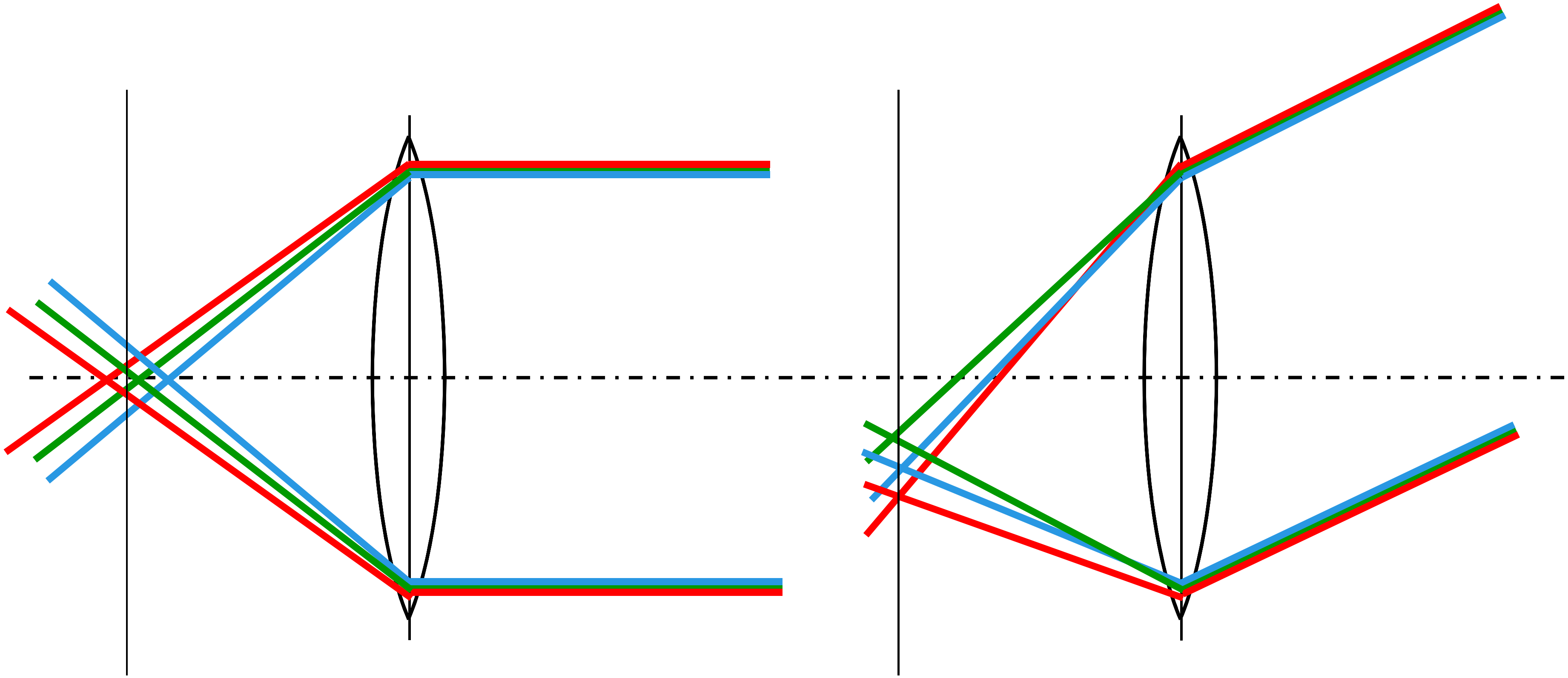
More light passes through lens L3 for scene point A than scene point B

Results in spatially non-uniform brightness (in the periphery of the image)

Vignetting

photo by Robert Johnes

Chromatic Aberration

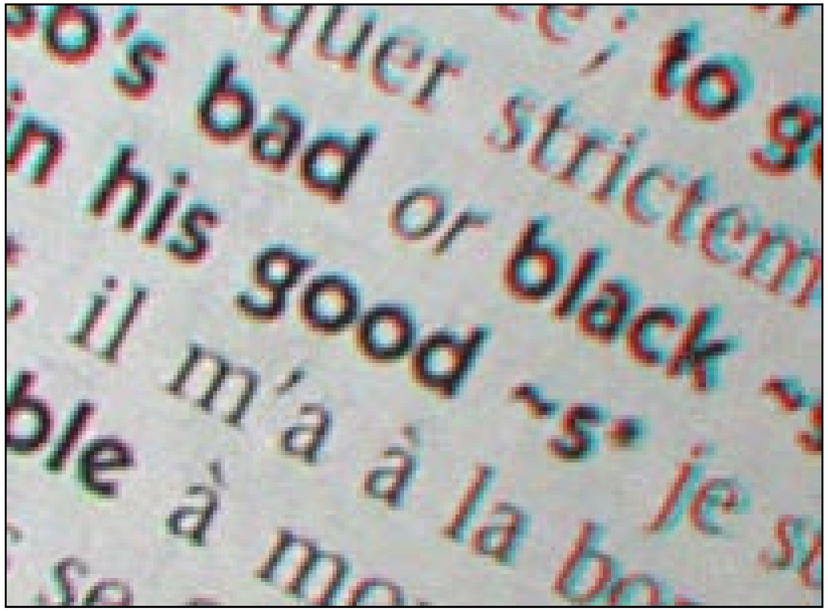


longitudinal chromatic aberration
(axial)

transverse chromatic aberration
(lateral)

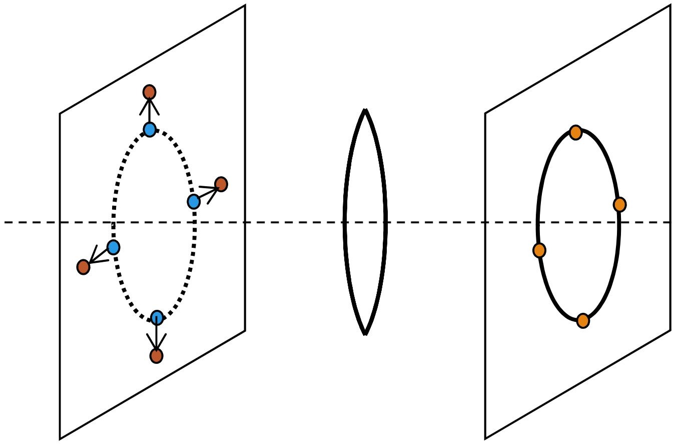
Chromatic Aberrations

longitudinal chromatic aberration
(axial)

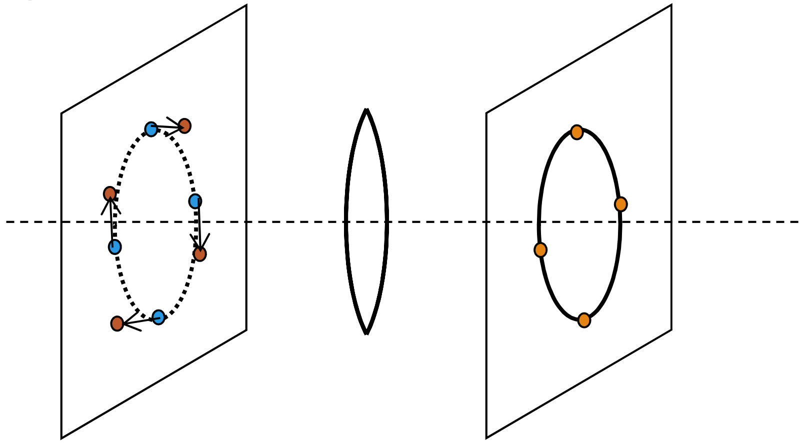


transverse chromatic aberration
(lateral)

Geometric Lens Distortions



Radial distortion



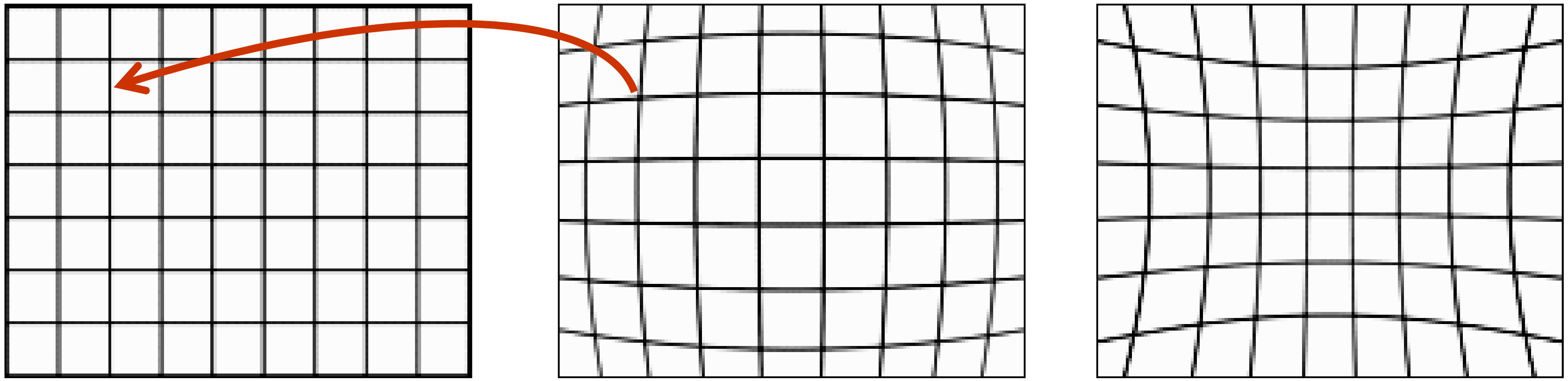
Tangential distortion

Both due to lens imperfection

Rectify with geometric camera calibration

Photo by Helmut Dersch

Radial Lens Distortions



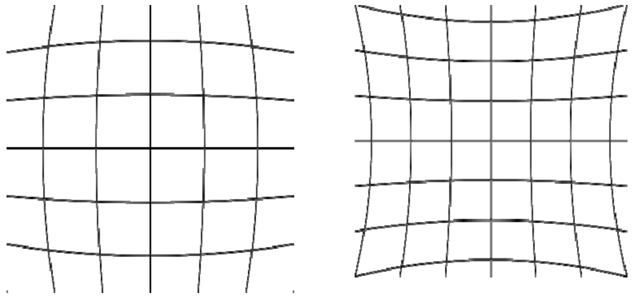
No Distortion | Barrel Distortion | Pincushion Distortion

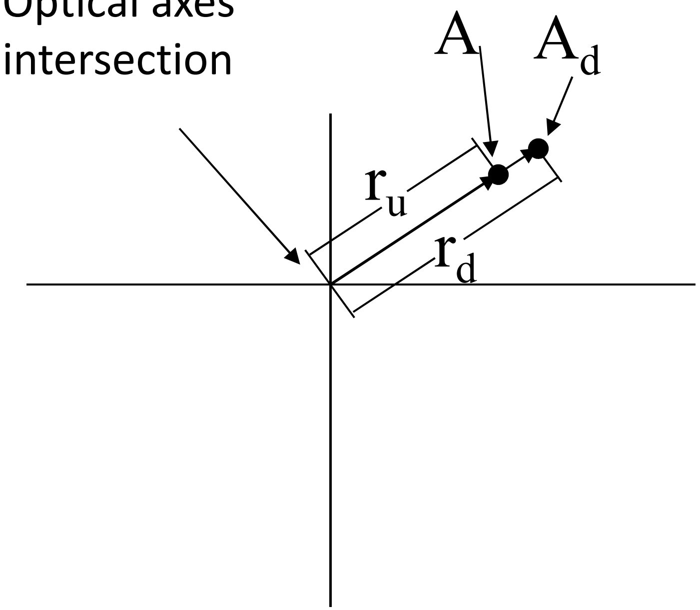
- Radial distance from Image Center:

$$r_u = r_d + k_1 r_d^3$$

Real lenses: Radial Distortion

In many cases the lens distortion can be well-modeled as radial. This is the case of the common pincushion and barrel distortions.





$$
\begin{aligned}
(x_u, y_u) &= \text{undistorted image point},\\
(x_d, y_d) &= \text{distorted image point},\\
(x_c, y_c) &= \text{centre of distortion},\\
K_n &= \text{N}^{\text{th}} \text{ radial distortion coefficient},\\
P_n &= \text{N}^{\text{th}} \text{ tangential distortion coefficient}\\
r &= \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2},
\end{aligned}
$$

$$
\begin{aligned}
x_u =& x_d + (x_d - x_c)(K_1 r^2 + K_2 r^4 + \ldots) + \\
& \left(P_1(r^2 + 2(x_d - x_c)^2) + 2P_2(x_d - x_c)(y_d - y_c)\right)(1 + P_3 r^2 + \ldots)\\
y_u =& y_d + (y_d - y_c)(K_1 r^2 + K_2 r^4 + \ldots) + \\
& \left(2P_1(x_d - x_c)(y_d - y_c) + P_2(r^2 + 2(y_d - y_c)^2)\right)(1 + P_3 r^2 + \ldots))
\end{aligned}
$$

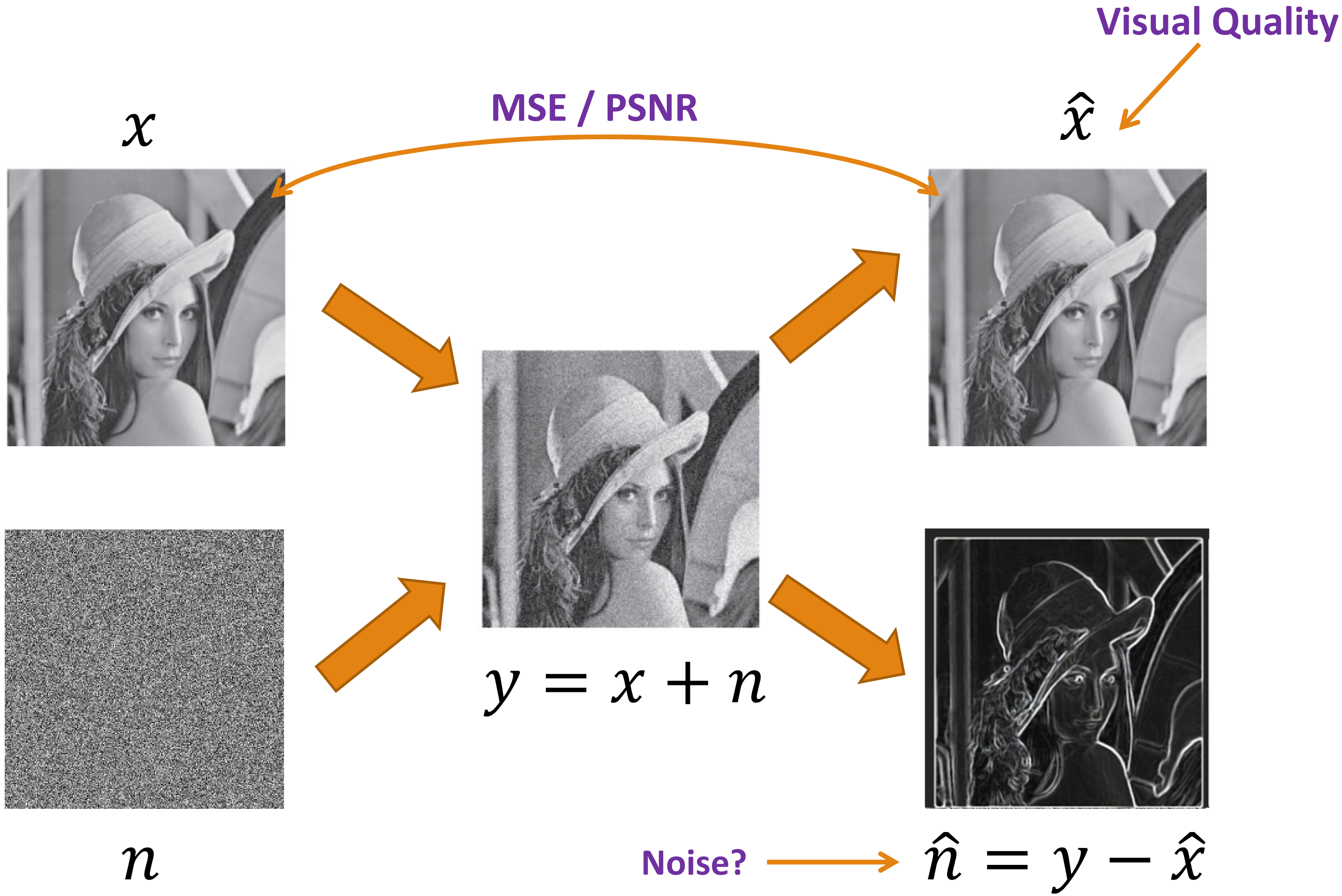
Correcting Radial Lens Distortions

Before

After

IMAGE DENOISING

Problem Definition



Problem Definition

Visual Quality

MSE / PSNR

x

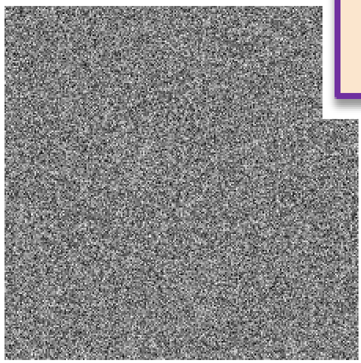
$\hat{x}$

<u>Mean Square Error</u>

$$\mathrm{MSE} = \|\hat{x} - x\|_2^2$$

<u>Peak Signal-to-Noise ratio</u>

$$\mathrm{PSNR} = 20log_{10}\frac{255}{\sqrt{\mathrm{MSE}}}$$



$$y = x + n$$

n

Noise? → $\hat{n} = y - \hat{x}$

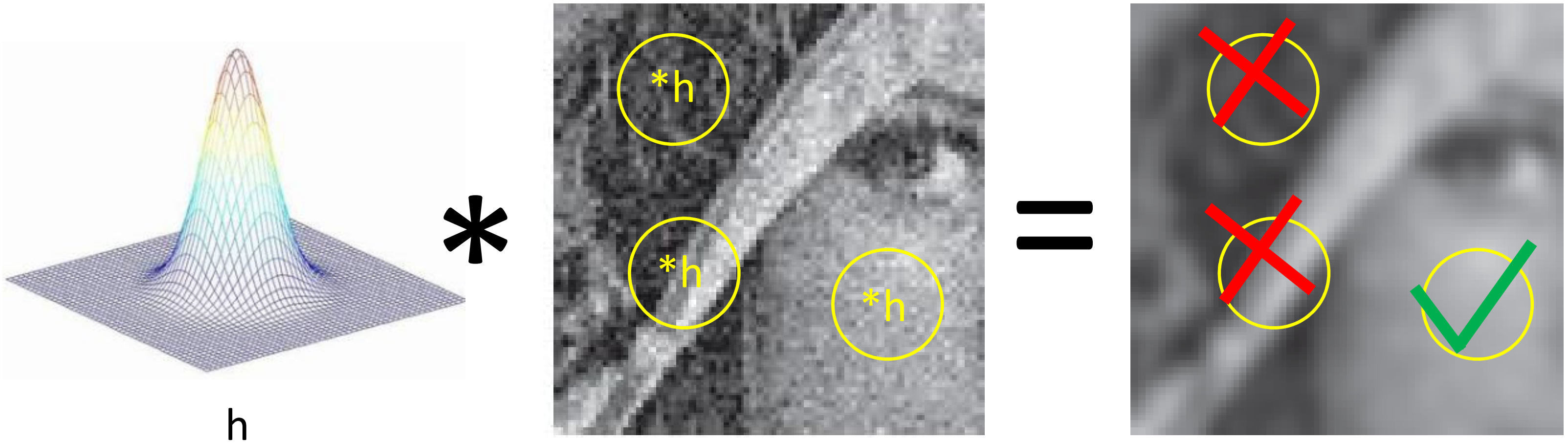
Denoising in the Spatial Domain

The “classical” assumption:
Images are piecewise constant

Neighboring pixels are highly correlated

$\Rightarrow$ Denoise = “Average nearby pixels” (filtering)

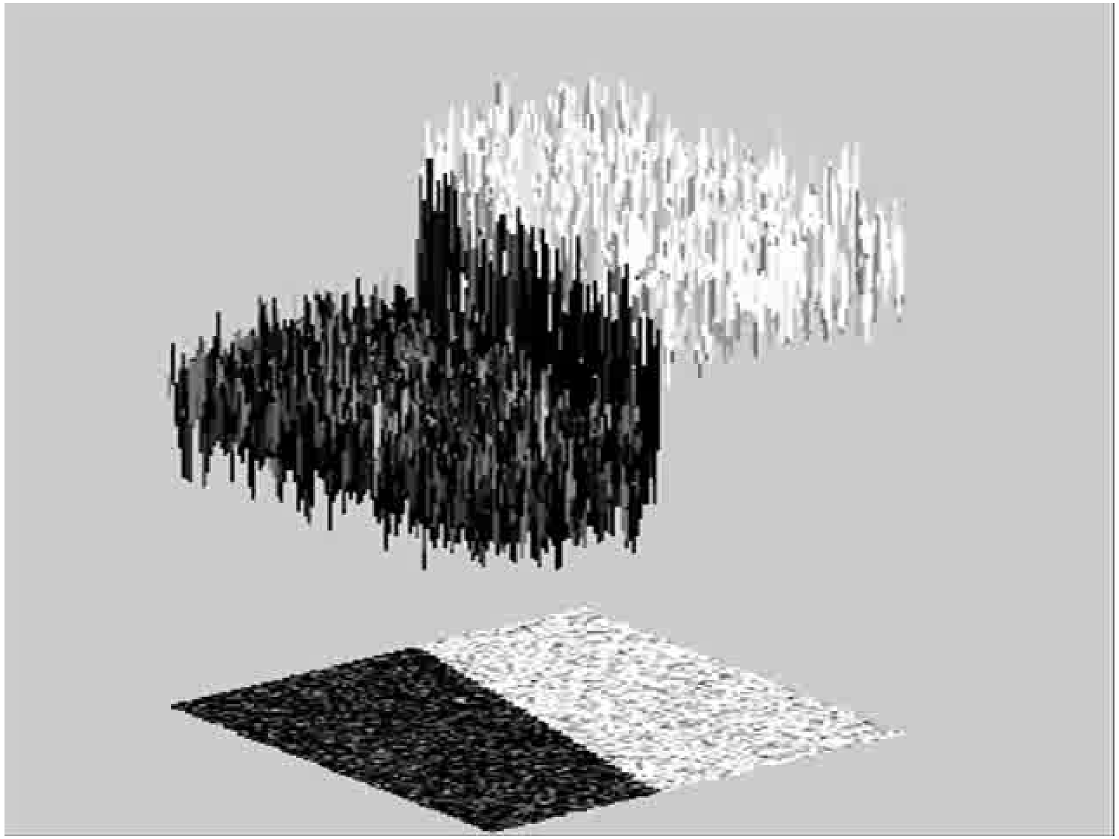
Gaussian Smoothing



h

$$\hat{x}(i) = \frac{1}{C_i}\sum_{j} y(j)\, e^{-\frac{\|i-j\|^2}{2\sigma^2}}$$

Toy Example



How can we preserve the fine details?

