

Basic Image Processing Algorithms

PPKE-ITK, 2016

Lecture 5.

Image Recovery

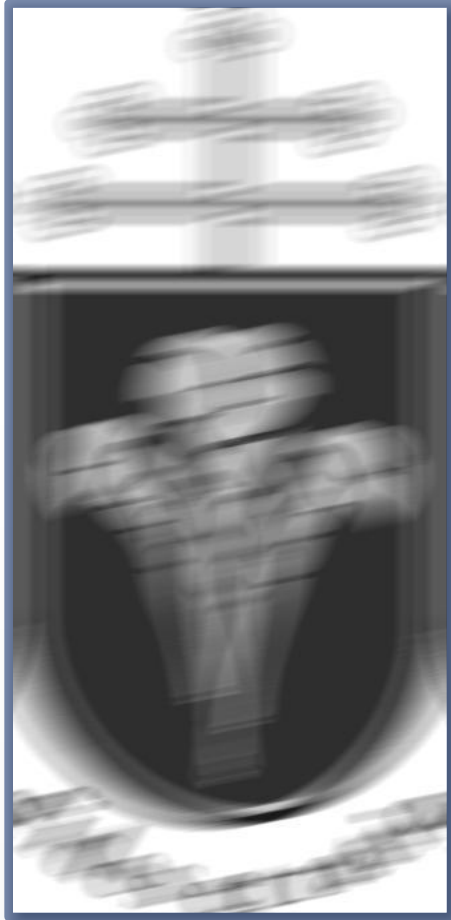
Image Recovery

What is Image Recovery?

Recovery vs Enhancement

- ◎ (Recap) **Image enhancement** is the manipulation or transformation of the image to **improve the visual appearance** or to **help further automatic processing steps**.
 - We don't add new information to the image, just make it more visible (e.g. increasing contrast) or highlight a part of it (e.g. dynamic range slicing).
- ◎ **Recovery**: the **modeling** and **removal** of the degradation the image is subjected to, based on some **optimality criteria**.
 - In case of recovery there is a ***degradation we want to remove***, lost ***information we want to recover*** (e.g. make blurred text readable again). It is done by modeling the degradation and making assumptions about the degradation and the original image.

Image Recovery Examples



Blurred Image



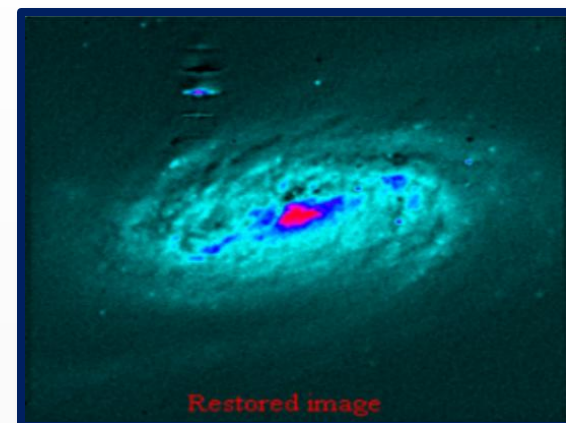
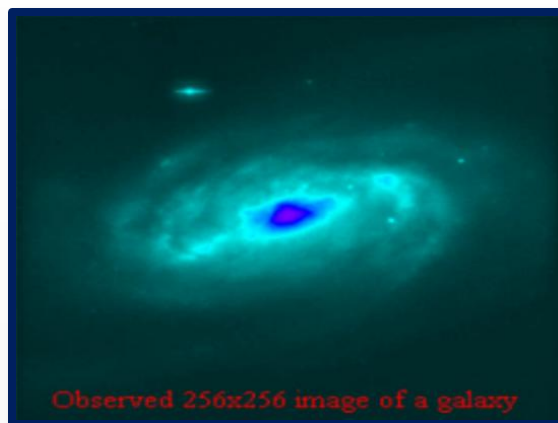
Restored Image



Original Image

Image Recovery Examples

- Images from the Hubble Space Telescope, taken with a defective mirror.