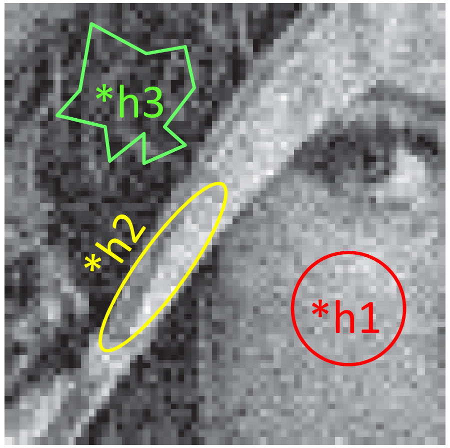
Local adaptive smoothing

Non uniform smoothing

Depending on image content:

- Smooth where possible
- Preserve fine details

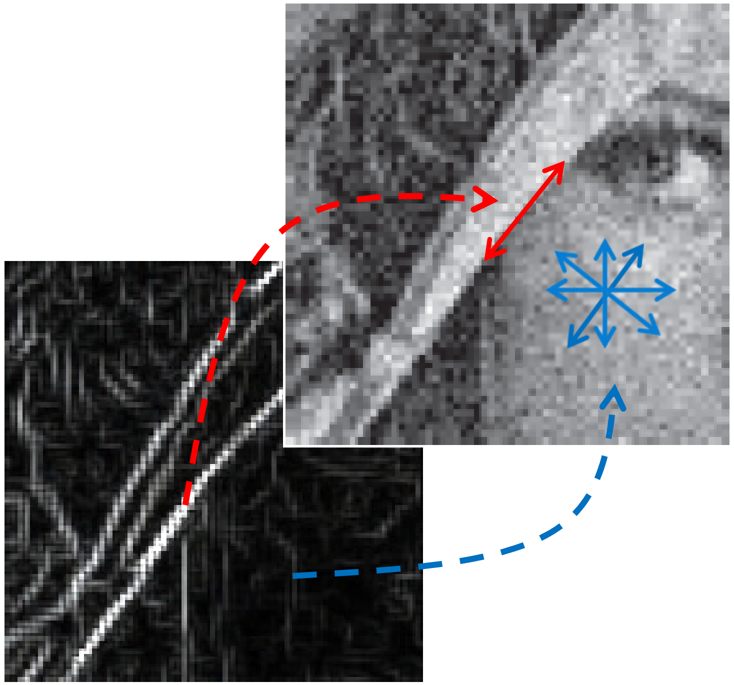
How?



Anisotropic Filtering

Edges ⇒ smooth only along edges

"Smooth" regions ⇒ smooth isotropically

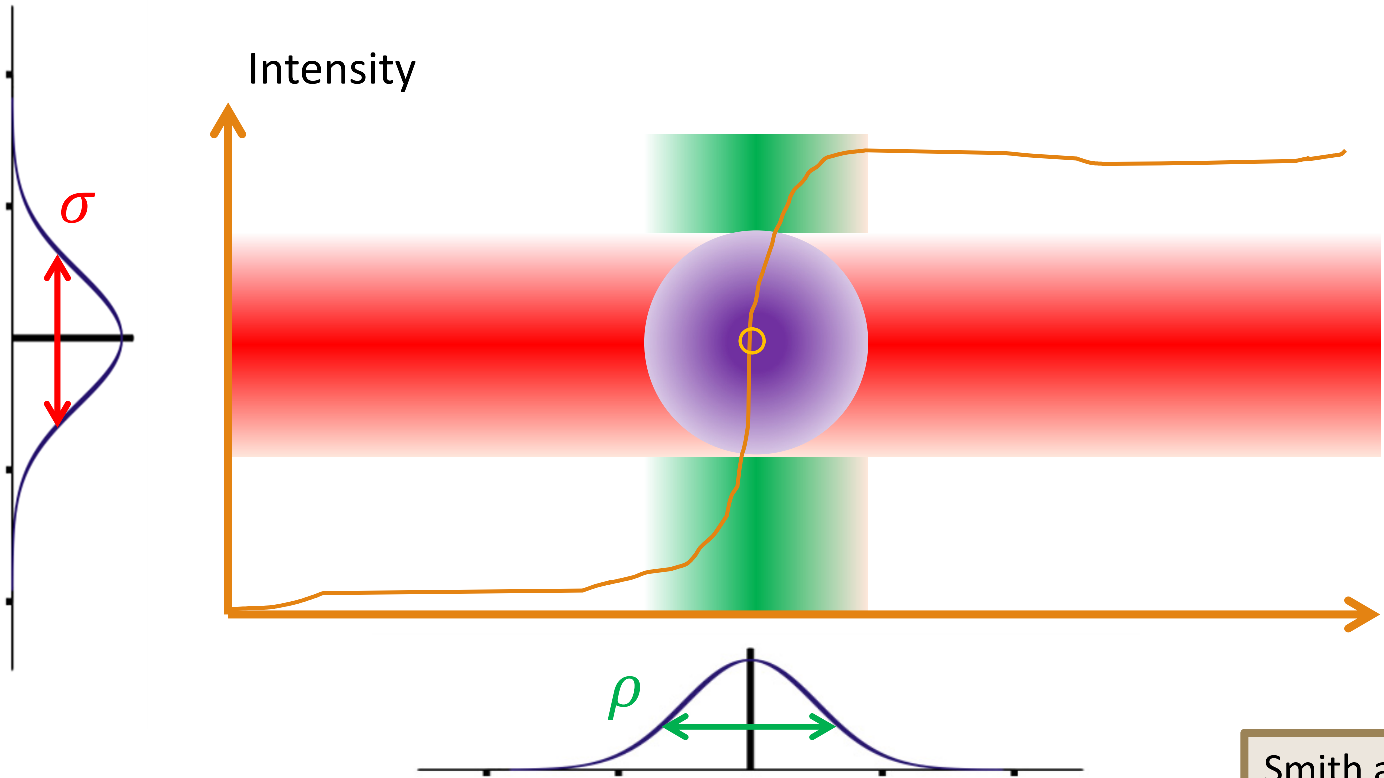


gradient

Perona and Malik (1990)

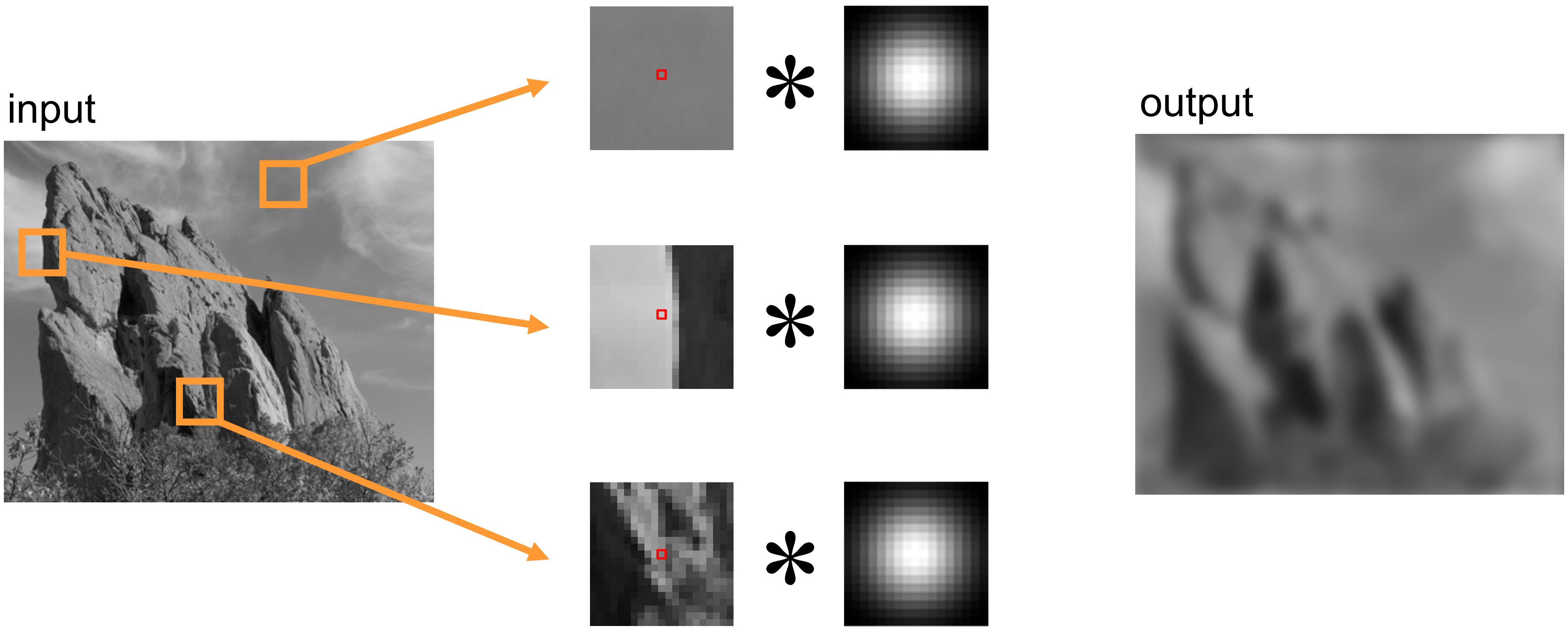
Bilateral Filtering

$$\hat{x}(i) = \frac{1}{C_i}\sum_j y(j) e^{-\frac{\|i-j\|^2}{2\rho^2}} e^{-\frac{|y(i)-y(j)|^2}{2\sigma^2}}$$



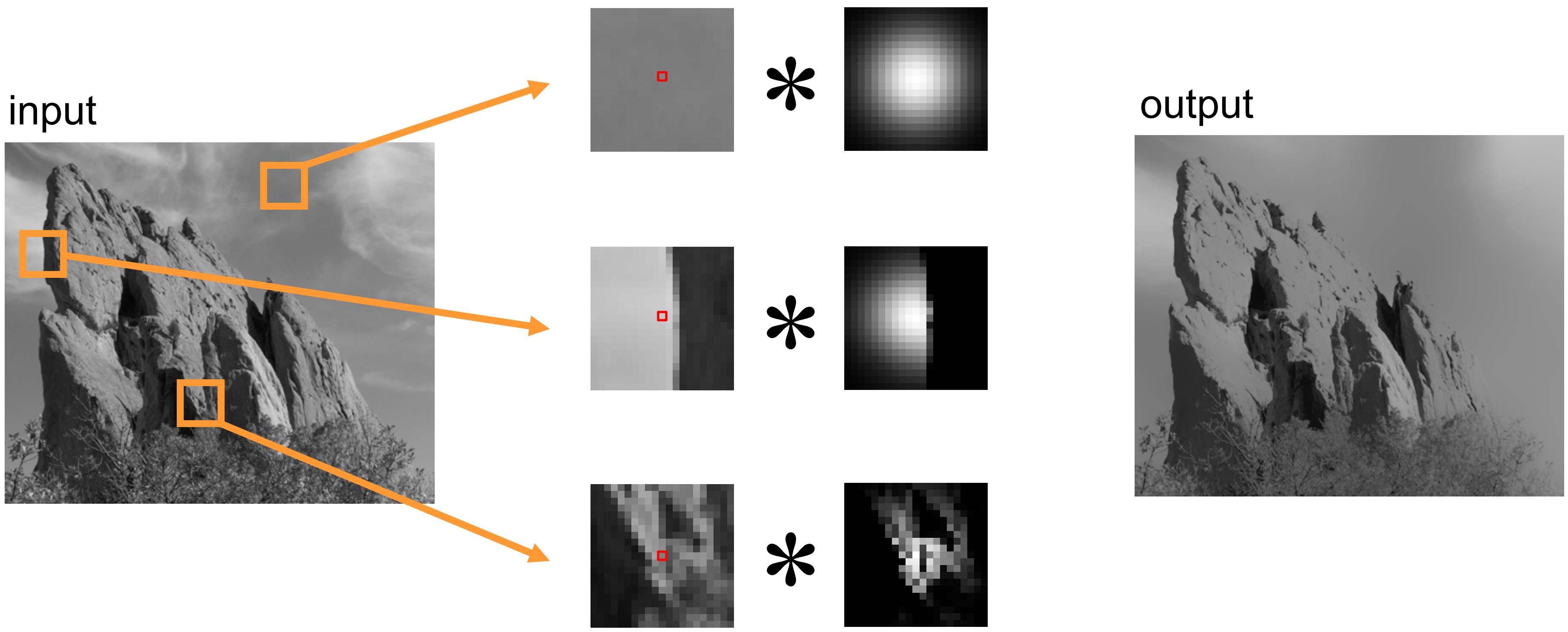
Smith and Brady (1997)

Gaussian Smoothing



Same Gaussian kernel everywhere
Averages across edges ⇒ blur

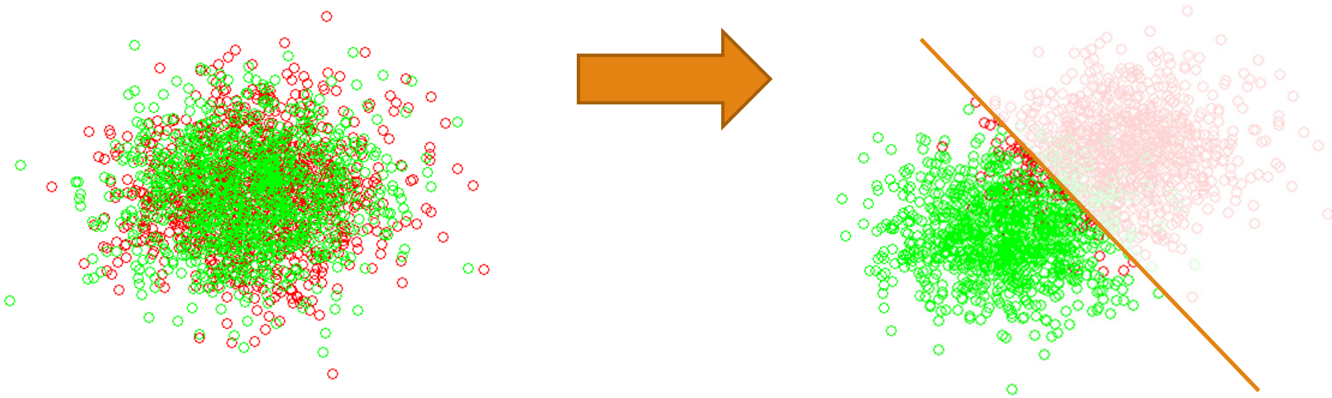
Bilateral Filtering



Kernel shape depends on image content
Avoids averaging across edges

Denoising in the Transform Domain

Motivation – New representation where signal and noise are more separated

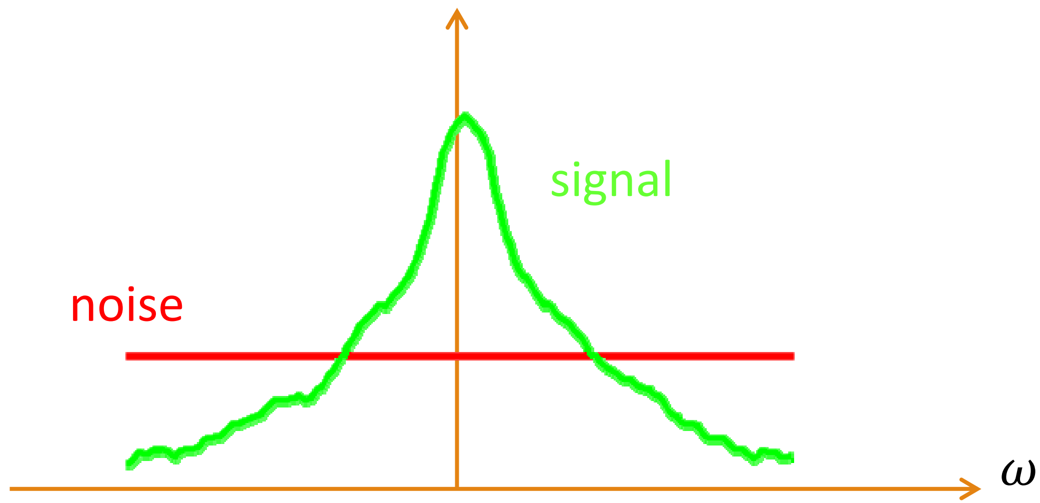


Denoise = "Suppress noise coefficients while preserving the signal coefficients"

Fourier Domain

<u>Noise</u>
White ⇒ spread uniformly in Fourier domain

<u>Signal</u>
Spread non-uniformly in the Fourier domain

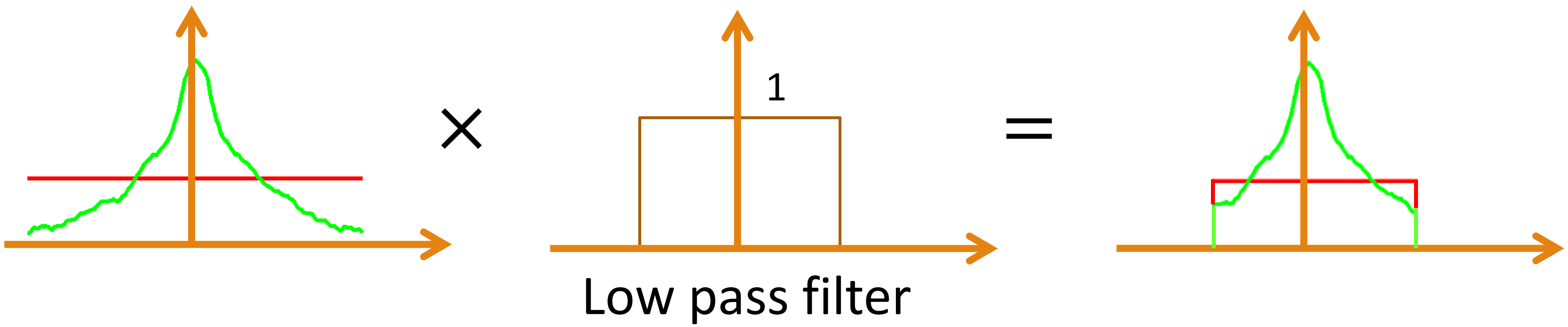


Low-Pass Filtering

Low pass with some cut-off frequency

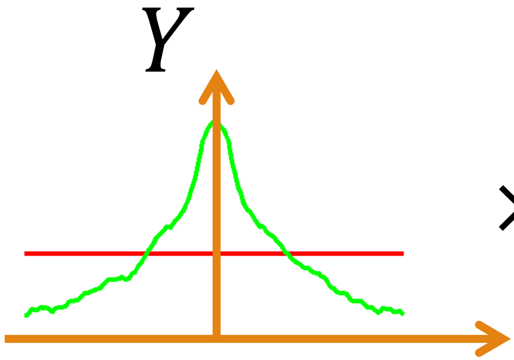
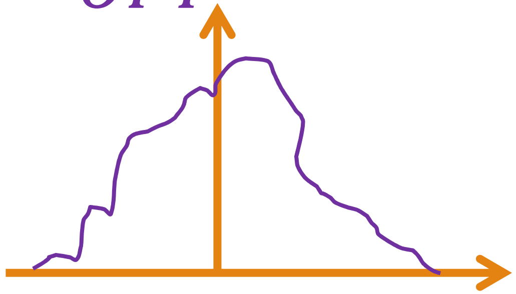
Keeps most of the signal energy

Equivalent to Global Smoothing



Looking for an Optimal Filter

$$\hat{X}(\omega) = Y(\omega)H(\omega)$$

$\hat{X} =$  $\times$ 

Assumption: Signal and Noise are Stationary independent random processes

Spatial filtering restoration

The noise is assumed to be uncorrelated with the original image:

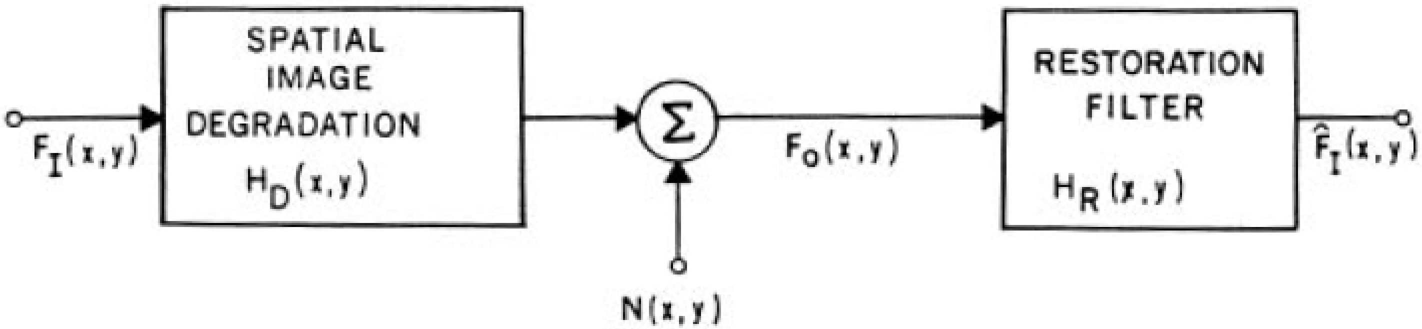
$$F_O(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_I(\alpha,\beta)\, H_D(x-\alpha, y-\beta)\, d\alpha\, d\beta + N(x,y)$$

Or:

$$F_O(x,y) = F_I(x,y) \circledast H_D(x,y) + N(x,y)$$

The restoration is a LTI filter:

$$\hat{F}_I(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_O(\alpha,\beta) H_R(x-\alpha, y-\beta)\, d\alpha\, d\beta$$