Source of the images: Fundamentals of Digital Image and Video Processing lectures by Aggelos K. Katsaggelos

Image Recovery Examples

- ◎ Blind restoration of image corrupted by motion blur:



Original Image with Motion Blur



Restored Image

Zhaofu Chen; Derin Babacan, S.; Molina, R.; **Katsaggelos**, AK., "Variational Bayesian Methods For Multimedia Problems," *Multimedia, IEEE Transactions on* , vol.16, no.4, pp.1000,1017, June 2014

Image Recovery Examples

◎ Super-resolution:

- The process of combining multiple low resolution images to form a high resolution image.



Single, non-enhanced
frame

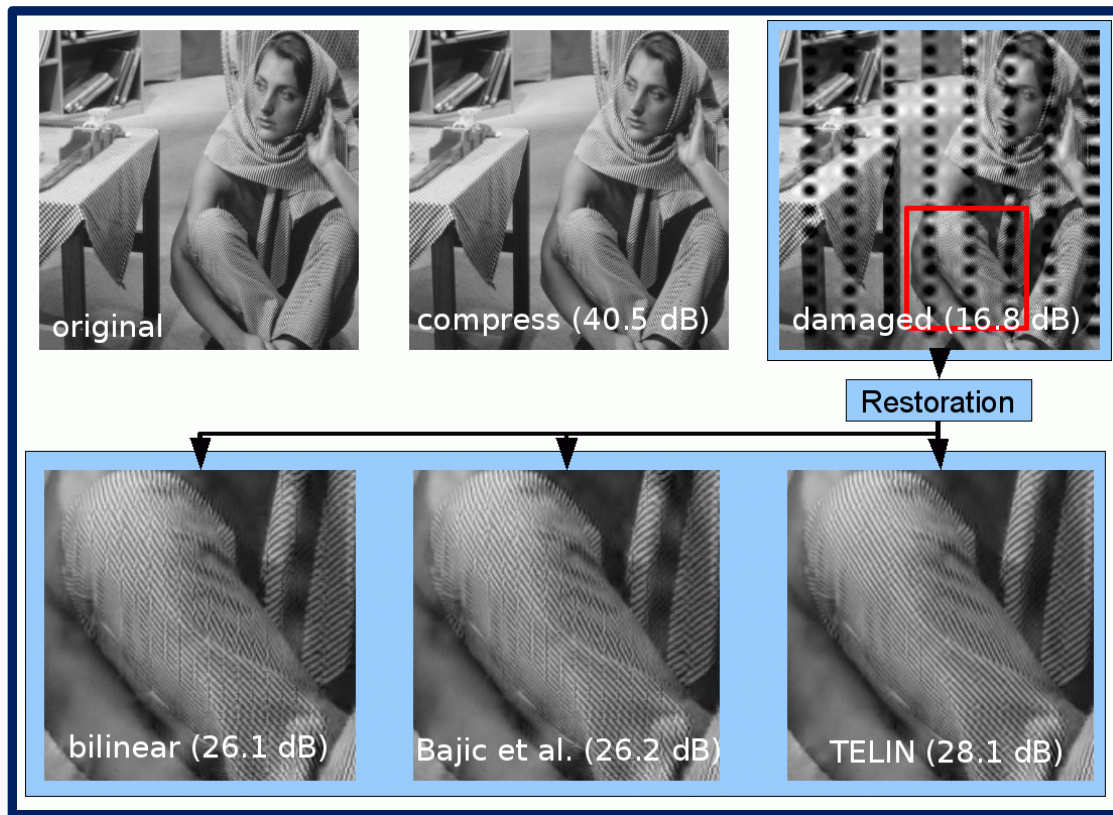
Multiple Frames of
the same scene

Reconstructed Frame

Source of the Image: <http://www.motiondsp.com/products/ikena/super-resolution>

Image Recovery Examples

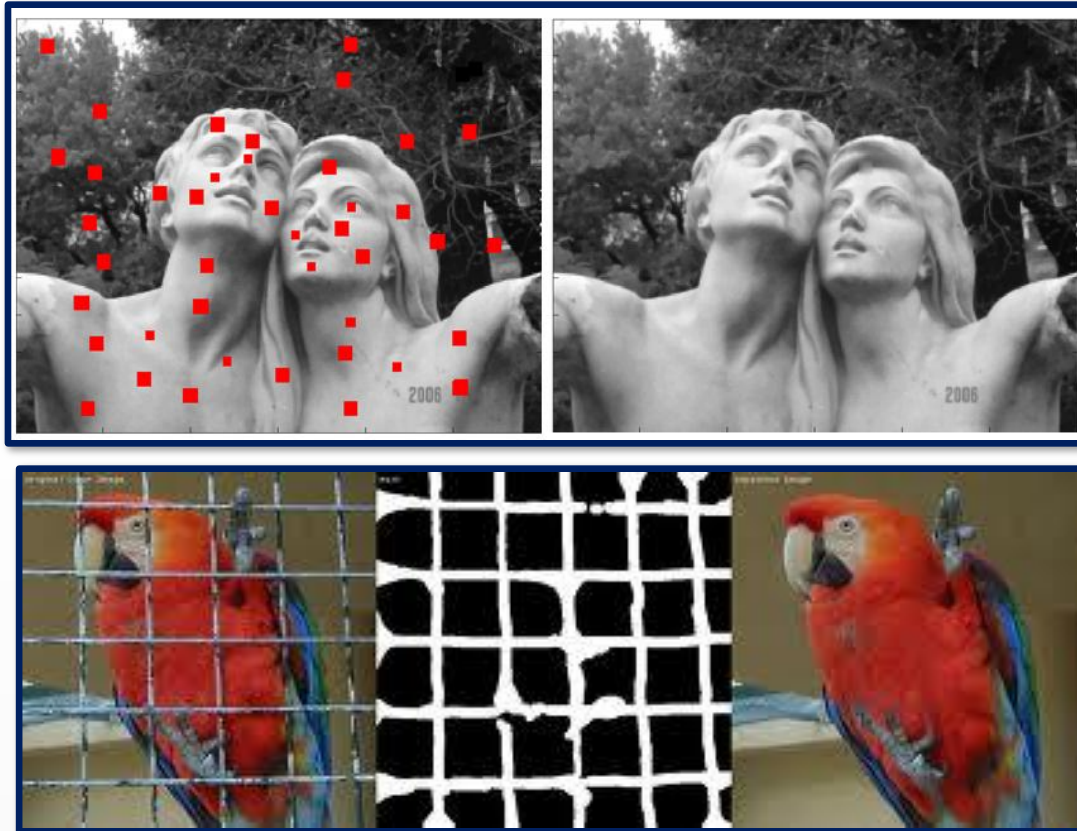
◎ Error Concealment:



J. Rombaut, A. Pizurica, and W. Philips, "Locally adaptive passive error concealment for wavelet coded images," IEEE Signal Processing Letters, vol. 15, pp. 178-181, 2008.

Image Recovery Examples

◎ Inpainting



<http://www.mathworks.com/company/newsletters/articles/applying-modern-pde-techniques-to-digital-image-restoration.html>

http://nbviewer.ipython.org/github/chintak/inpainting-demo/blob/master/Hello_ShopSense.ipynb

Image Recovery Examples

◉ Deblocking:

- removal of blocking artifacts introduced by compression



S. Alireza Golestaneh, D. M. Chandler, "An Algorithm for JPEG Artifact Reduction via Local Edge Regeneration" *Journal of Electronic Imaging (JEI)*, Jan 2014

Sources of Degradation and Forms of Recovery

Sources of Degradation

1. Motion
2. Atmospheric turbulence
3. Out-of-focus lens
4. Finite resolution of the sensors
5. Limitations of the acquisition system
6. Transmission error
7. Quantization error
8. Noise