Spatial filter restoration

Substituting

By Fourier transform:

$$\hat{F}_I(x, y) = [F_I(x, y) \circledast H_D(x, y) + N(x, y)] \circledast H_R(x, y)$$

$$\hat{\mathcal{F}}_I(\omega_x, \omega_y) = [\mathcal{F}_I(\omega_x, \omega_y)\mathcal{H}_D(\omega_x, \omega_y) + \mathcal{N}(\omega_x, \omega_y)]\mathcal{H}_R(\omega_x, \omega_y)$$

Inverse Filter

In this case the restoration filter is chosen so that:

$$\mathcal{H}_R(\omega_x, \omega_y) = \frac{1}{\mathcal{H}_D(\omega_x, \omega_y)}$$

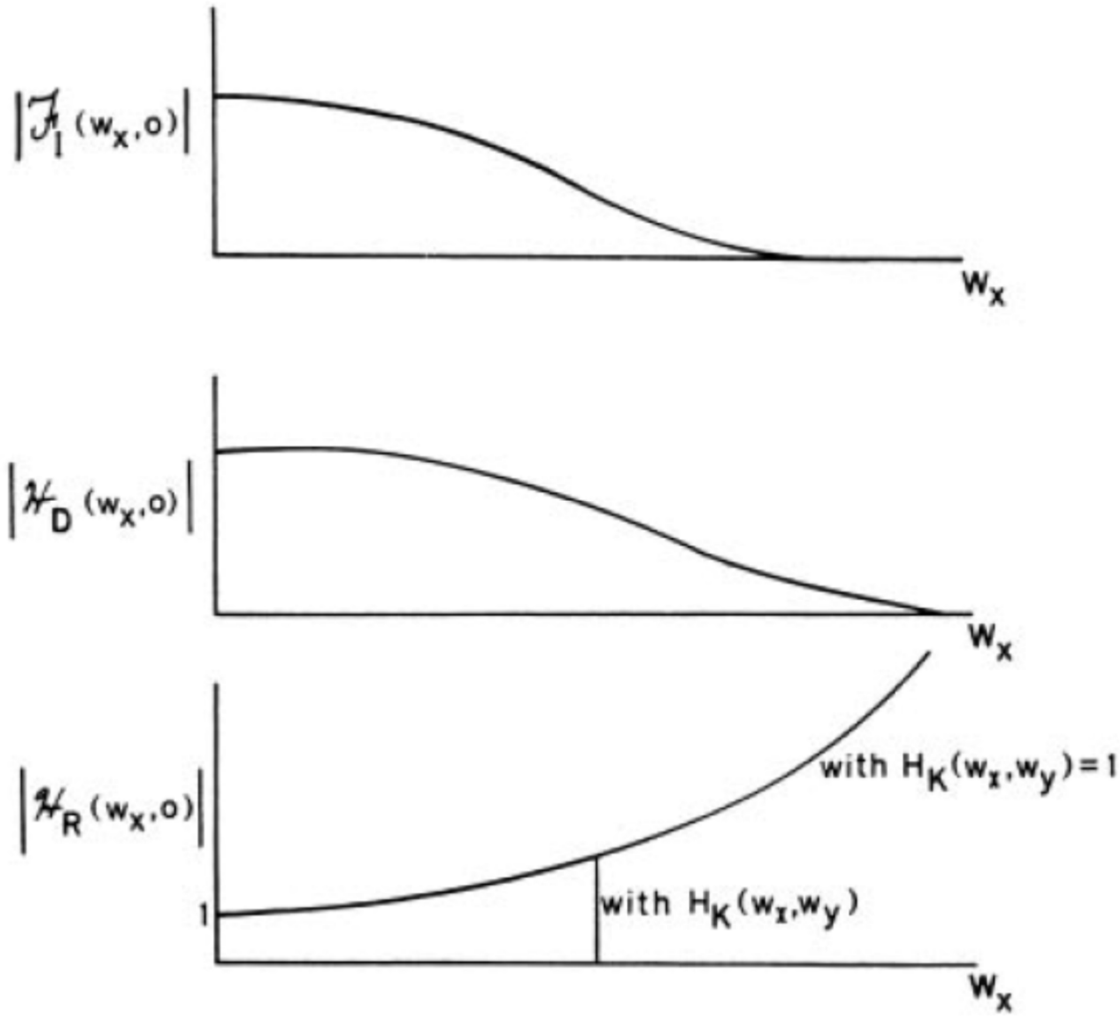
The spectrum of the reconstructed image becomes:

$$\hat{\mathcal{F}}_I(\omega_x, \omega_y) = \mathcal{F}_I(\omega_x, \omega_y) + \frac{\mathcal{N}(\omega_x, \omega_y)}{\mathcal{H}_D(\omega_x, \omega_y)}$$

The restored image field becomes:

$$\hat{F}_I(x, y) = F_I(x, y) + \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\mathcal{N}(\omega_x, \omega_y)}{\mathcal{H}_D(\omega_x, \omega_y)} \exp\{i(\omega_x x + \omega_y y)\}\, d\omega_x\, d\omega_y$$

The inverse filter



The Wiener filter

In the general derivation of the Wiener filter we assume that the ideal image $F_I(x,y)$ and the observed image $F_O(x,y)$ are samples of a two dimensional continuous stochastic field with zero value spatial mean.

Wiener imposes the minimization of the mean-square restoration error:

$$\mathcal{E} = \mathbb{E}\left\{\left[F_I(x,y) - \hat{F}_I(x,y)\right]^2\right\}$$

Wiener criterium for Restoration filter

$$\mathcal{E} = \mathbb{E}\left\{\left[F_I(x,y) - \hat{F}_I(x,y)\right]^2\right\} =$$

$$= \mathbb{E}\left\{\left[F_I(\omega_x,\omega_y) - \hat{F}_I(\omega_x,\omega_y)\right]^2\right\} =$$

$$= \mathbb{E}\left\{\left[F_I - \left(\mathcal{F}_I\mathcal{H}_D + \mathcal{N}\right)\mathcal{H}_R\right]^2\right\}$$

Minimization by derivation

$$\frac{\partial\mathcal{E}}{\partial\mathcal{H}_R} = \mathbb{E}\left\{\frac{\partial}{\partial\mathcal{H}_R}\left[F_I - \left(F_I\mathcal{H}_D + \mathcal{N}\right)\mathcal{H}_R\right]^2\right\} =$$

$$\mathbb{E}\left\{2\left(F_I(\omega_x,\omega_y) - \hat{F}_I(\omega_x,\omega_y)\right)\left(-\hat{F}_O(\omega_x,\omega_y)\right)\right\} = 0$$

Orthogonality condition

The mean square error is minimized when the following orthogonality condition is met for all image points:

$$\mathbb{E}\left\{\left[F_I\left(\omega_x,\omega_y\right)-\hat{F}_I\left(\omega_x,\omega_y\right)\right]F_O\left(\omega_x,\omega_y\right)\right\}=0$$

Considering:

$$\hat{F}_I(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}F_O(\alpha,\beta)H_R(x-\alpha,y-\beta)\,d\alpha\,d\beta$$

We obtain:

$$\mathbb{E}\left\{F_I(x,y)F_O(x,y)\right\}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\mathbb{E}\left\{F_O(\alpha,\beta)F_O(x,y)\right\}\,H_R(x-\alpha,y-\beta)\,d\alpha\,d\beta$$

Wiener formulation (spatial domain)

If the images are assumed as jointly stationary processes the expectation value can be expressed as covariance functions:

$$\mathbb{E}\{F_I(x,y)F_O(x,y)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbb{E}\{F_O(\alpha,\beta)F_O(x',y')\}\, H_R(x-\alpha, y-\beta)\, d\alpha\, d\beta$$

$$\mathbb{E}\{F_I(x,y)F_O(x,y)\}\, K_{F_I F_O}$$

$$K_{F_I F_O}(x-x', y-y') = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} K_{F_O F_O}(\alpha-x', \beta-y')H_R(x-\alpha, y-\beta)\, d\alpha\, d\beta$$

Wiener formulation

The Fourier transform of the covariance function is the Power Spectral Density (PSD), or power spectrum.

Taking the two-dimensional Fourier transform:

$$\mathcal{H}_R(\omega_x, \omega_y) = \frac{\mathcal{W}_{F_I F_O}(\omega_x, \omega_y)}{\mathcal{W}_{F_O F_O}(\omega_x, \omega_y)}$$

Wiener formulation

Since:

$$K(x_1, y_1, t_1; x_2, y_2, t_2) = E\{[F(x_1, y_1, t_1) - \eta_F(x_1, y_1, t_1)][F^*(x_2, y_2, t_2) - \eta_F^*(x_2, y_2, t_2)]\}$$

And:

$$\mathcal{W}_{F_I F_O}(\omega_x, \omega_y) = \mathcal{H}_D^*(\omega_x, \omega_y)\mathcal{W}_{F_I}(\omega_x, \omega_y)$$

since the cross-correlation between $F_I(x,y)$ and $N(x,y)$ is a zero matrix.

We obtain:

$$\mathcal{W}_{F_O F_O}(\omega_x, \omega_y) = \left|\mathcal{H}_D(\omega_x, \omega_y)\right|^2 \mathcal{W}_{F_I}(\omega_x, \omega_y) + \mathcal{W}_N(\omega_x, \omega_y)$$

Wiener filter model

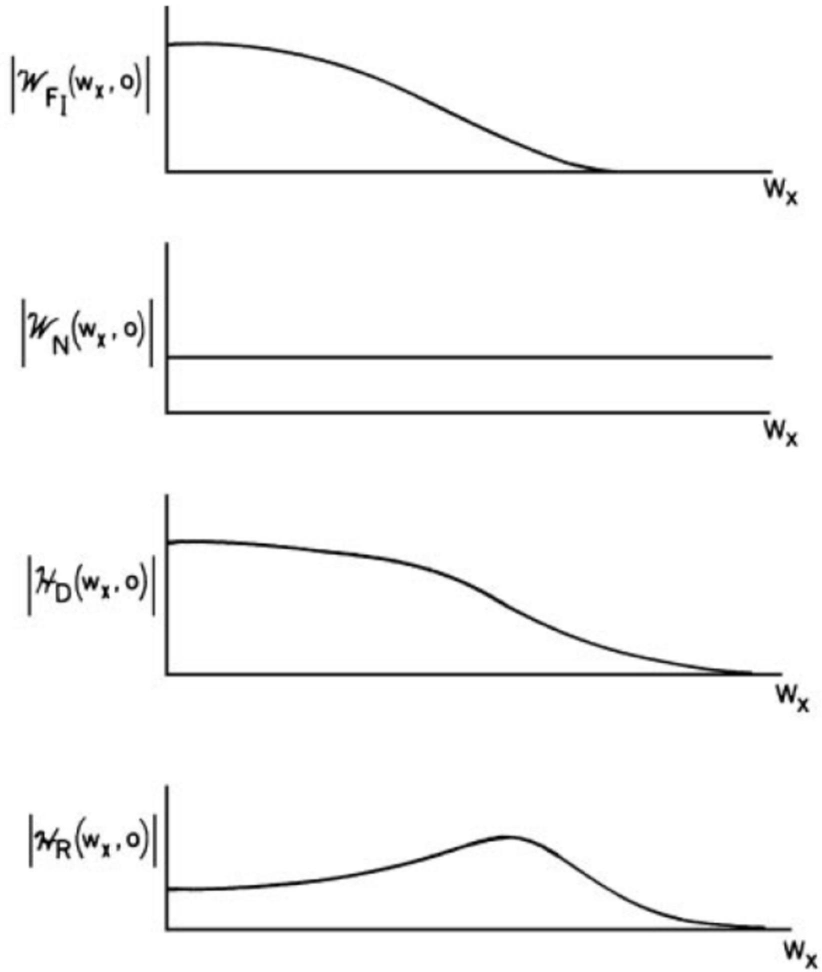
The resulting filter is then:

$$\mathcal{H}_R(\omega_x, \omega_y) = \frac{\mathcal{H}_D^*(\omega_x, \omega_y)\mathcal{W}_{F_I}(\omega_x, \omega_y)}{\left|\mathcal{H}_D(\omega_x, \omega_y)\right|^2 \mathcal{W}_{F_I}(\omega_x, \omega_y) + \mathcal{W}_N(\omega_x, \omega_y)}$$

Or:

$$\mathcal{H}_R(\omega_x, \omega_y) = \frac{\mathcal{H}_D^*(\omega_x, \omega_y)}{\left|\mathcal{H}_D(\omega_x, \omega_y)\right|^2 + \mathcal{W}_N(\omega_x, \omega_y)/\mathcal{W}_{F_I}(\omega_x, \omega_y)}$$

Wiener results



Degradation & Restoration Examples (Atmospheric Turbulence Model)

$$H(u,v) = e^{-k\left[(u+M/2)^2+(v-N/2)^2\right]^{5/6}}$$

FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

Degradation & Restoration Examples (inverse filter)

$$H(u,v) = e^{-k\left[(u+M/2)^2+(v-N/2)^2\right]^{5/6}}$$

a	b
c	d

FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with H cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.

Degradation & Restoration Examples: Gonzalez & Woods

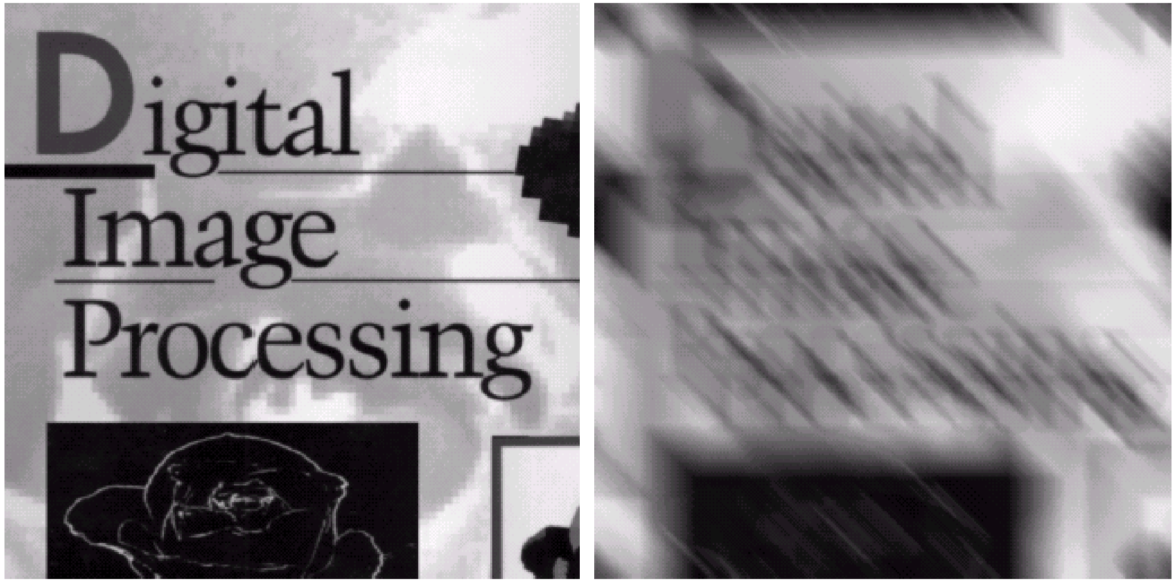
a b c

FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

$$H(u,v) = e^{-k\left[(u+M/2)^2+(v-N/2)^2\right]^{5/6}}$$

$$\begin{aligned}\hat{F}(u,v) &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)\left|H(u,v)\right|^2 + S_\eta(u,v)}\right]G(u,v) \\ &= \left[\frac{H^*(u,v)}{\left|H(u,v)\right|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v) \\ &= \left[\frac{1}{H(u,v)}\frac{\left|H(u,v)\right|^2}{\left|H(u,v)\right|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)\end{aligned}$$

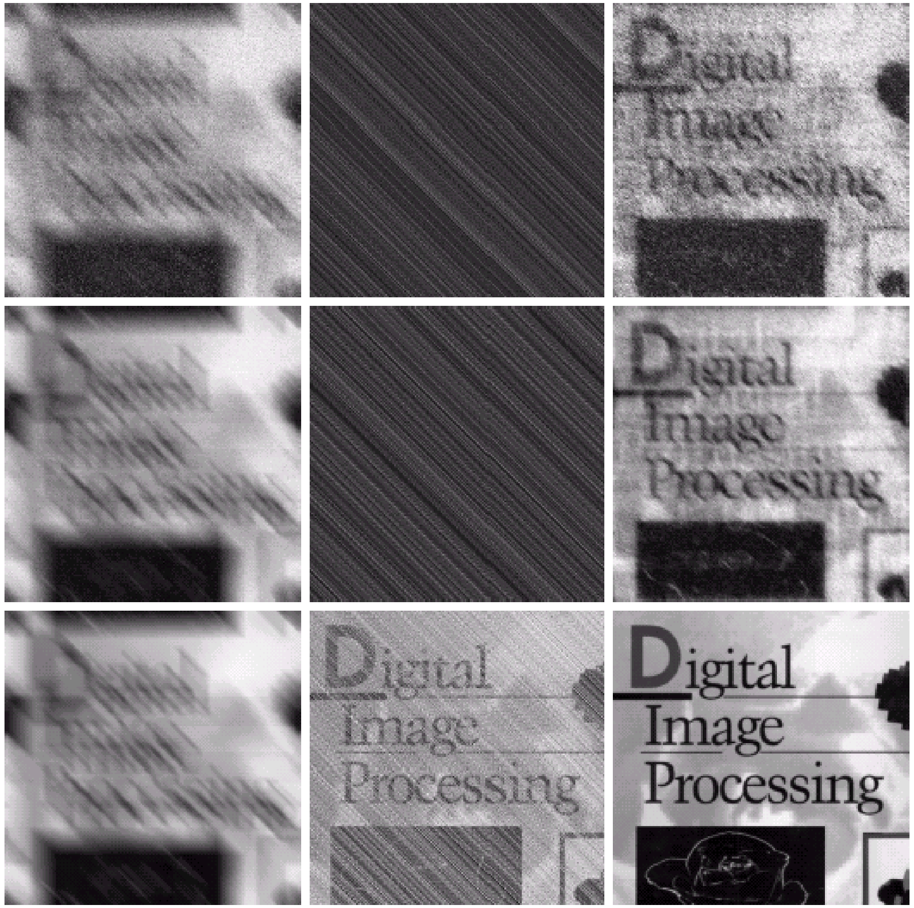
Degradation & Restoration Examples (Planar motion)



a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

Degradation & Restoration (inverse and Wiener Filters)



a b c
d e f
g h i

FIGURE 5.29 (a) Image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a "curtain" of noise.

Wiener Deconvolution

Wiener deconvolution can be used effectively when the frequency characteristics of the image and additive noise are known, to at least some degree. In the absence of noise, the Wiener filter reduces to the ideal inverse filter.

Parametric estimation filter

Cole proposed a restoration filter with a transfer function:

$$\mathcal{H}_R(\omega_x, \omega_y) = \left[\frac{\mathcal{W}_{F_I}(\omega_x, \omega_y)}{\left|\mathcal{H}_D(\omega_x, \omega_y)\right|^2 \mathcal{W}_{F_I}(\omega_x, \omega_y) + \mathcal{W}_N(\omega_x, \omega_y)}\right]^{1/2}$$

The power spectrum of the filter output is:

$$\mathcal{W}_{\hat{F}_I}(\omega_x, \omega_y) = \left|\mathcal{H}_R(\omega_x, \omega_y)\right|^2 \mathcal{W}_{F_O}(\omega_x, \omega_y)$$

where

$$\mathcal{W}_{F_O}(\omega_x, \omega_y) = \left|\mathcal{H}_D(\omega_x, \omega_y)\right|^2 \mathcal{W}_{F_I}(\omega_x, \omega_y) + \mathcal{W}_N(\omega_x, \omega_y)$$

Parametric estimation filter

The previous filter gives a reconstructed image with the same power spectrum of the original image:

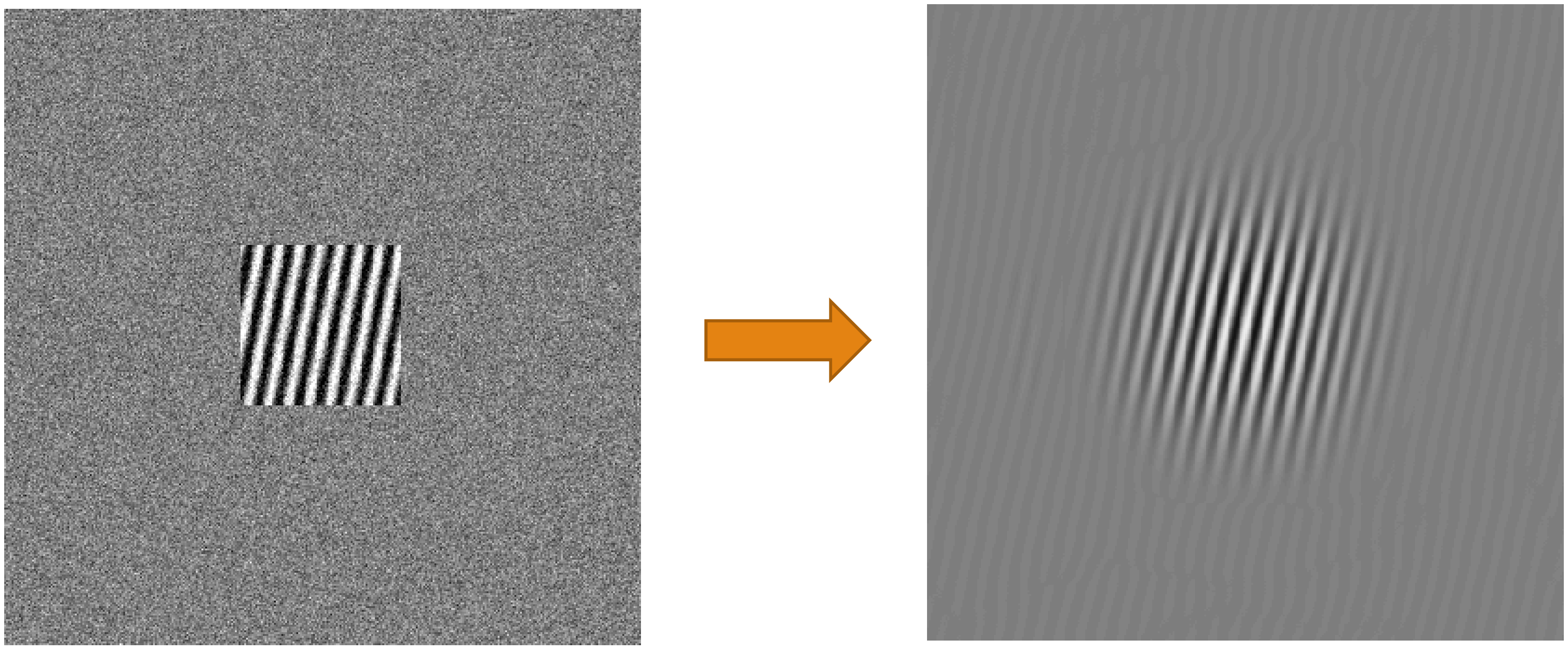
$$\mathcal{W}_{\hat{F}_I}(\omega_x, \omega_y) = \mathcal{W}_{F_I}(\omega_x, \omega_y)$$

And it is called the power spectrum filter, while for the Wiener filter:

$$\mathcal{W}_{\hat{F}_I}(\omega_x, \omega_y) = \frac{\left|\mathcal{H}_D(\omega_x, \omega_y)\right|^2 [\mathcal{W}_{F_I}(\omega_x, \omega_y)]^2}{\left|\mathcal{H}_D(\omega_x, \omega_y)\right|^2 \mathcal{W}_{F_I}(\omega_x, \omega_y) + \mathcal{W}_N(\omega_x, \omega_y)}$$

Why isn't it enough?

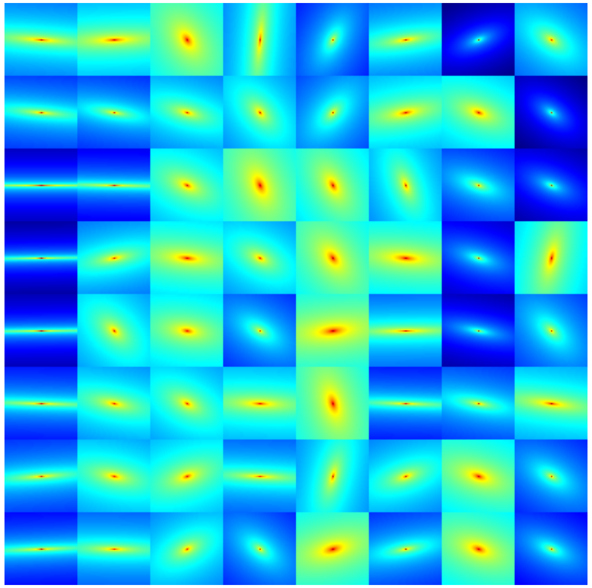
Mismatches and errors ⇒ global artifacts



The Windowed Fourier & Wiener Filter

Image has a local structure

⇒ Denoise each region based on its own statistics



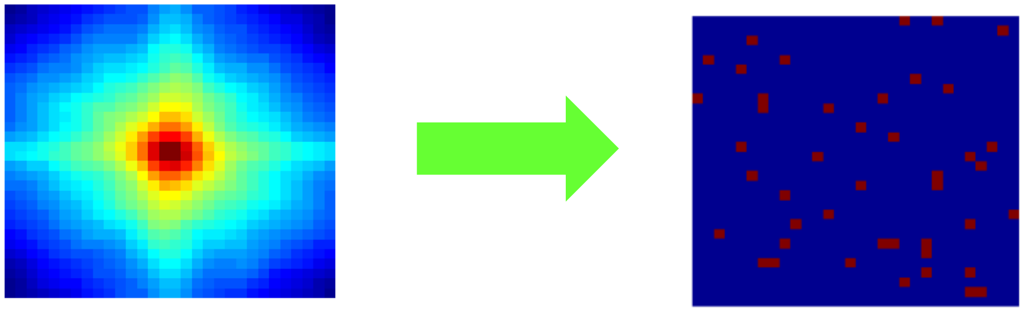
Can we do better?

Why restrict ourselves to a Fourier basis?

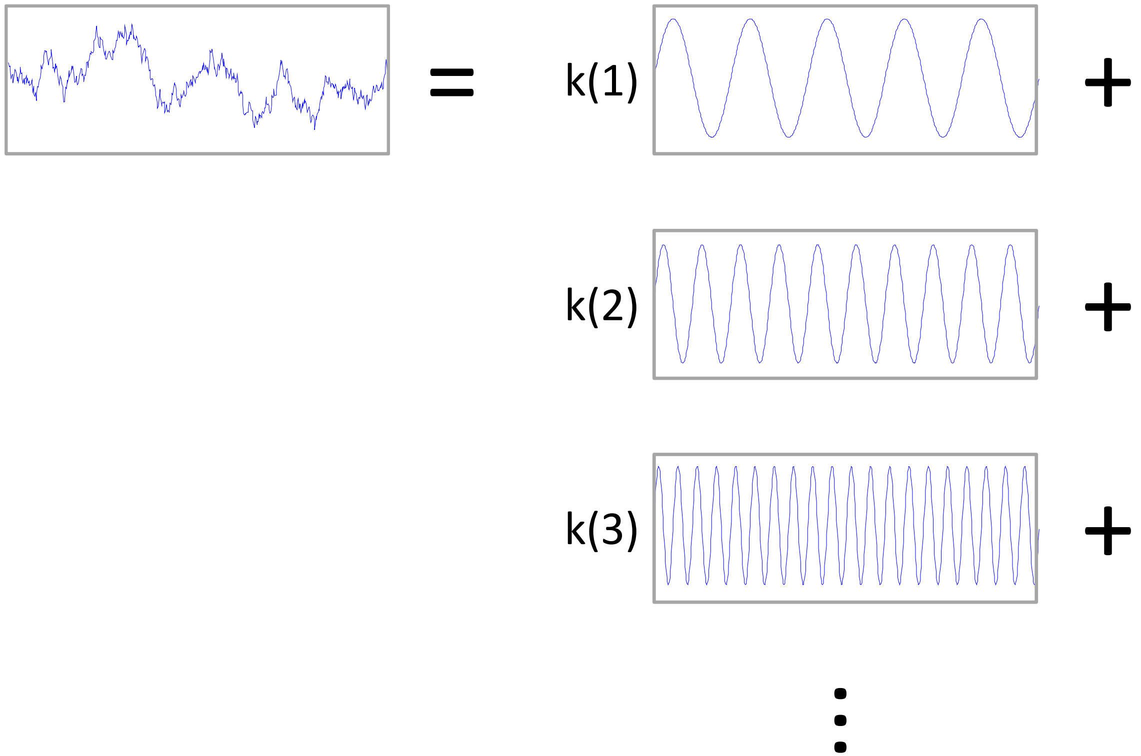
Other representations can be better:

- Sparsity ⇒ Signal/Noise separation
- Localization of image details

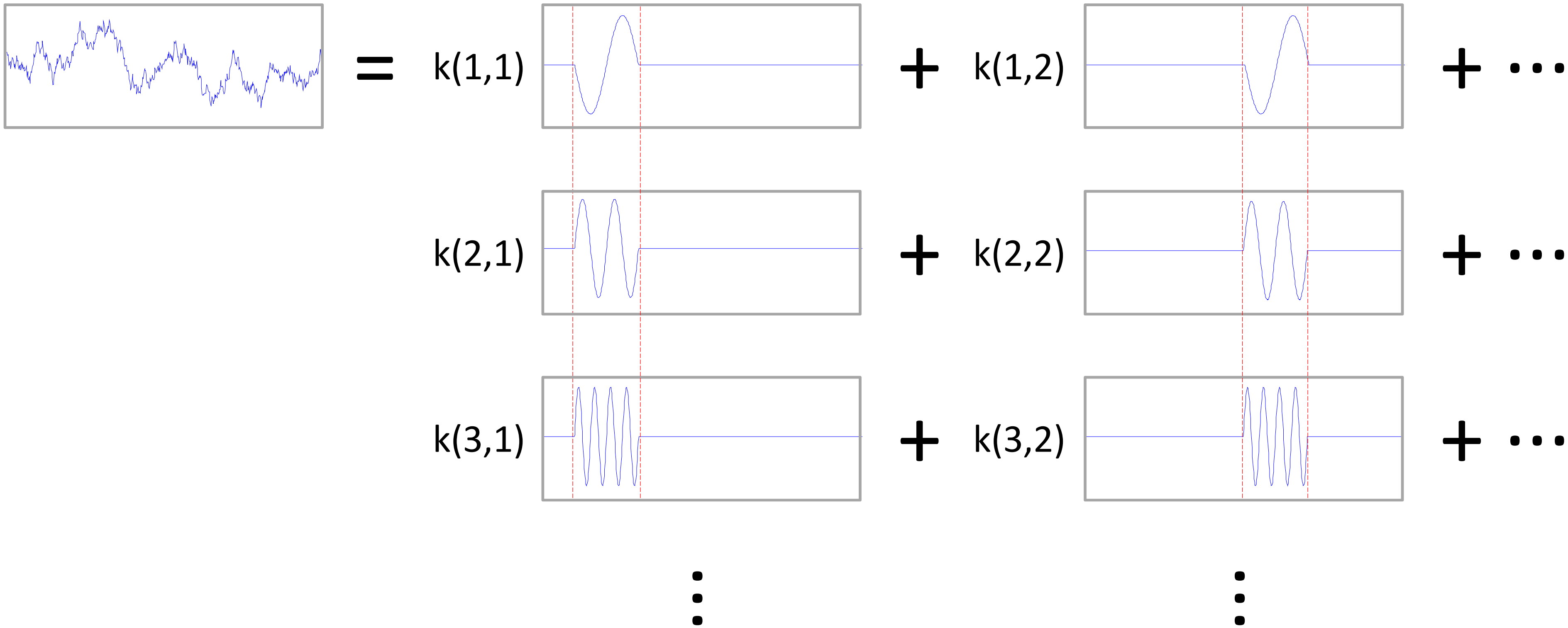
Wavelets



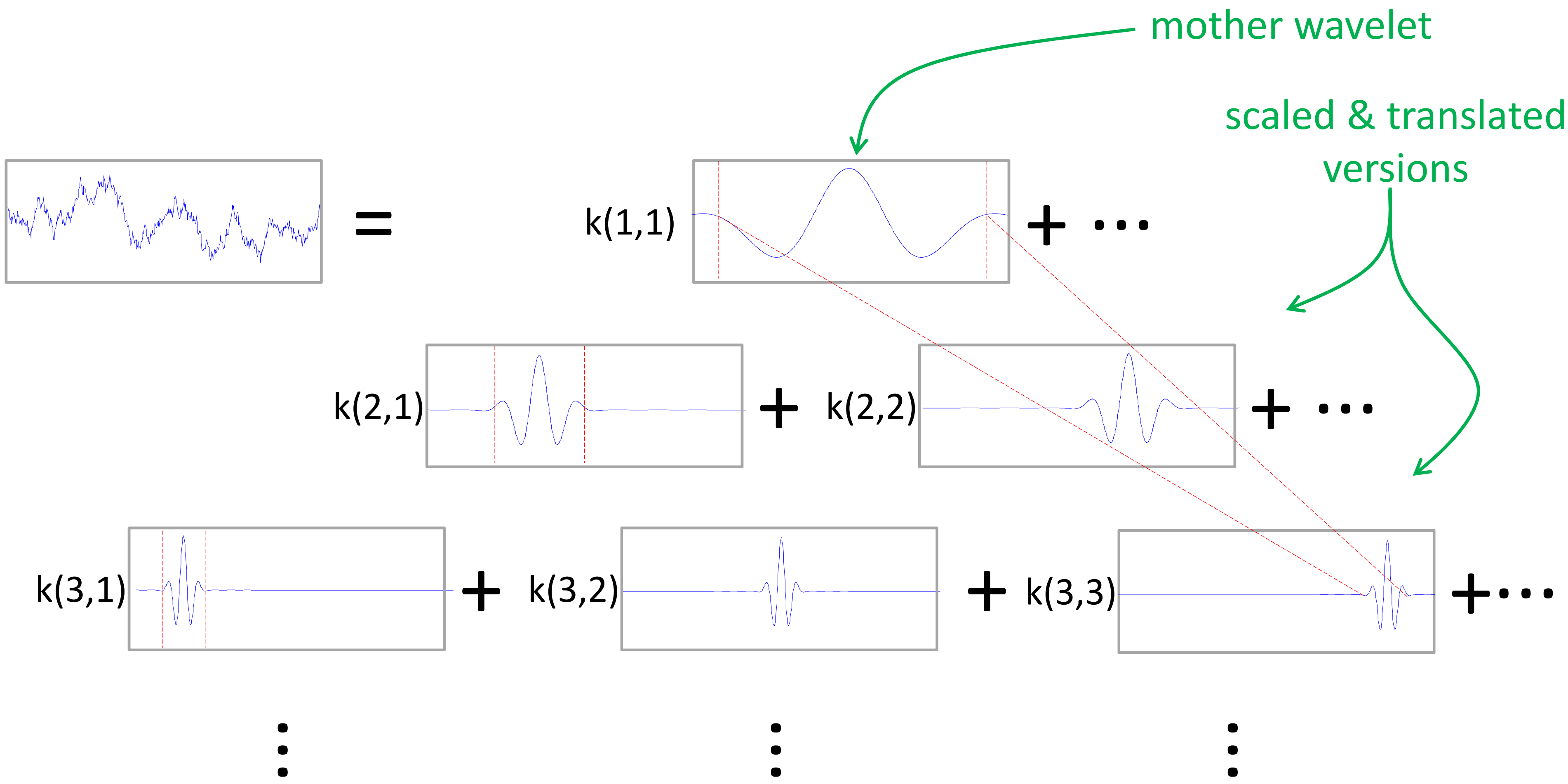
Fourier Decomposition



Windowed Fourier Decomposition

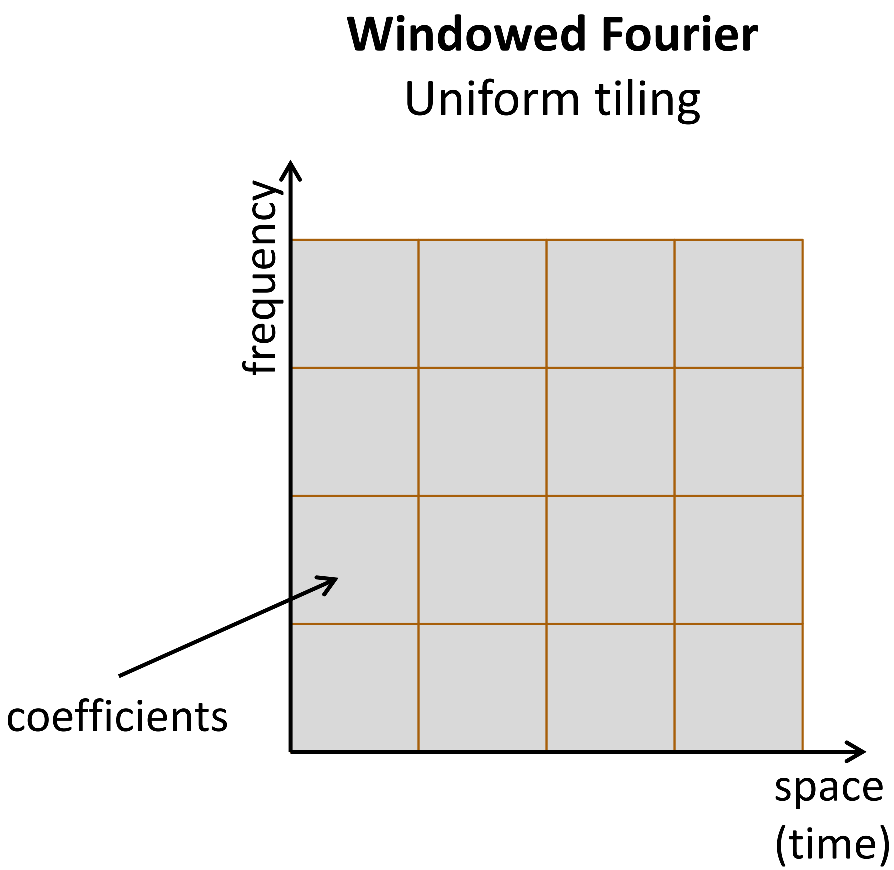


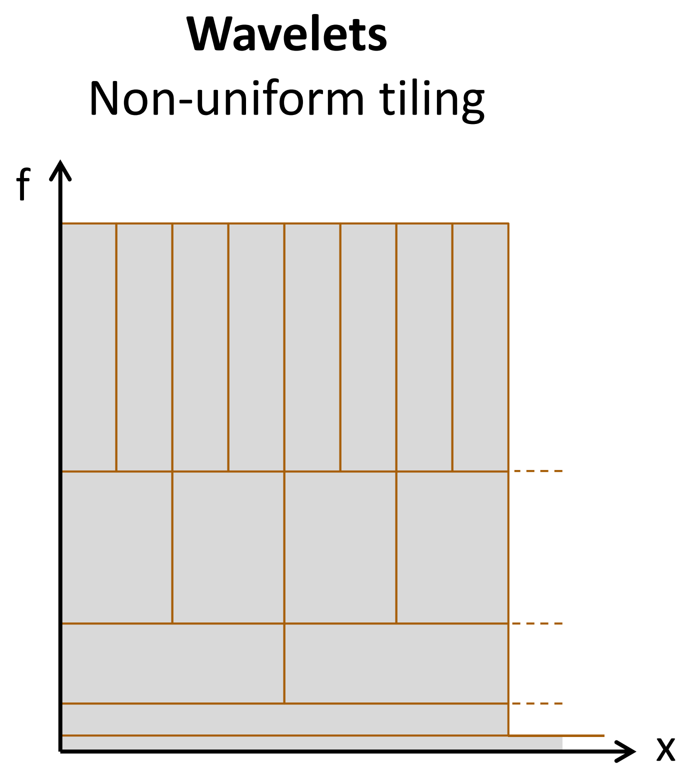
Wavelet Decomposition



Space-Frequency Localization

Better distribution of the "Coefficient Budget"

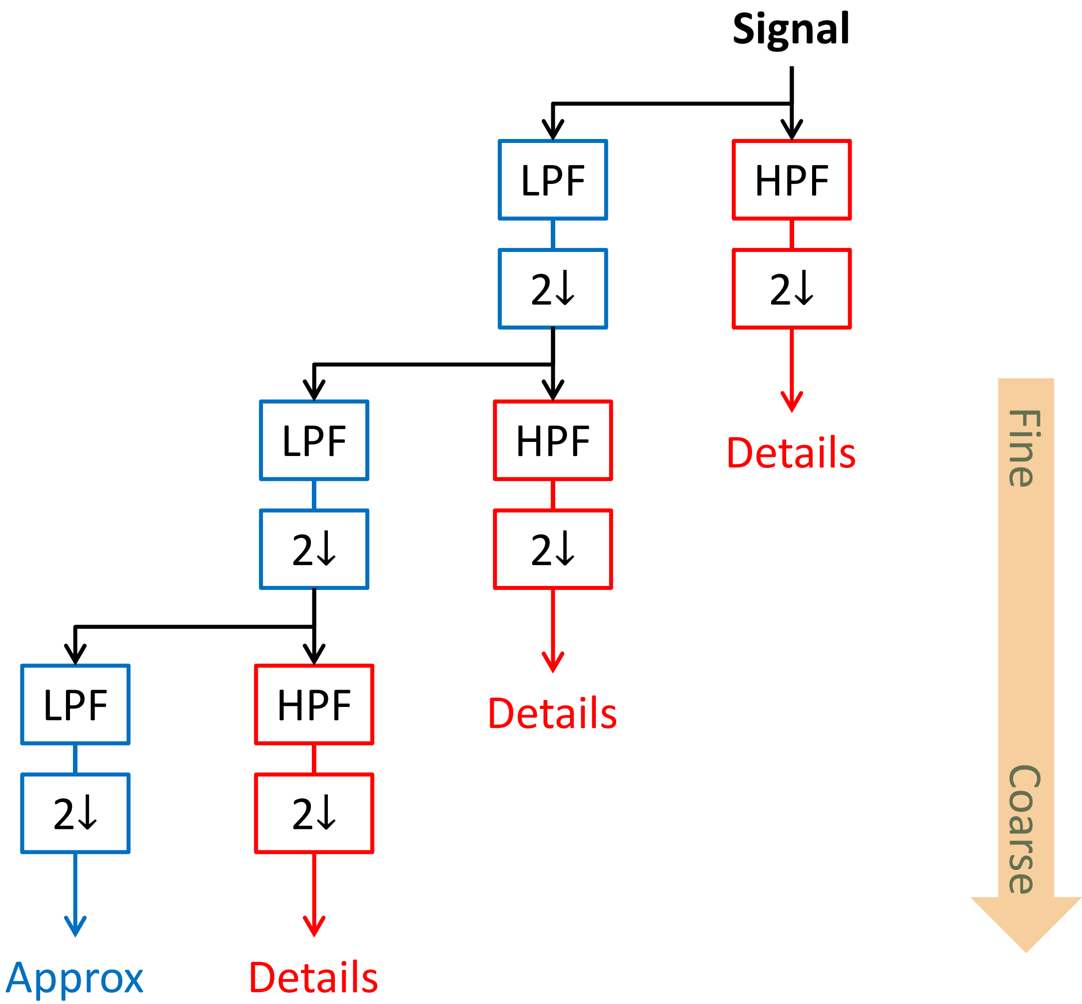




Discrete Wavelet Transform (DWT)