Forms of Restoration

1. Restoration/Deconvolution
2. Removal of Compression Artifacts
3. Super-Resolution
4. Inpainting/Concealment
5. Noise smoothing

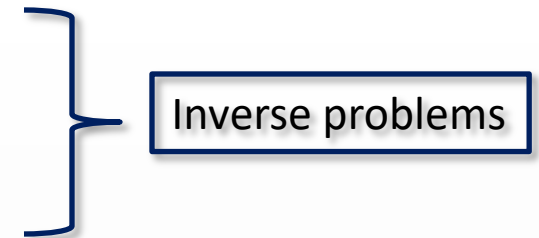
Inverse problem formulation of Recovery

- ⊙ The original image x goes through a system (H), that introduces some type of degradation resulting the observed image y :

$$x(n_1, n_2) \rightarrow H \rightarrow y(n_1, n_2)$$

- ⊙ The objective is to reconstruct x based on...

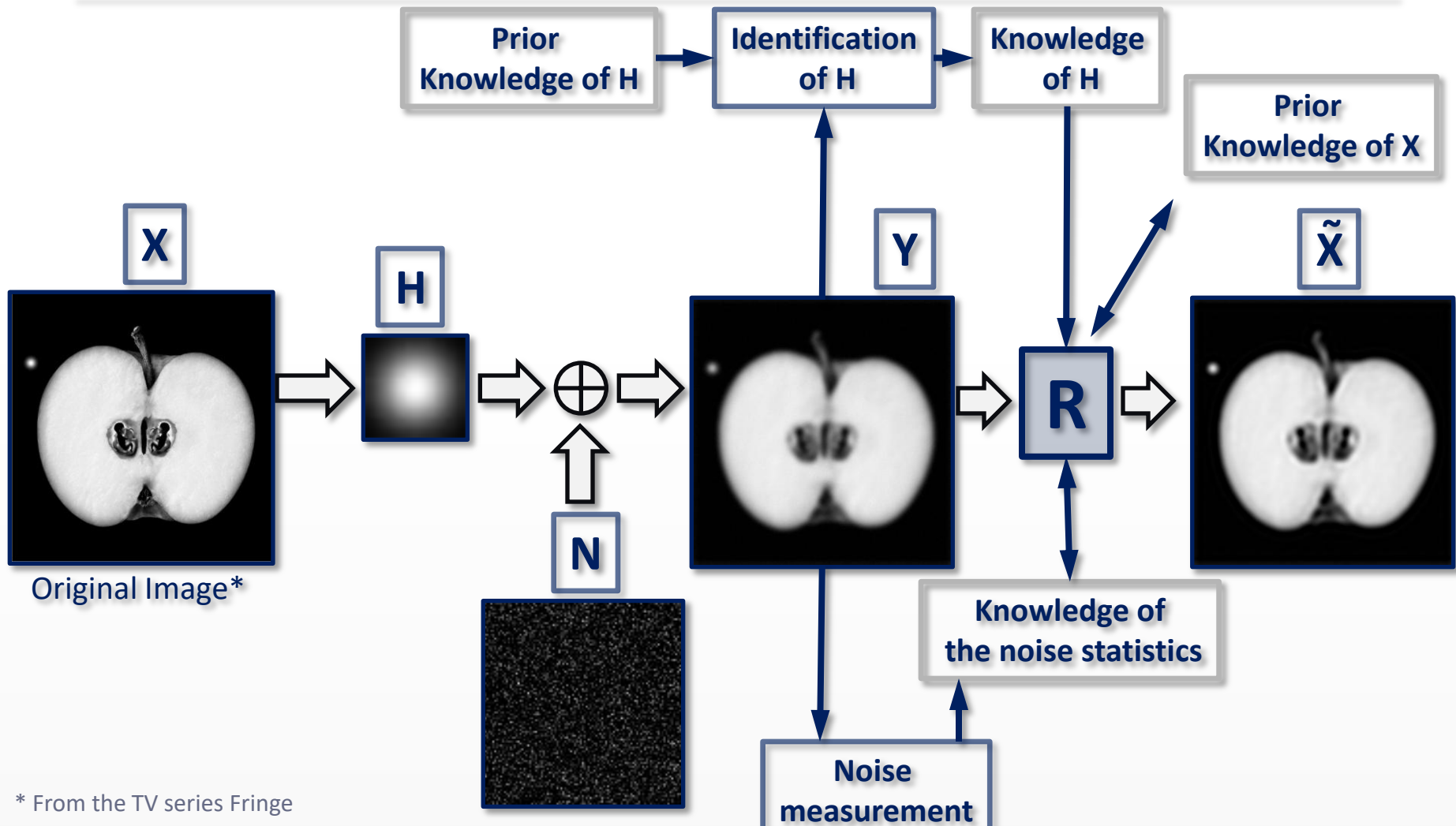
- y and H $\rightarrow$ recovery
- y $\rightarrow$ blind recovery
- y and partially H $\rightarrow$ semi-blind recovery