Recursively, split to

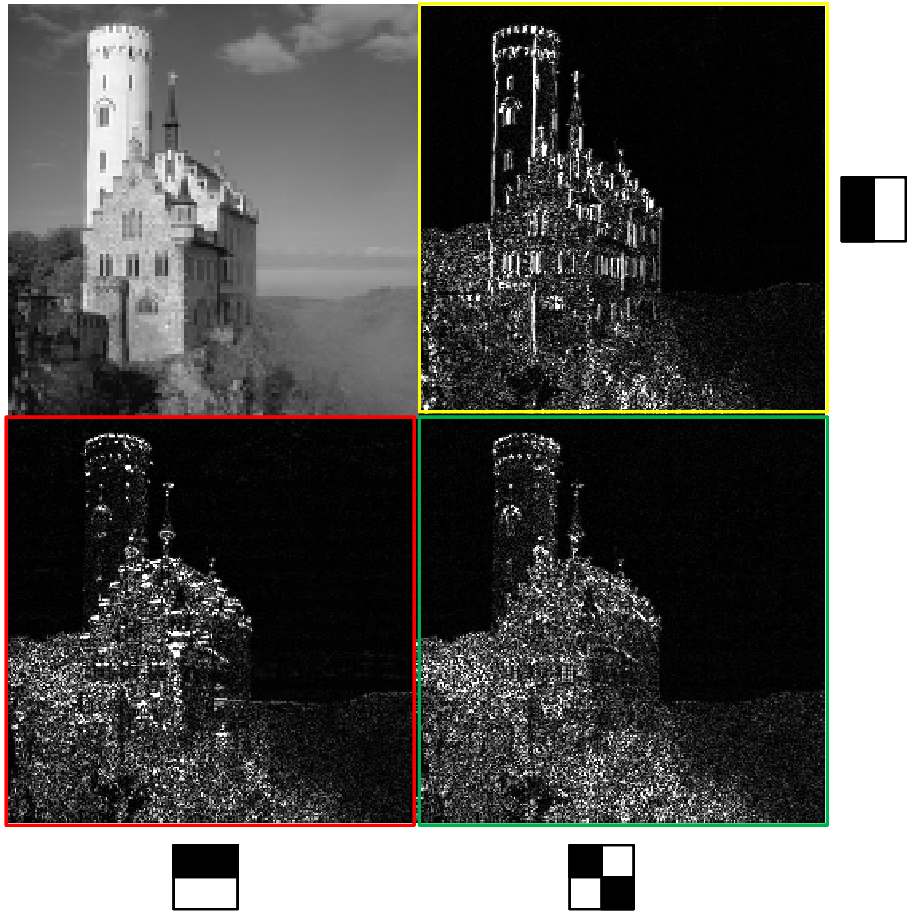
- Approximation
- Details



Wavelet Transform - Example

Original image

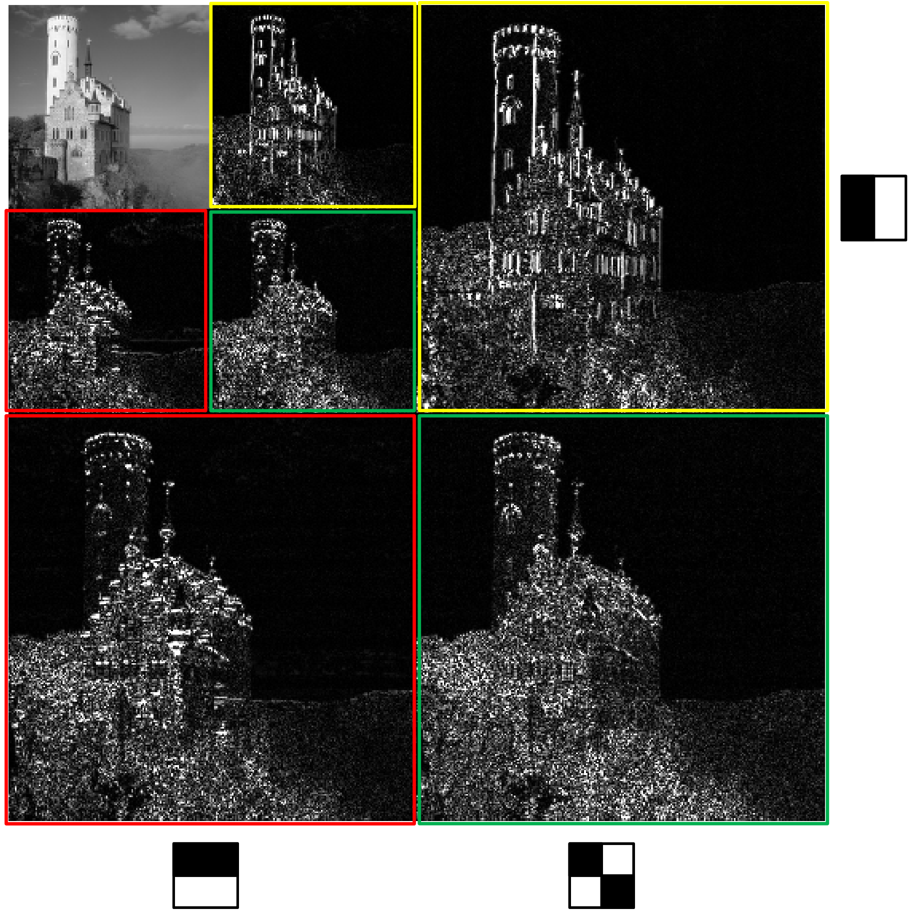
1 level DWT



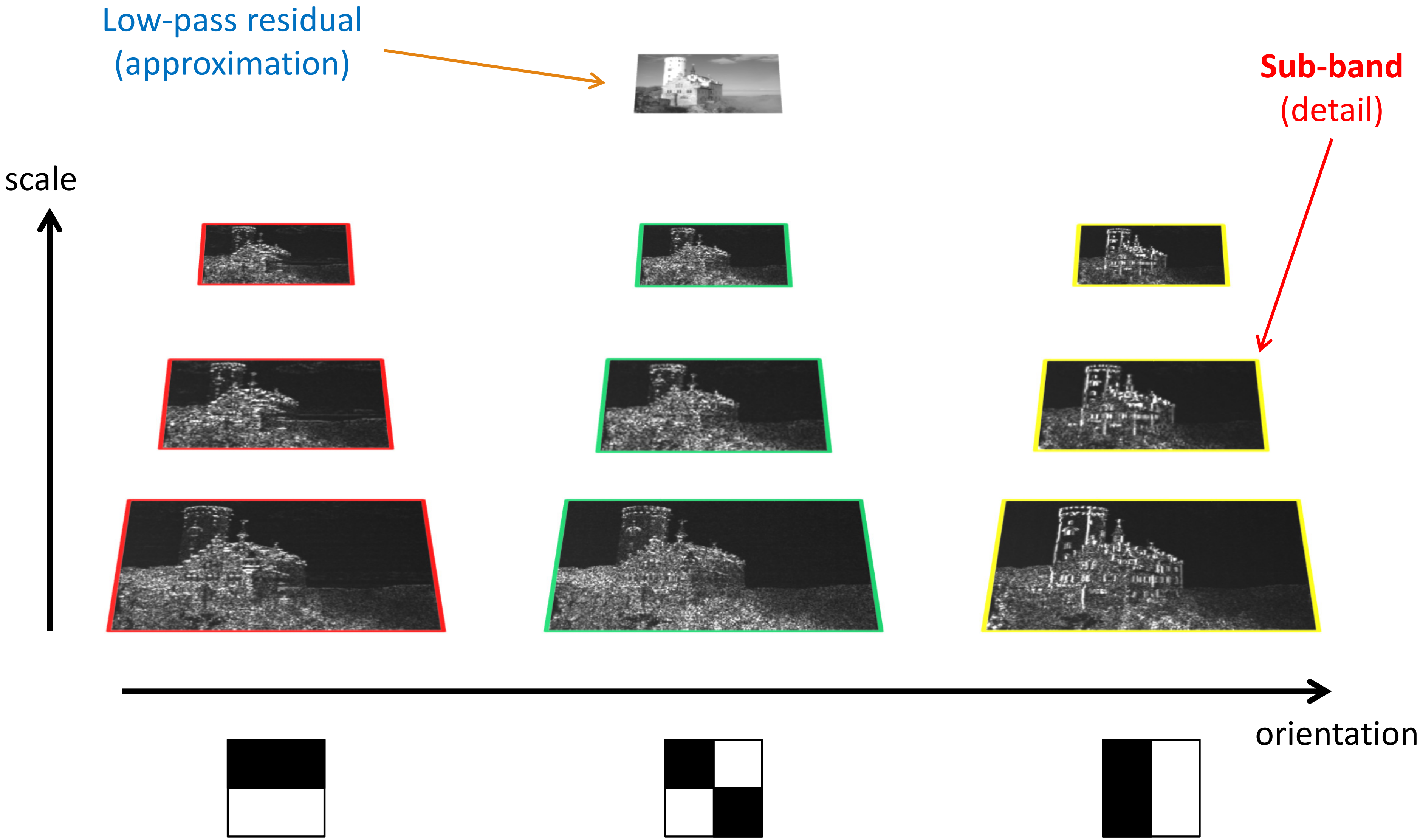
Wavelet Transform - Example

Original image

2 level DWT



Wavelet Pyramid

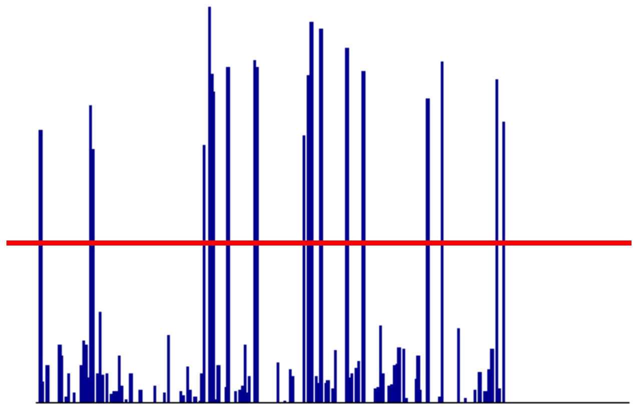


Wavelet Thresholding (WT)

Wavelet $\Rightarrow$ Sparser Representation

Improved separation between signal and noise at different scales and orientations

Thresholding (hard/soft) is more meaningful



Performance Evaluation Denoised Images

Original
$\sigma = 20$

Gaussian
Smoothing

Anisotropic
Filtering

Bilateral
Filtering

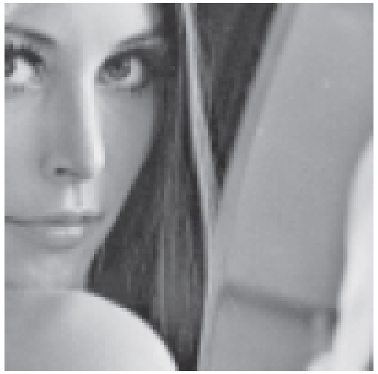
Windowed
Weiner

Hard WT

Soft WT

Buades *et al*. (2005)

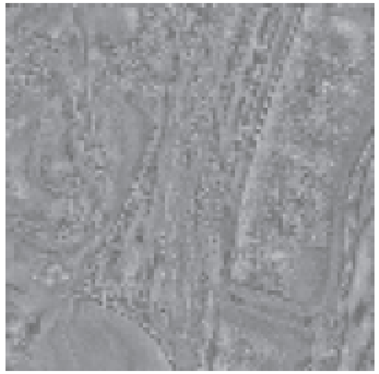
Performance Evaluation
Method Noise

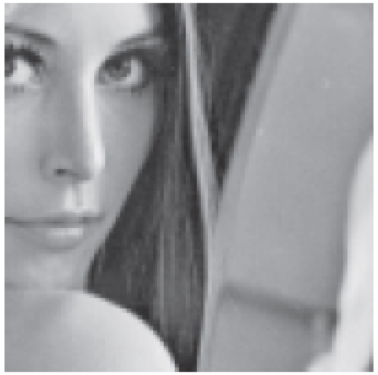


Gaussian Smoothing

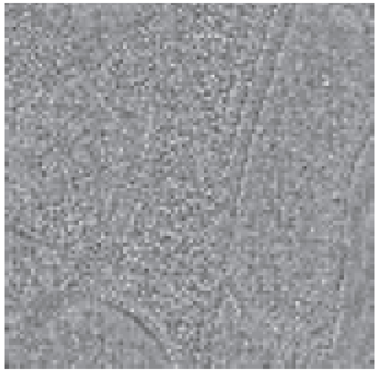
Anisotropic Filtering

Bilateral Filtering





Windowed Weiner



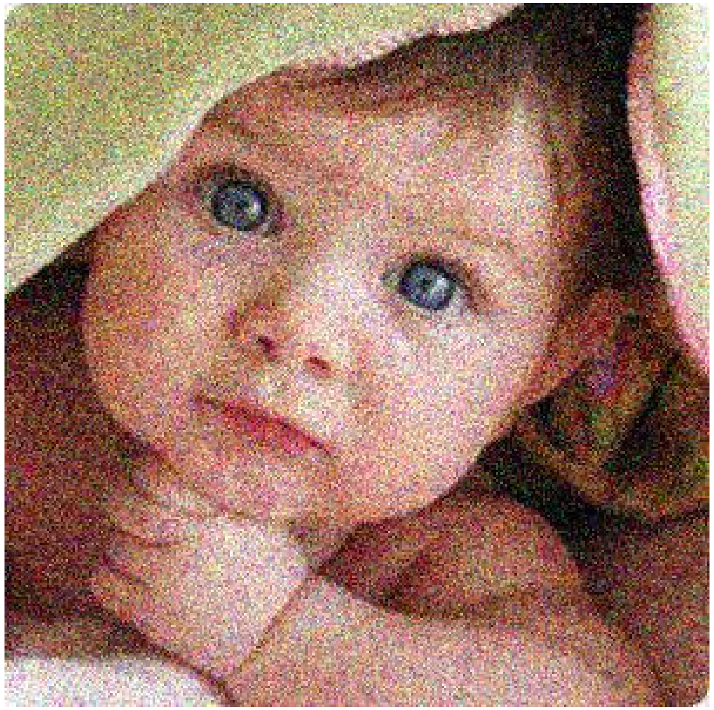
Hard WT

Soft WT

Buades *et al*. (2005)

A Probabilistic Perspective

Which image do you prefer?



A Probabilistic Perspective

With some prior knowledge about images

Denoise = “find an optimal explanation”:

- **MAP** – Maximum a posterior

$$\hat{x} = \mathrm{argmax}_x p(x|y)$$

- **MMSE** – Minimum Mean Square Error

$$\hat{x} = \mathrm{argmin}_{\hat{x}} E\{(\hat{x}(y) - x)^2\} = E(x|y)$$

Motivation - Drawback of Locality

Previous methods perform some local filtering

$\Rightarrow$ mixing of pixels from different statistics

$\Rightarrow$ blur

Goal:

Reduce the mixing $\Leftrightarrow$ "smarter" localization

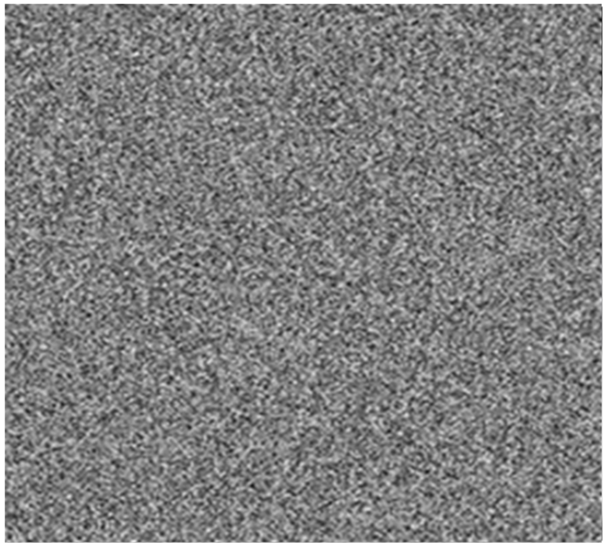
Motivation - Temporal perspective

Assume a static scene

Consider multiple images $y(t)$ at different times

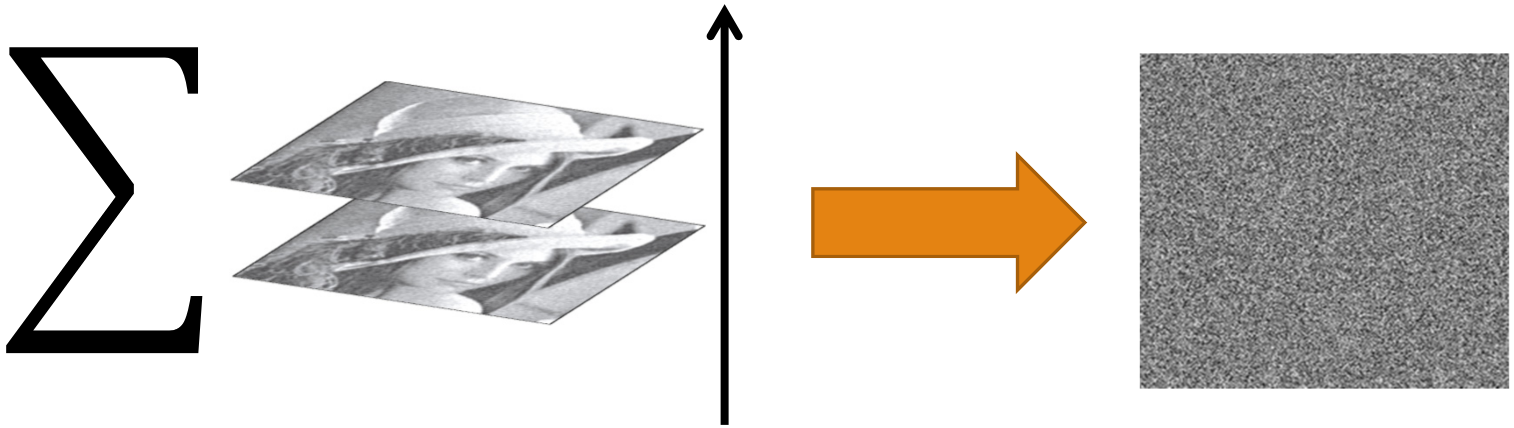
The signal $x(t)$ remains constant

$n(t)$ varies over time with zero mean



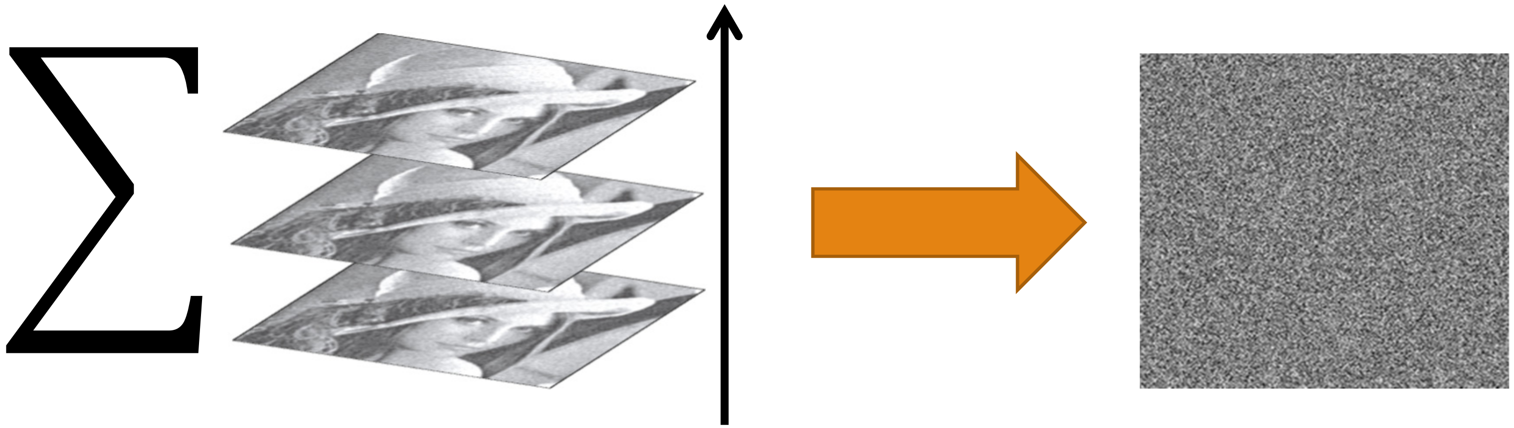
“Temporal Denoising”

Average multiple images over time



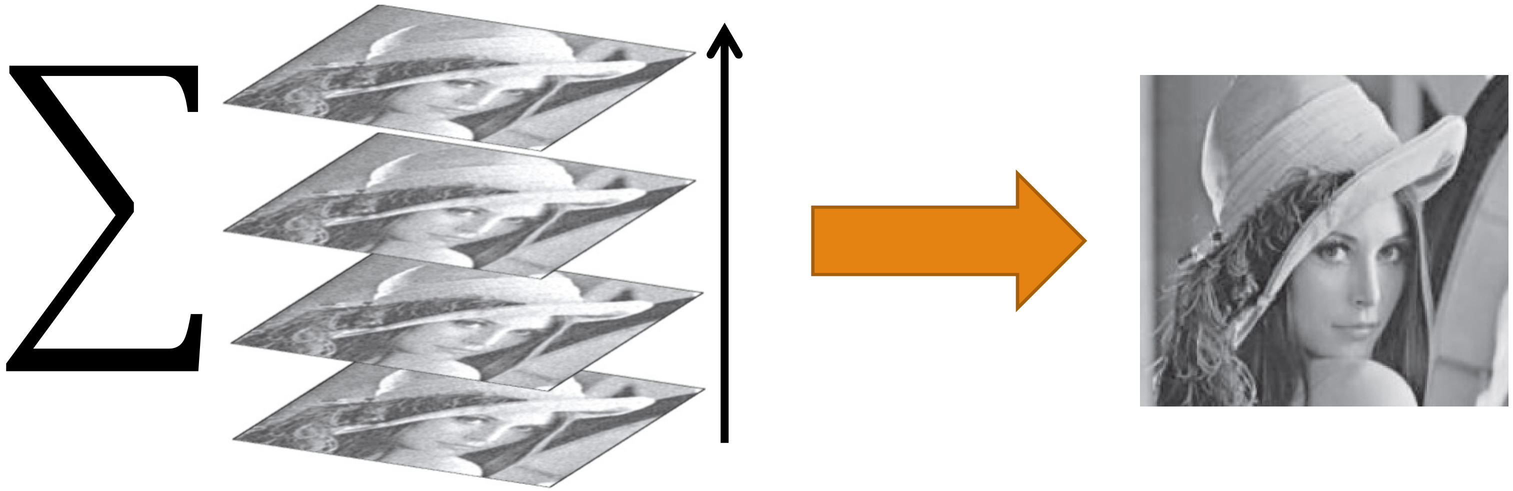
“Temporal Denoising”