- ⊙ If we know x and

- y $\rightarrow$ system identification
- H $\rightarrow$ system implementation

Degradation and Restoration



Degradation Model

- ⊙ The model of degradation for **restoration** problems:

$$y(n_1, n_2) = H[x(n_1, n_2)] + n(n_1, n_2)$$

- ⊙ If an LSI degradation system is assumed, with signal independent additive noise:

$$y(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2) + n(n_1, n_2)$$

- ⊙ The restoration problem in this case is called **deconvolution**.

Convolution in matrix-vector form

- 1D convolution can be represented in a matrix-vector form:

$$y(n) = x(n) * h(n) = \sum_k x(k)h(n - k), \text{ where } \begin{array}{l} x : 1 \times N \\ h : 1 \times L \\ y : 1 \times (N + L - 1) \end{array}$$

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N + L - 2) \end{bmatrix} = \begin{bmatrix} h(0) & 0 & \dots & \dots & 0 \\ h(1) & h(0) & 0 & \dots & 0 \\ h(2) & h(1) & h(0) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h(L-1) & \vdots & \vdots & \vdots & h(0) \\ 0 & h(L-1) & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & h(L-1) \end{bmatrix} \cdot \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

$$\mathbf{y} = \mathbf{H} \cdot \mathbf{x}$$

Convolution in matrix-vector form

- ⊙ If we stack the observed image lexicographically into a vector, the degradation can be written in matrix-vector form:

$$y = Hx + n$$

- ⊙ If the system is LSI, then H is a **block circulant matrix**.

For block circulant matrices it can be shown, that we can take it to the Fourier domain elementwise, so the following equation holds:

$$Y(\omega_1, \omega_2) = H(\omega_1, \omega_2) \cdot X(\omega_1, \omega_2) + N(\omega_1, \omega_2), \text{ where } \begin{cases} \omega_1 = 0, \dots, M-1 \\ \omega_2 = 0, \dots, M-1 \end{cases}$$

Image Recovery

Deconvolution Algorithms

Inverse Filter

- ◉ Simplest deconvolution filter, developed for LSI systems.
- ◉ Can be easily implemented in the frequency domain as the inverse of the degradation filter.
- ◉ Main limitations and drawbacks:
 - Strong noise amplification
 - The degradation system has to be known a priori.
- ◉ The degradation equation: $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$
 - The objective is to find $\mathbf{x}$ that minimizes the following goal function:

$$\underset{\mathbf{x}}{\operatorname{argmin}}(J(\mathbf{x})) = \underset{\mathbf{x}}{\operatorname{argmin}}(\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2)$$
$$\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}} = 0 \Rightarrow \frac{\partial}{\partial \mathbf{x}} (\mathbf{y}^T \mathbf{y} - 2\mathbf{x}^T \mathbf{H}^T \mathbf{y} + \mathbf{x}^T \mathbf{H}^T \mathbf{H} \mathbf{x}) = -2\mathbf{H}^T \mathbf{y} + 2\mathbf{H}^T \mathbf{H} \mathbf{x} = 0$$