Average multiple images over time

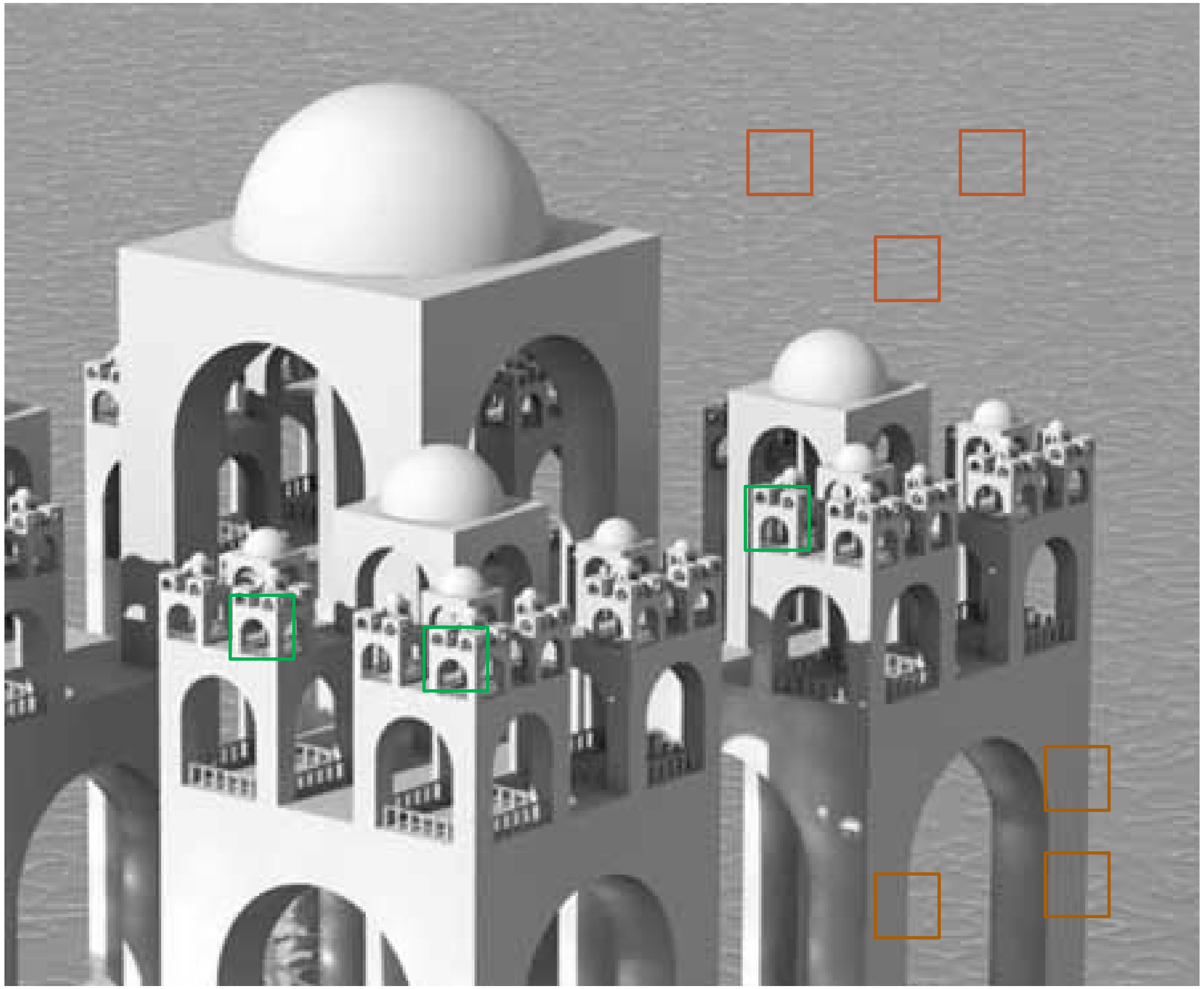


“Temporal Denoising”

Average multiple images over time

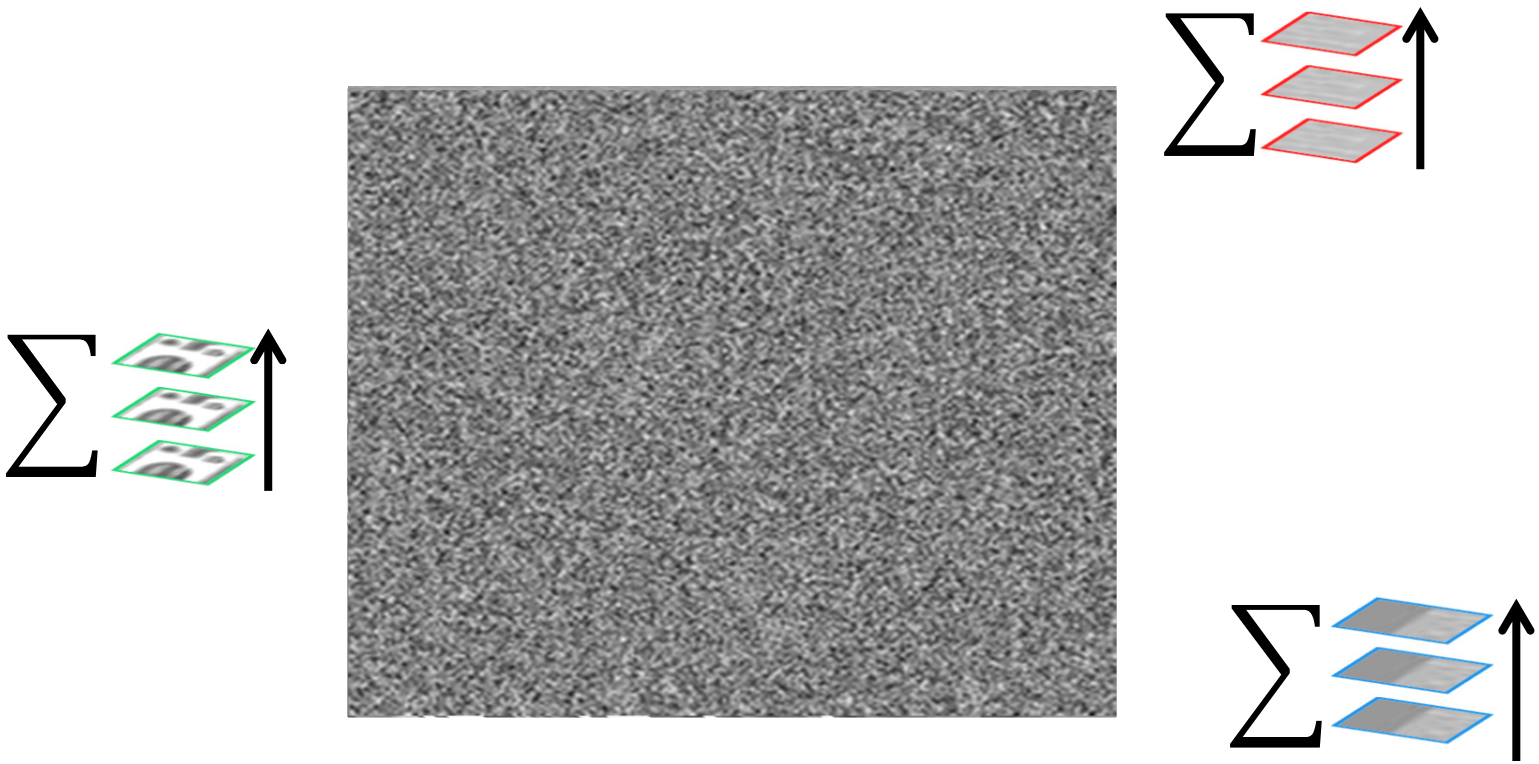


Redundancy in natural images



Glasner *et al*. (2009)

Single image “time-like” denoising



Unfortunately, patches are not exactly the same

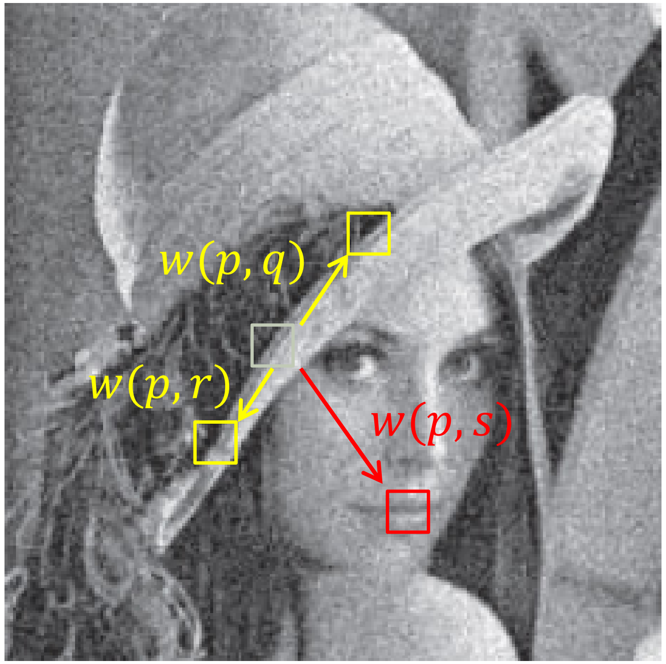
⇒ simple averaging just won’t work

Non Local Means (NLM)

Baudes *et al*. (2005)

Use a weighted average based on similarity

$$\hat{x}(i) = \frac{1}{C_i}\sum_j y(j)\, \underbrace{e^{-\frac{SSD\left(y(N_i)-y(N_j)\right)}{2\sigma^2}}}_{w(i,j)}$$



From Bilateral Filter to NLM

$$\hat{x}(i)_{BL} = \frac{1}{C_i}\sum_j y(j)\underbrace{e^{-\frac{|y(i)-y(j)|^2}{2\sigma^2}}}_{\text{intensity weight}}\underbrace{\boxed{e^{-\frac{\|i-j\|^2}{2\rho^2}}}}_{\text{spatial weight}}$$

$$\hat{x}(i)_{NLM_{1x1}} = \frac{1}{C_i}\sum_j y(j)\, e^{-\frac{|y(i)-y(j)|^2}{2\sigma^2}} \quad \boxed{\rho \rightarrow \infty}$$

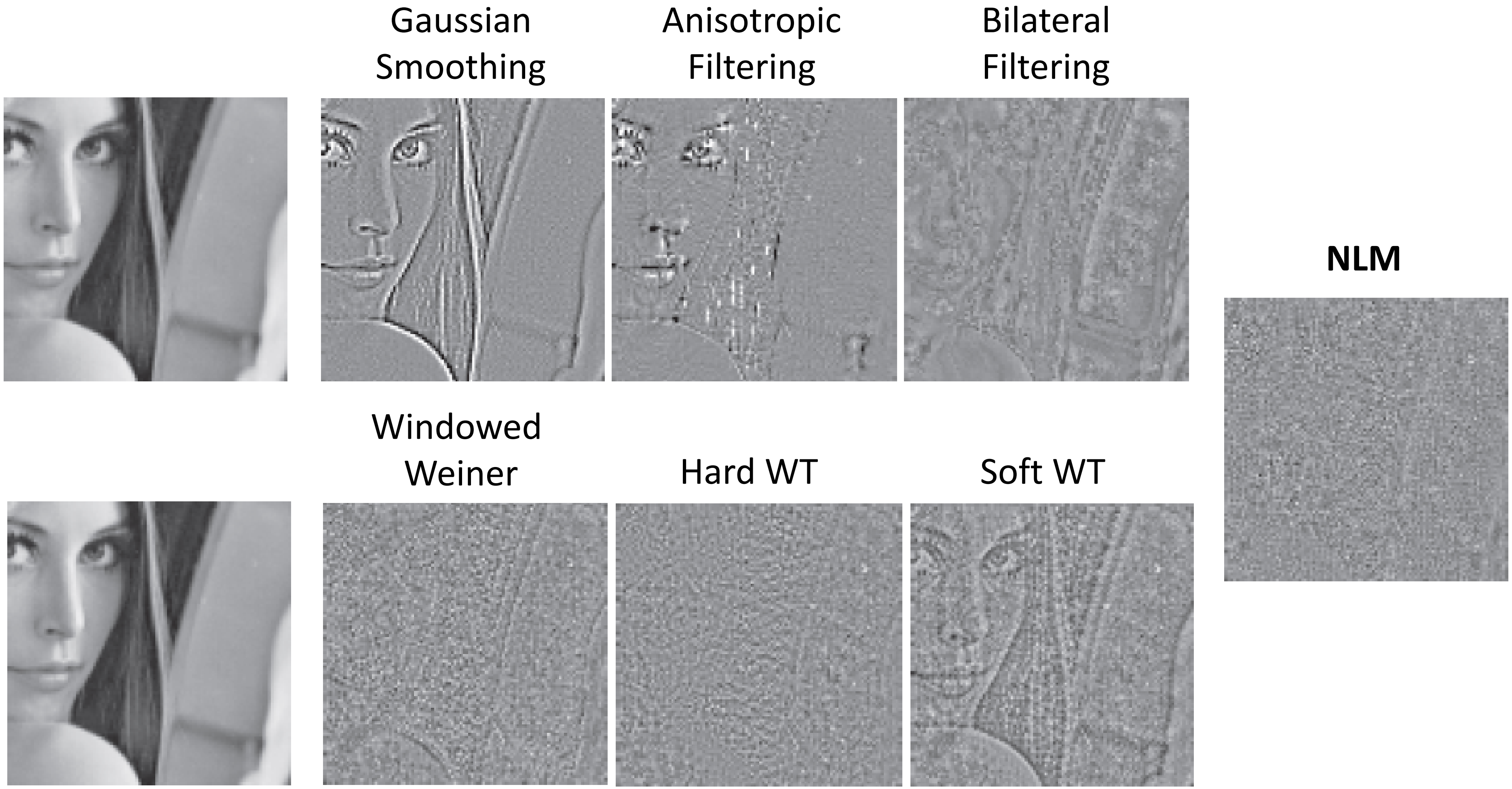
From Bilateral Filter to NLM

$$\hat{x}(i)_{NLM_{1x1}} = \frac{1}{C_i}\sum_j y(j)\, \boxed{e^{-\frac{|y(i)-y(j)|^2}{2\sigma^2}}}$$

Patch similarity

$$\hat{x}(i)_{NLM} = \frac{1}{C_i}\sum_j y(j)\, \boxed{e^{-\frac{SSD\left(\mathrm{y}(N_i)-\mathrm{y}(N_j)\right)}{2\sigma^2}}}$$

Performance Evaluation
Method Noise



Buades *et al*. (2005)

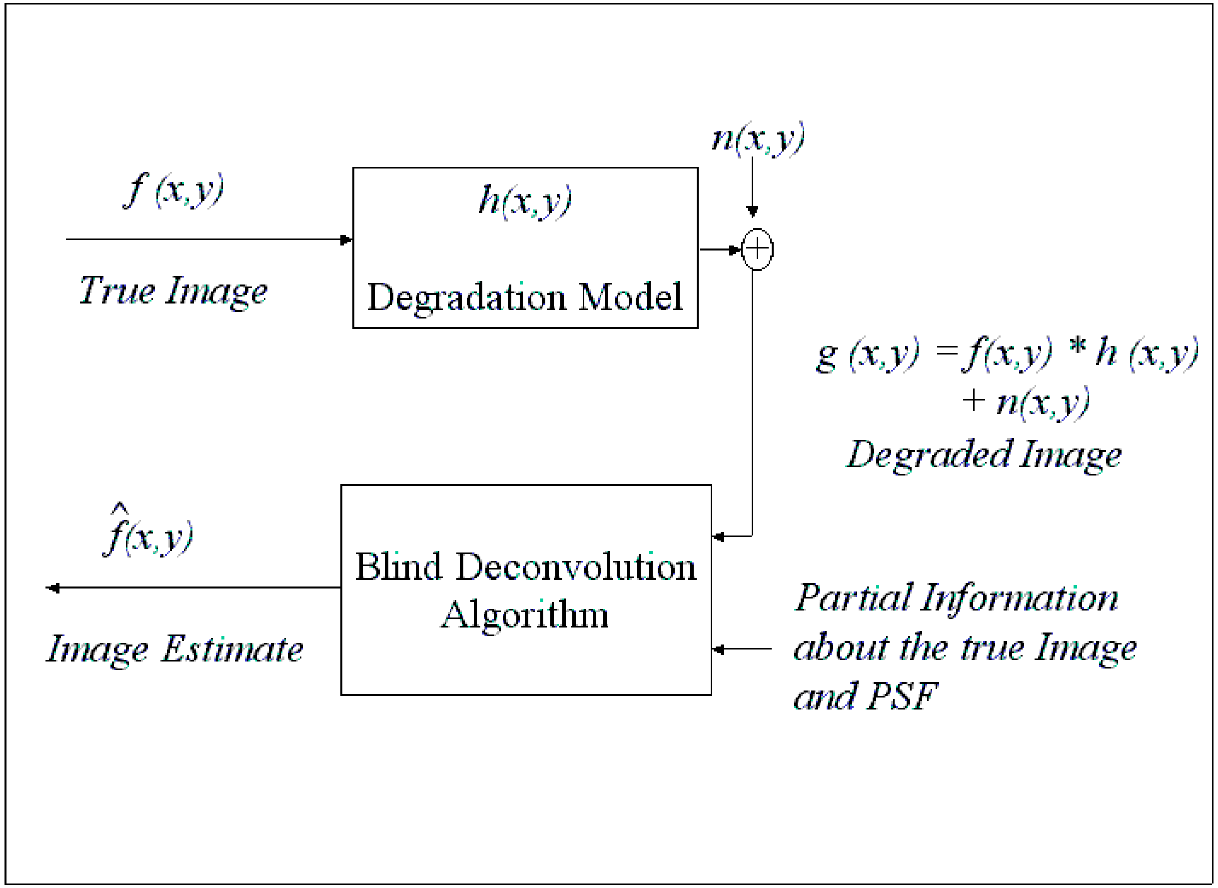
BLIND – DECONVOLUTION

Introduction

Image Restoration is an important part of many image processing applications. Its main goal is to recover the original image form a degraded observation.

The <u>existing linear image restoration algorithms</u> assume that the **PSF** (point spread function) <u>is known *a priori*</u> and attempt to invert it. However for many situations <u>PSF is not known explicitly</u> and one has to estimate the true image and the PSF simultaneously using partial or no information about the imaging system, hence the process is called ***blind image restoration***

Blind image restoration



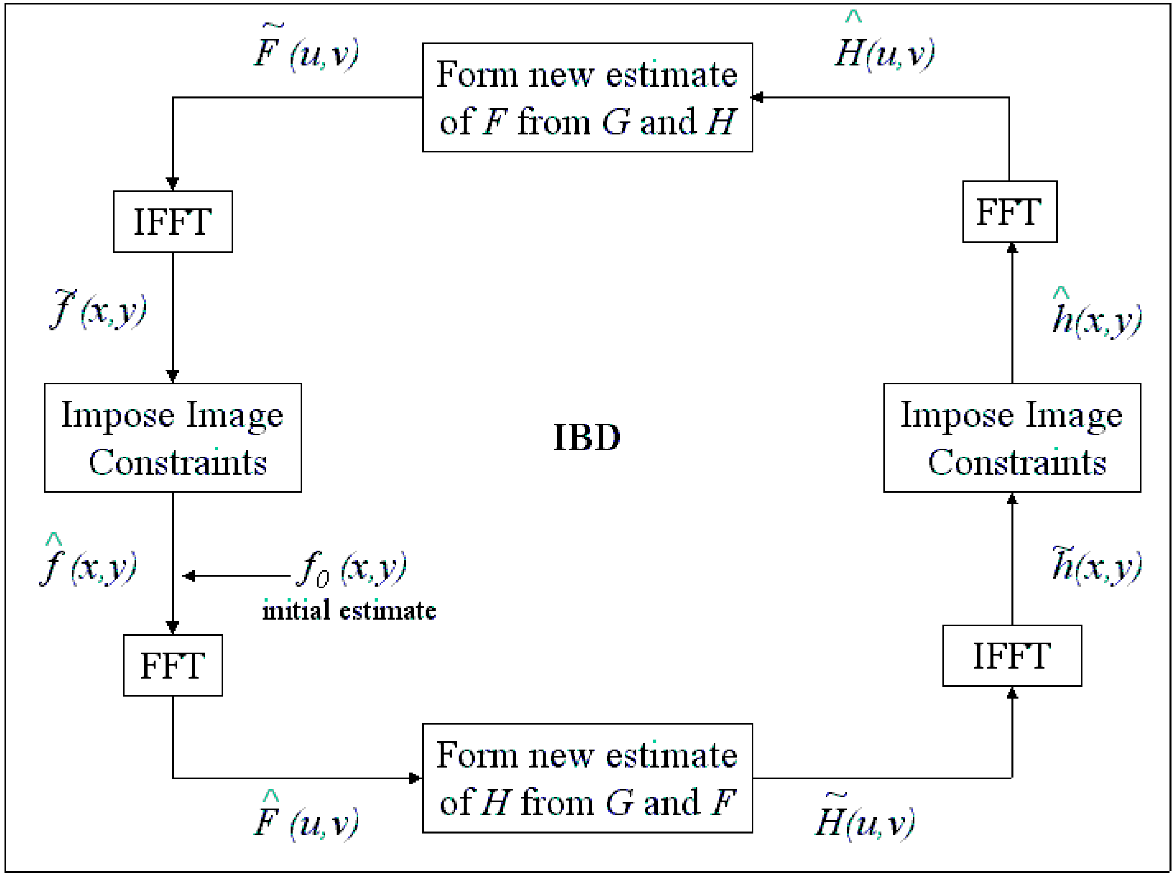
Problem Analysis

In practice there will always be some additive noise. Therefore the degradation of the image is represented by

$$g(x,y) = f(x,y) * h(x,y) + n(x,y)$$

Typically the PSF is a low pass filter and deconvolution will behave as a high pass filter, which will result in amplification of high frequency noise components. To avoid this, regularization of the problem will be required. For iterative algorithms, this can be achieved by stopping the iterations at the point where the total error (due to blurring and due to noise amplification) reaches a minimum.

IBD Algorithm



All negative Values are put to zero.

Common problems

This basic approach has two major problems to deal with:

1) Defining the inverse filter in regions where the function to be inverted has low values, is difficult.

2) We do not have any information at the spatial frequencies where $G(u,v)$ or $F(u,v)$ are zero.

To attack these problems, we change the way we implement the Fourier constraints. So at each iteration those estimates are averaged to form a new estimate. The β weight parameter is important for the convergent rate. The regions below the noise level in the convolution, are dealt with by only using the estimate

$$\hat{F}_{i+1}(u,v) = (1-\beta)\hat{F}_i(u,v) + \beta \frac{G(u,v)}{\hat{H}(u,v)}$$

Simulated Annealing Algorithm

McCallum has shown that the simulated annealing algorithm (a Monte-Carlo global minimization technique) can be applied to the blind deconvolution problem, via the minimization of the following cost function.

$$\frac{En\{f * h - g\}}{En\{g\}}$$

Where the energy of an image is defined by

$$En\left\{a\right\} = \sum_{x=-\infty}^{\infty} \left[a\left(x,y\right)\right]^2 .$$

Simulated Annealing 2

The algorithm will randomly perturb the images and will calculate the change in the cost function ΔQ. If $\Delta Q \leq 0$ then it will accept the perturbation. If $\Delta Q > 0$ then it will accept the perturbation with the probability $e^{-\Delta Q/T}$, where T is the temperature parameter. The algorithm will start with a large value of T and gradually lower it as the iteration process progresses. When T is large, the algorithm is unlikely to become trapped in local minima of Q, since the perturbations which increase Q can be accepted.

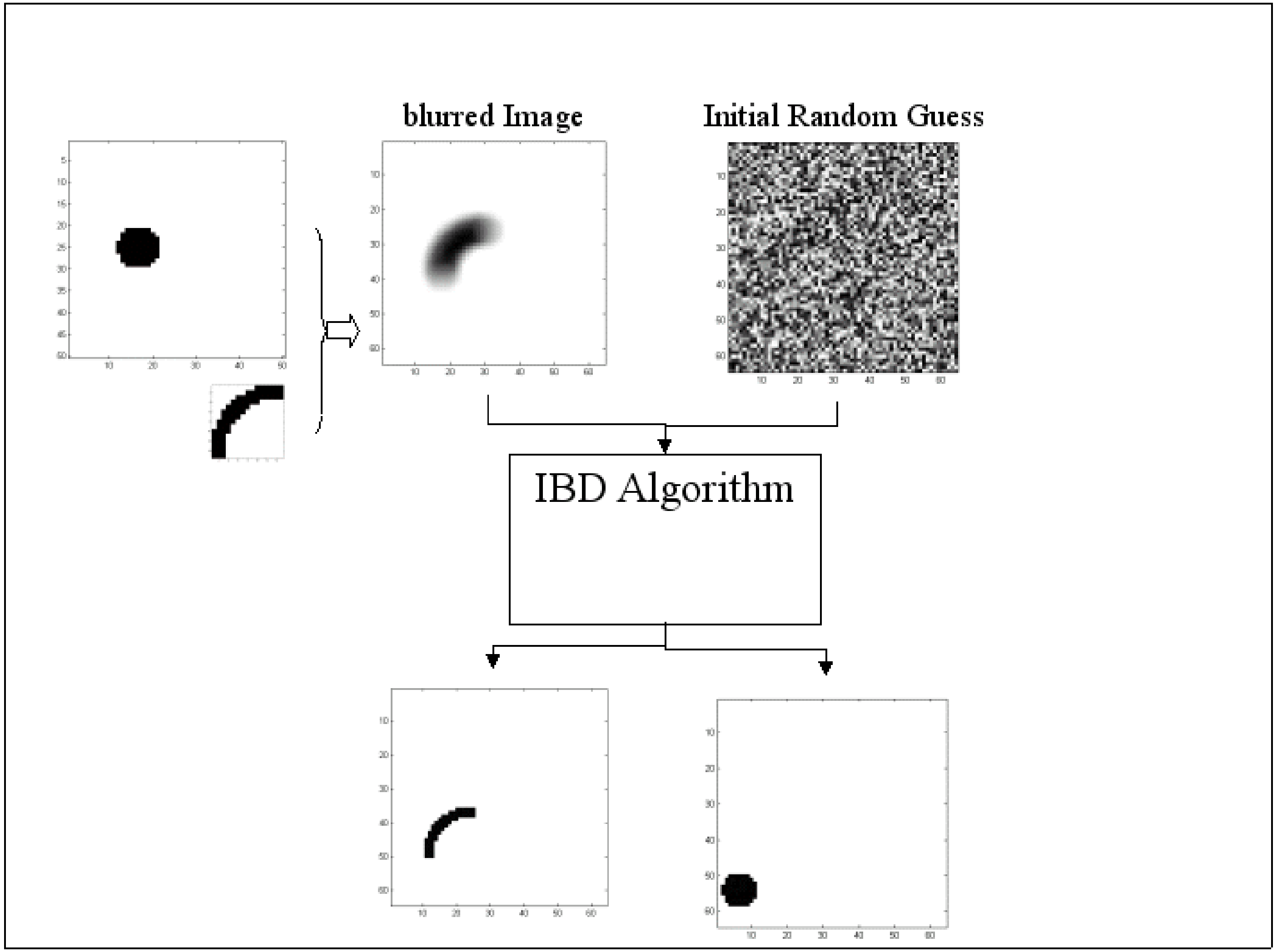
Simulated annealing optimization is similar to the annealing of metals. If the liquid metal is cooled slowly, it will reach to the absolute minimum energy state related to the complete atomic ordering of the metal. If the liquid is cooled too quickly, then the atoms will reach to a suboptimal energy state.

Simulated Annealing 3

The algorithm starts with images $f_0(x,y)$ and $h_0(x,y)$ whose pixels are pseudo-randomly distributed.

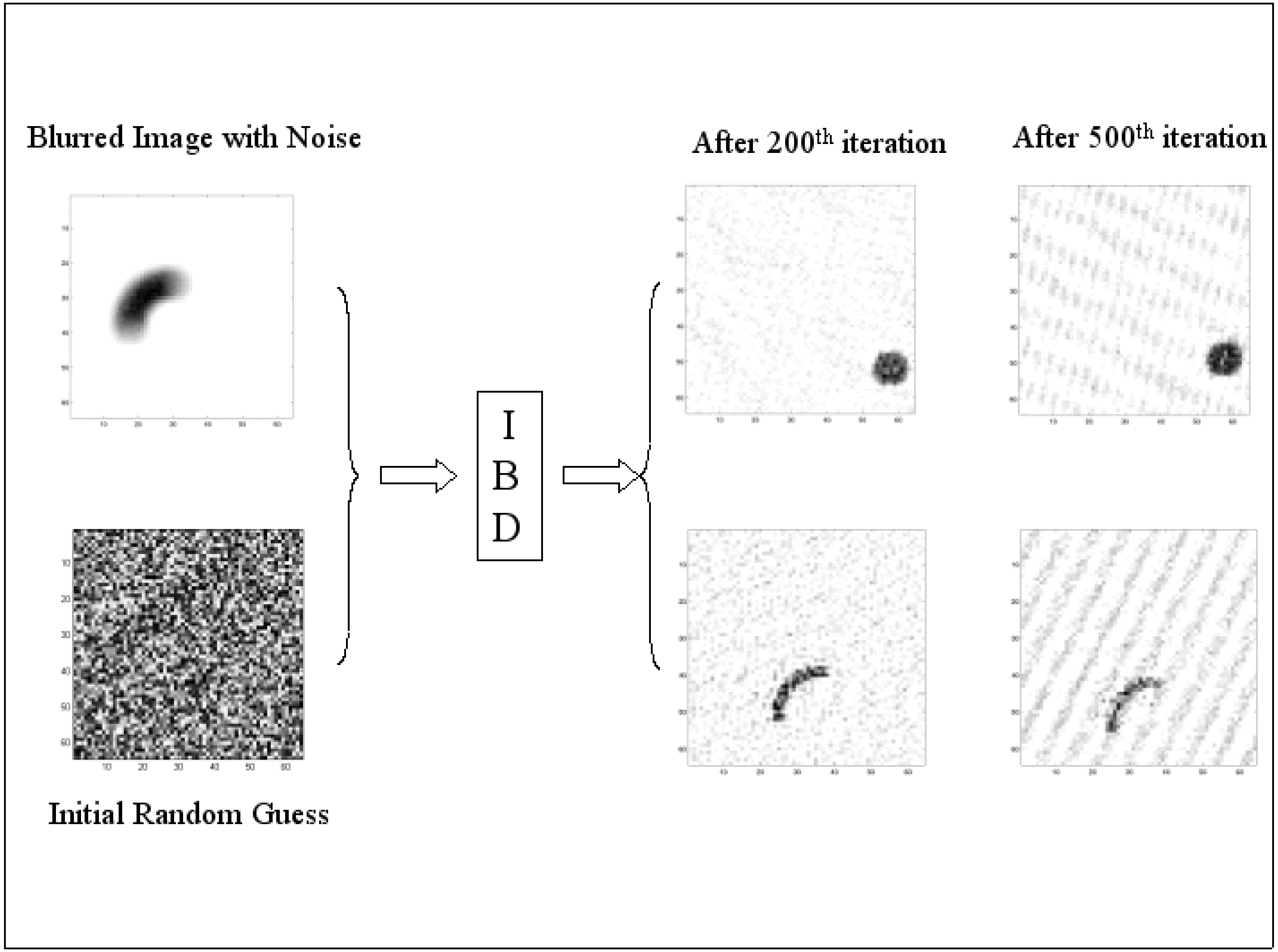
1. Calculate values of T (temperature) and α (scale of perturbation)
2. Scale $f(x,y)$ and $h(x,y)$ by factors β and $1/\beta$ respectively. So that the scaled images have equal Root Mean Square (RMS) value.
3. For each pixel of an image:
    1. $f_p(x,y) = f(x,y) + \alpha r_1$ where r_1 is a pseudo-random number, uniformly distributed in the range [-0.5, 0.5]
    2. *If* $f_p(x,y) < 0$ set $f_p(x,y) = 0$
    3. $\Delta Q = Q(f_p, h, g) - Q(f, h, g)$
    4. If $\Delta Q \le 0$ or if $\Delta Q > 0$ and $e^{-\Delta Q/T} > r_2$ *(where* r_2 is a pseudo random number, uniformly distributed in the range [0, 1]), then accept the perturbation otherwise reject.

Results



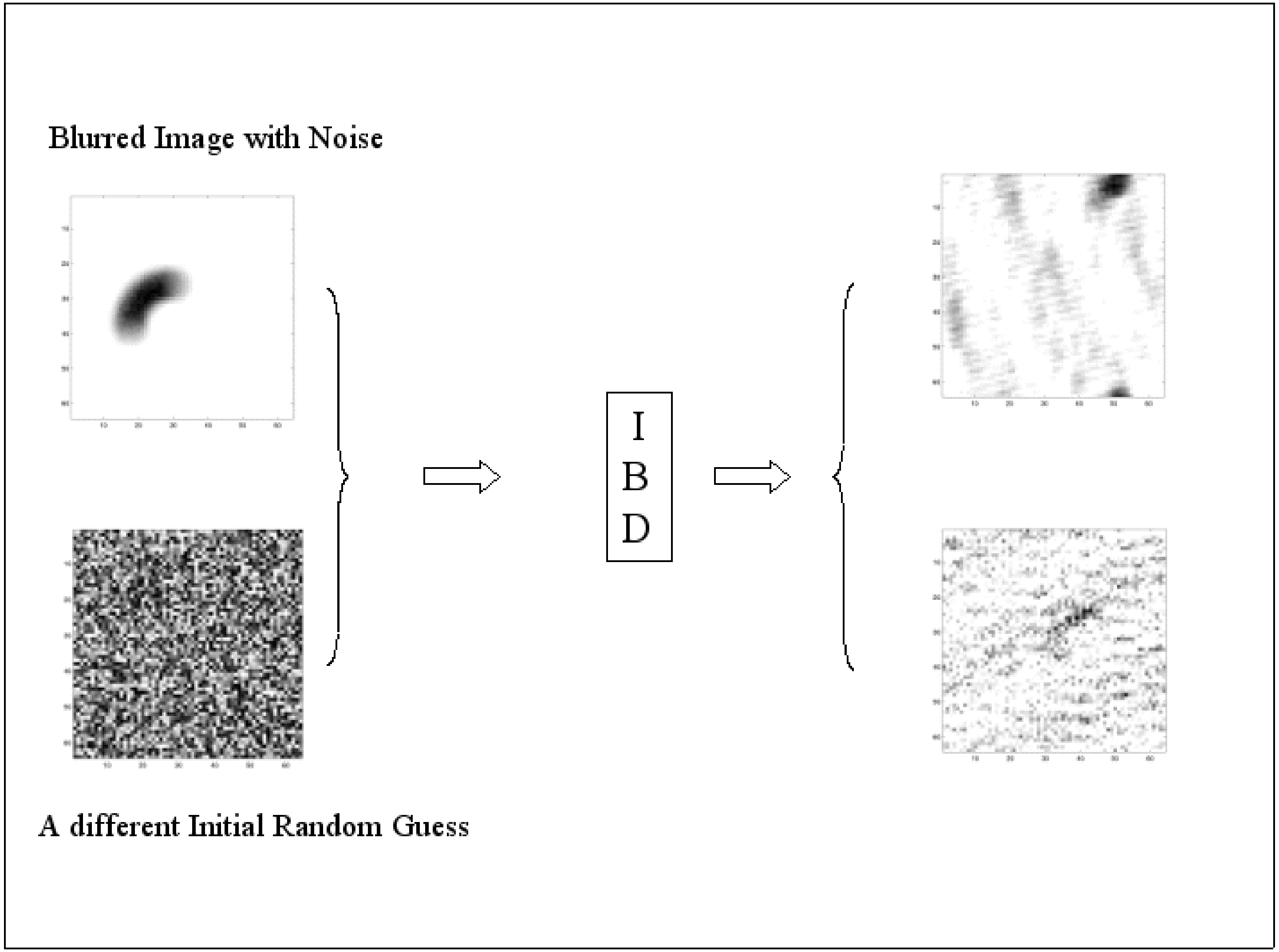
IBD result without any noise

Results



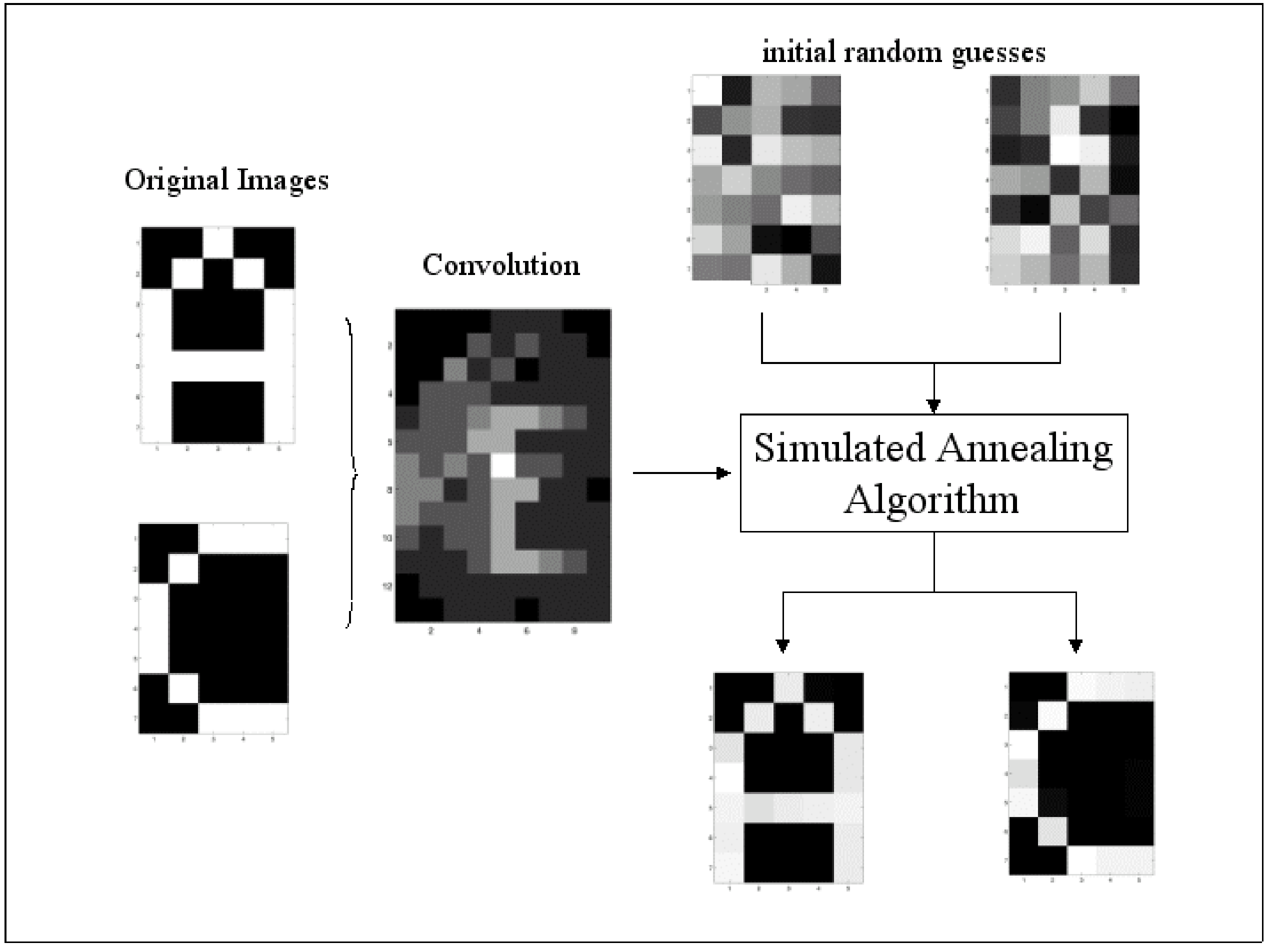
IBD result with 3% noise

Results



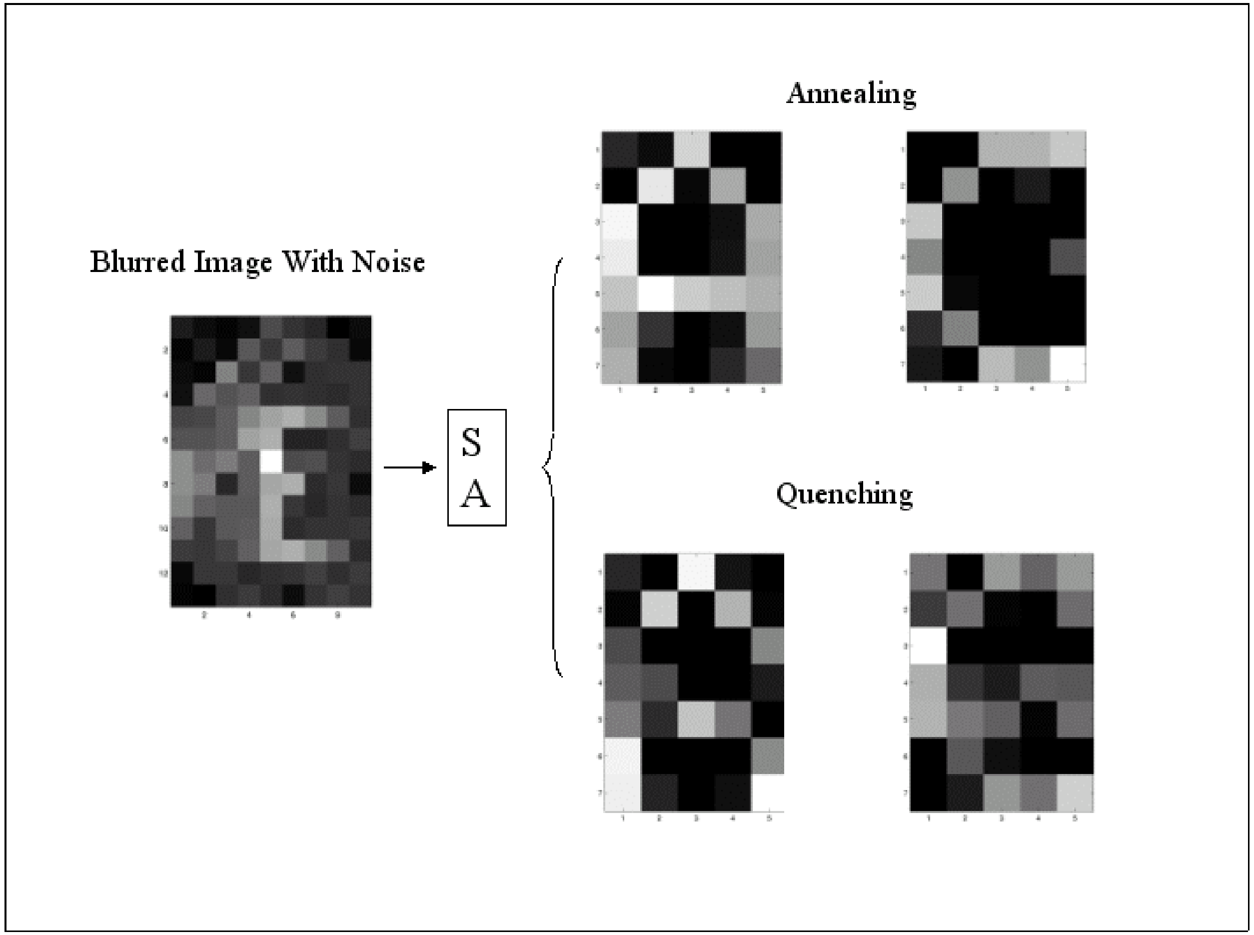
IBD result with a different initial guess

Results from SA



SA result without any noise

Results from SA



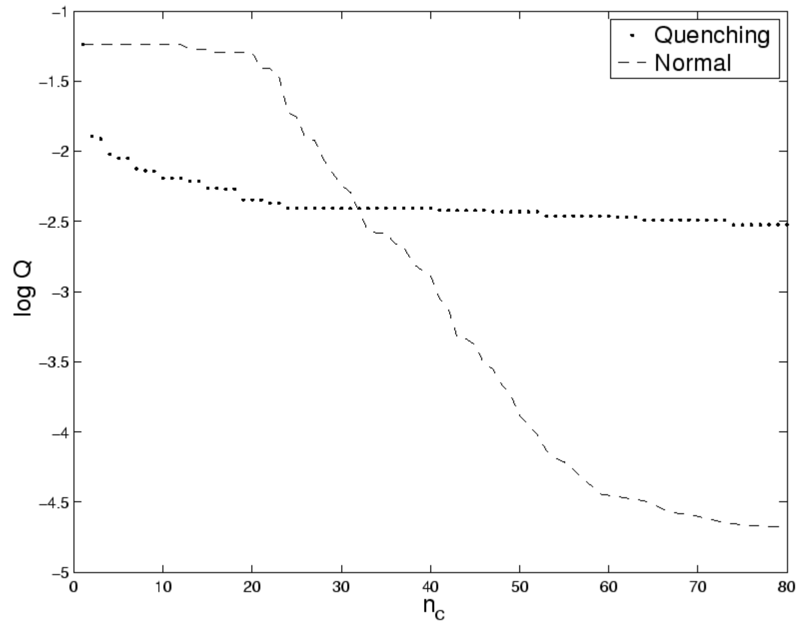
SA result with contamination level 10^{-2}

Contamination Level and Quenching

The Contamination Level is defined as:

$$CL = En\{noise(x,y)\}/En\{g\}$$

Quenching is a simulation with *T=0* and with *α=1.* Slowly reducing the temperature is crucial for simulated annealing algorithm that it may become trapped in a local minimum close to the starting point.