$$\mathbf{H}^T \mathbf{H} \mathbf{x} = \mathbf{H}^T \mathbf{y}$$

$$\mathbf{x} = (\mathbf{H}^T \mathbf{H})^+ \mathbf{H}^T \mathbf{y}$$

Generalized Inverse

Inverse Filter

- ◉ If the degradation system is LSI, then $\mathbf{H}$ is a block circulant:
 - We can take the calculation to the frequency domain:

$$x = (H^T H)^+ H^T y \xrightarrow{\text{DFT}} X(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)Y(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2}$$

$$X(\omega_1, \omega_2) = \begin{cases} \frac{H^*(\omega_1, \omega_2)Y(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2} & \text{if } |H(\omega_1, \omega_2)| \neq 0 \\ 0 & \text{if } |H(\omega_1, \omega_2)| = 0 \end{cases}$$

- The drawback of this method is the strong amplification of noise:

$$\frac{H^*(\omega_1, \omega_2)Y(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2} = \frac{Y(\omega_1, \omega_2)}{H(\omega_1, \omega_2)} = \frac{H(\omega_1, \omega_2)X(\omega_1, \omega_2)}{H(\omega_1, \omega_2)} + \frac{N(\omega_1, \omega_2)}{H(\omega_1, \omega_2)} = X(\omega_1, \omega_2) + \boxed{\frac{N(\omega_1, \omega_2)}{H(\omega_1, \omega_2)}}$$

- To reduce this effect a threshold can be used on $\mathbf{H}$:

$$X(\omega_1, \omega_2) = \begin{cases} \frac{H^*(\omega_1, \omega_2)Y(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2} & \text{if } |H(\omega_1, \omega_2)| \geq T \\ 0 & \text{if } |H(\omega_1, \omega_2)| < T \end{cases}$$

Inverse Filter



Original Image



Blurred Image

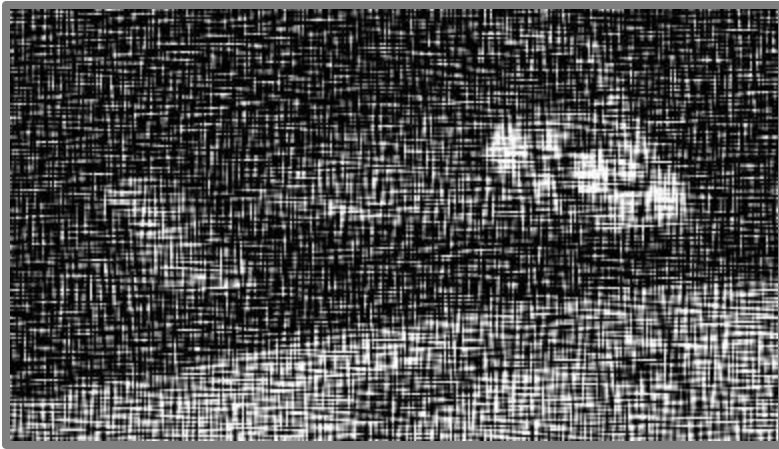


Blurred Image with Additional Noise



The Noise Component (amplified 10 times)

Inverse Filter



Reconstructed Image (with $T=0.005$)



Reconstructed Image (with $T=0.01$)