Noise models

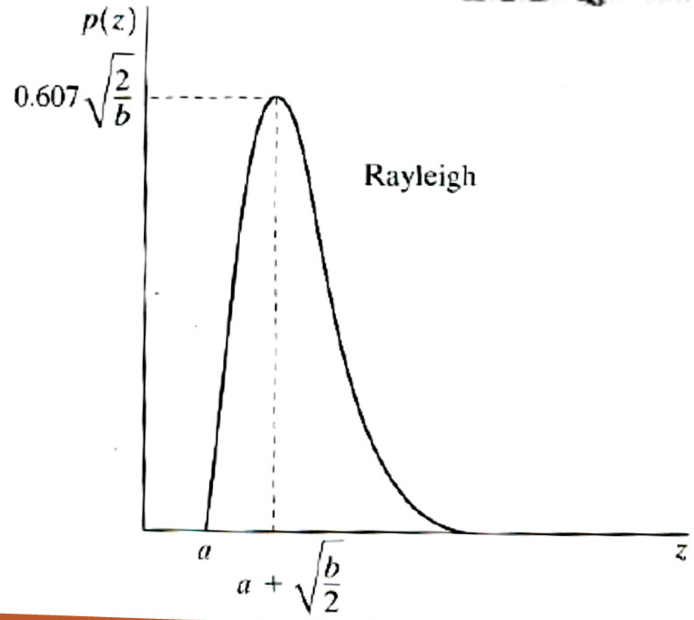
Noise Models

Rayleigh

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a. \end{cases}$$

$$\mu = a + \sqrt{\pi b/4}$$

$$\sigma^2 = \frac{b(4-\pi)}{4}.$$



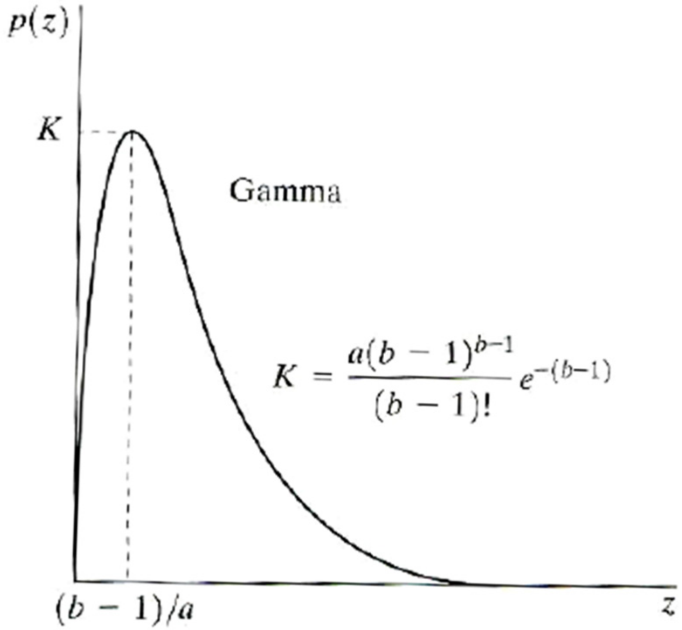
Noise Models

Erlang (Gamma)

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

$$\mu = \frac{b}{a}$$

$$\sigma^2 = \frac{b}{a^2}.$$



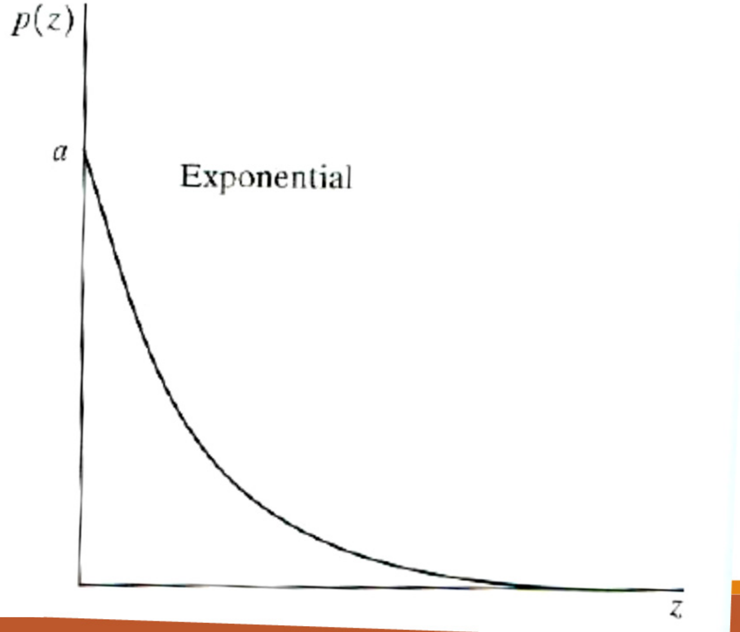
Noise Models

Exponential

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

$$\mu = \frac{1}{a}$$

$$\sigma^2 = \frac{1}{a^2}.$$



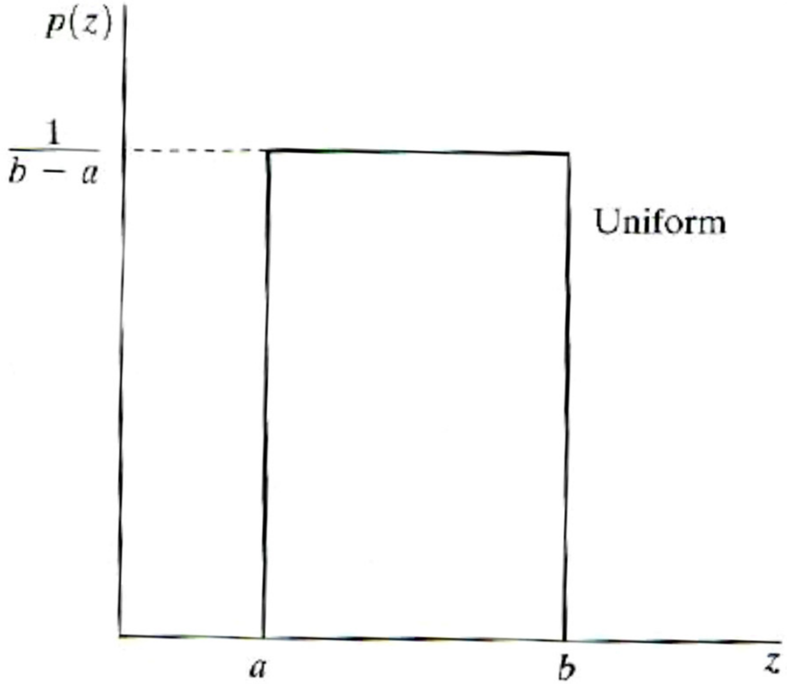
Noise Models

Uniform

$$p(z) = \begin{cases} \dfrac{1}{b-a} & \text{if } a \le z \le b \\ 0 & \text{otherwise.} \end{cases}$$

$$\mu = \frac{a+b}{2}$$

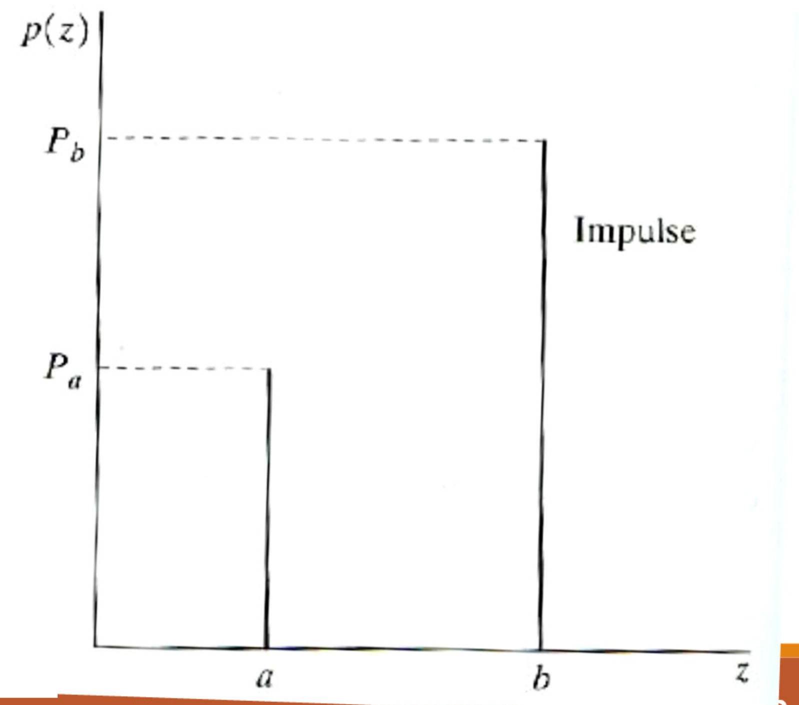
$$\sigma^2 = \frac{(b-a)^2}{12}.$$



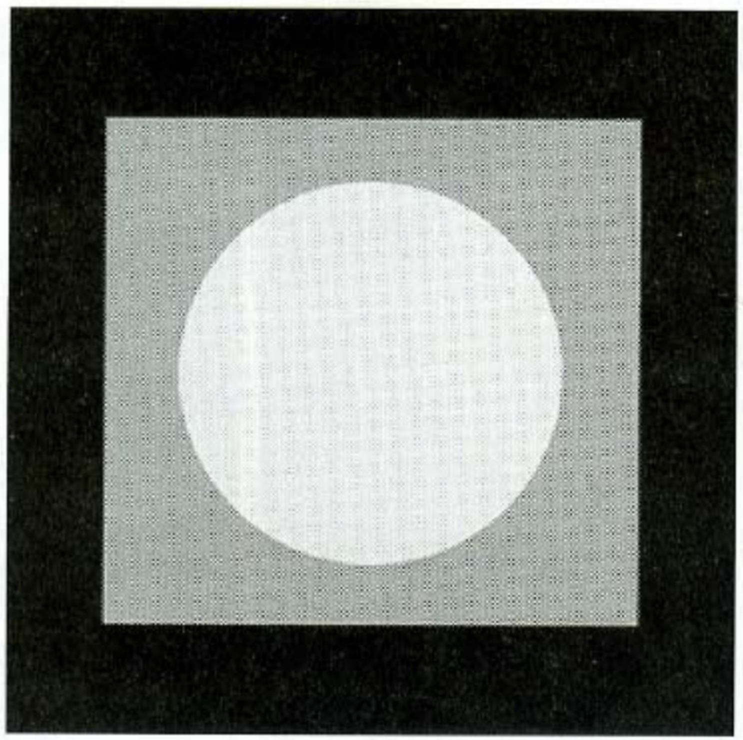
Noise Models

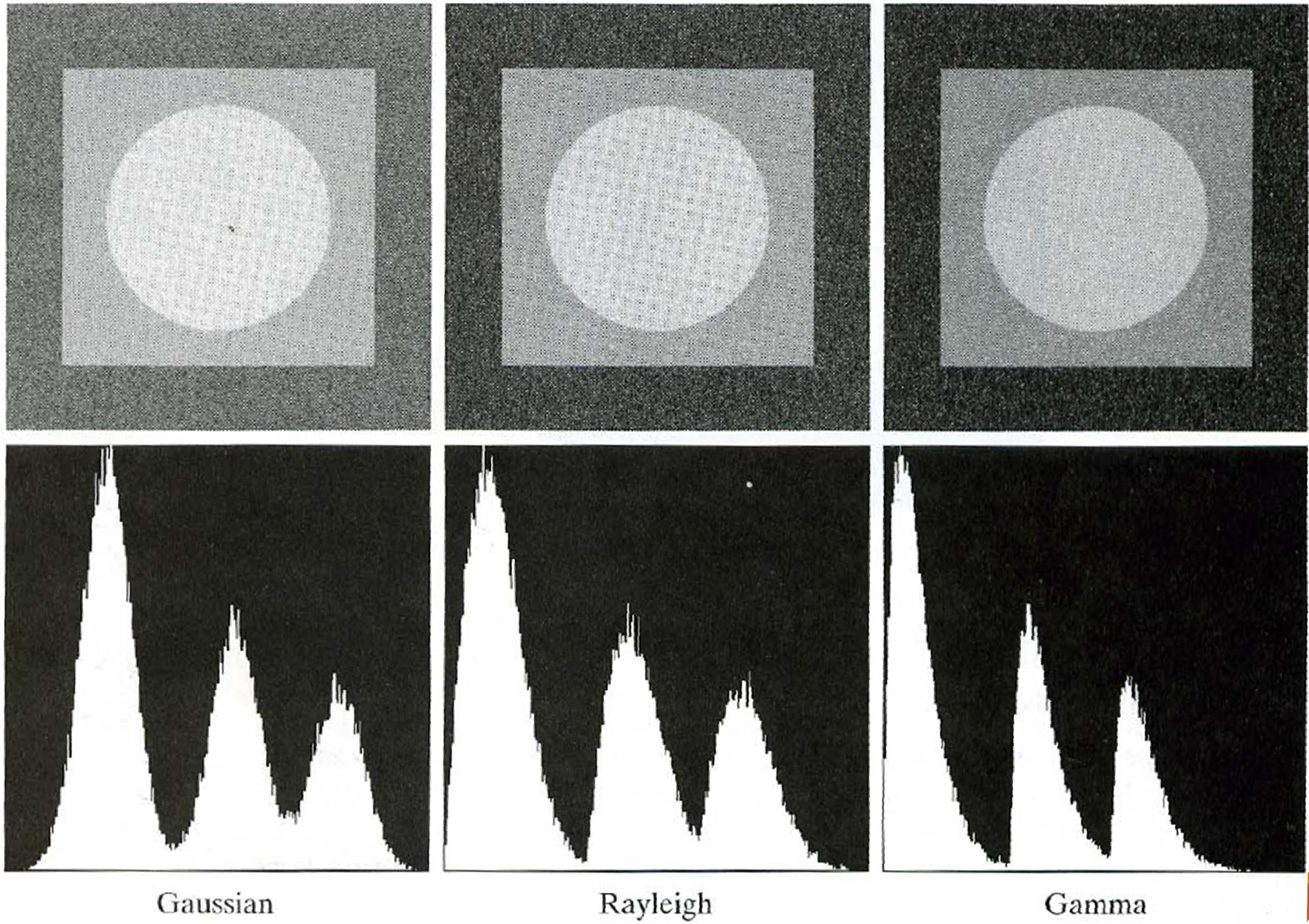
Impulsive (salt and pepper)

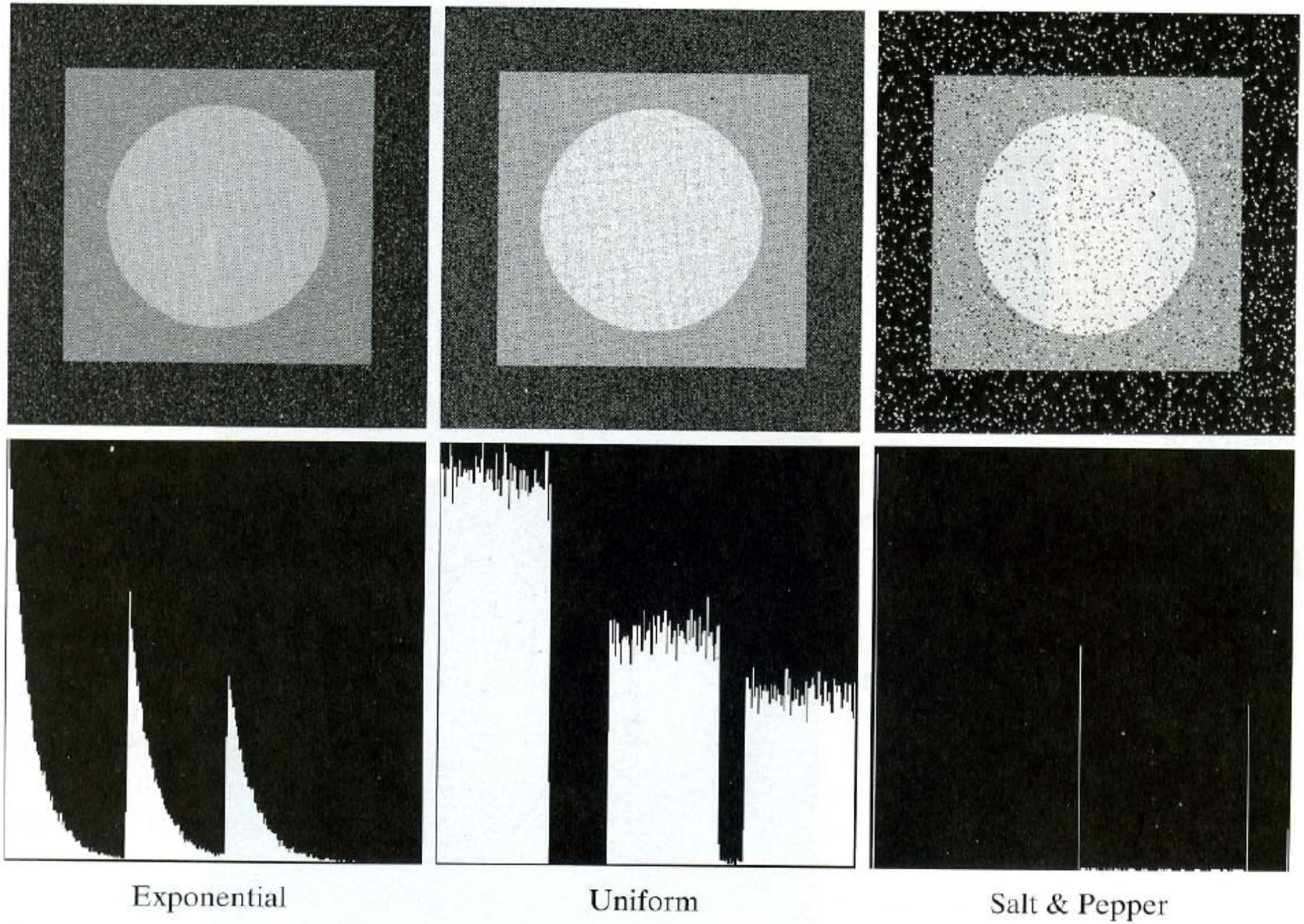
$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$



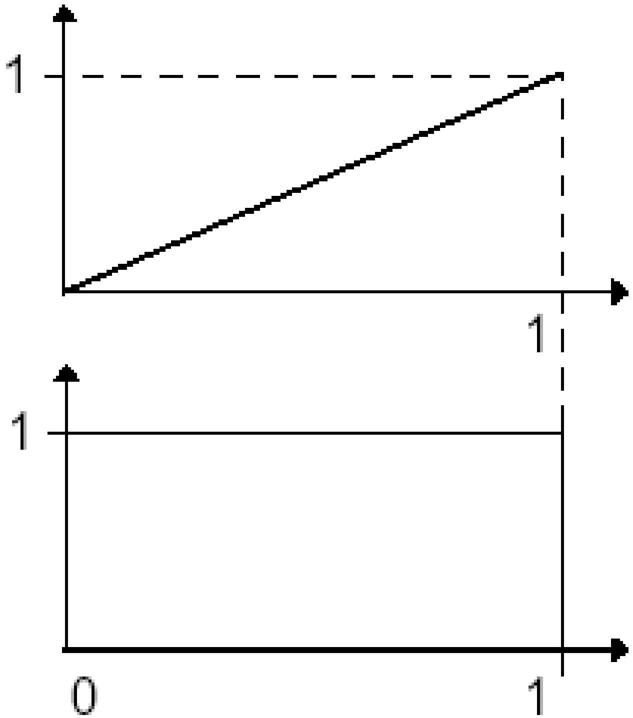
Original image




Gaussian
Rayleigh
Gamma


Exponential
Uniform
Salt & Pepper

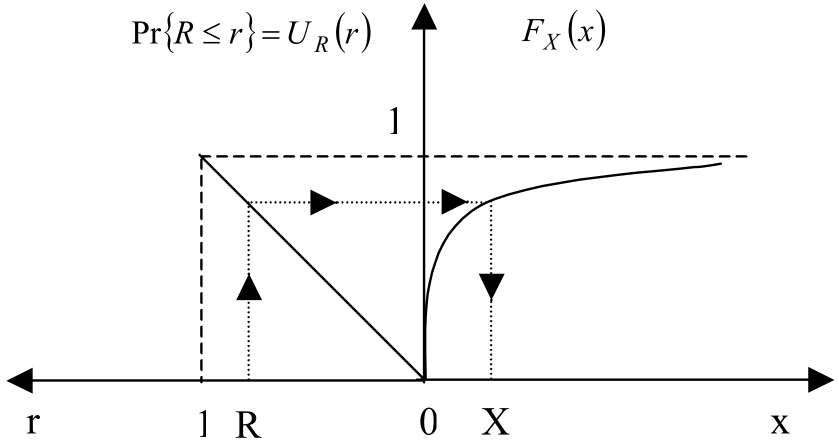
How to generate a generic noise random variable



cdf : $U_R(r) = P\{R \leq r\} = r$

pdf : $u_R(r) = \dfrac{dU_R(r)}{dr} = 1$

How to generate a generic noise random variable



Sample R from U_R(r) and find X:

$$X = F_X^{-1}(R)$$

$F_X(R)$ Is the desired cdf.

Question: which distribution does X obey?

$$P\{X \le x\} = P\{F_X^{-1}(R) \le x\}$$

Application of the operator Fx to the argument of P above yields

$$P\{X \le x\} = P\{R \le F_X(x)\} = F_X(x)$$