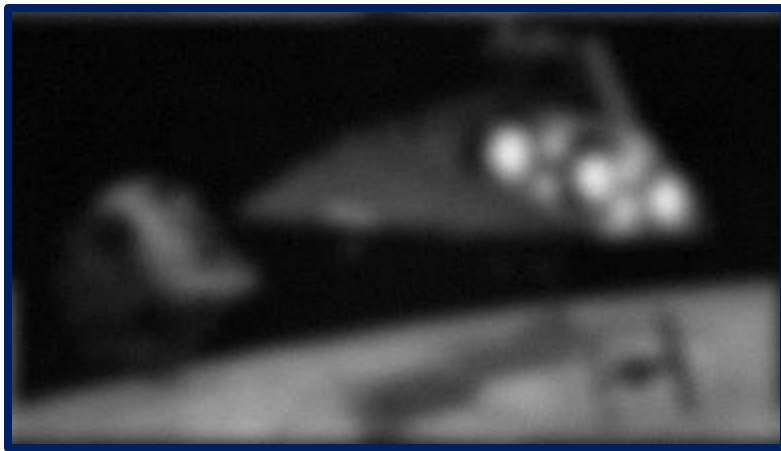


Reconstructed Image (with $T=0.02$)

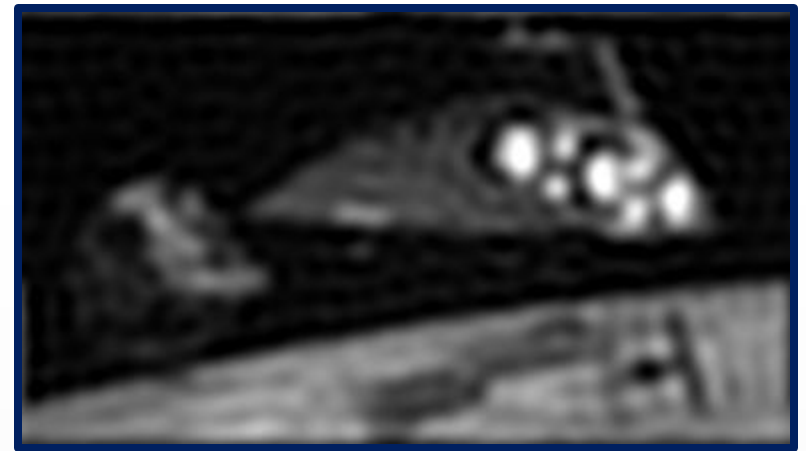


Reconstructed Image (with $T=0.1$)

Inverse Filter



Blurred Image with Additional Noise



Reconstructed Image (with $T=0.1$)

Constrained Least Square Methods

- The objective is to reduce the noise amplification effect of the inverse filter by adding extra constraints about the restored image:

- On one hand we still have the term describing the solution's fidelity to the data:

$$\operatorname{argmin}_x (J(x)) = \operatorname{argmin}_x (\|y - Hx\|^2)$$

- But we also have a second term, incorporating some *prior knowledge* about the smoothness of the original image:

$$\|Cx\|_2^2 < \varepsilon$$

- Putting the two together with the introduction of α :

$$\operatorname{argmin}_x (\|y - Hx\|_2^2 + \alpha \|Cx\|_2^2)$$

Constrained Least Square Methods

$$\operatorname{argmin}_x \left(\|y - Hx\|_2^2 + \alpha \|Cx\|_2^2 \right) \Rightarrow x = (H^T H + \alpha C^T C)^+ H^T y$$

- ⊙ C is a high pass filter
 - Intuitively this means that we want to keep under control the amount of energy contained in the high frequencies on the restored image.
- ⊙ α is the regularization parameter
- ⊙ In the frequency domain (for H and C block circulant) we have the following formula:

$$X(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2 + \alpha |C(\omega_1, \omega_2)|^2} Y(\omega_1, \omega_2)$$

- if α is 0, we get back the simple Least Square method (Inverse Filter)

Constrained Least Square Methods

◉ (Optional) Different types of regularization:

- CLS:

$$\tilde{x}(\alpha)_{CLS} = \underset{x}{\operatorname{argmin}} \left(\|y - Hx\|_2^2 + \alpha \|Cx\|_2^2 \right)$$

- Maximum Entropy Regularization:

$$\tilde{x}(\alpha)_{ME} = \underset{x}{\operatorname{argmin}} \left(\|y - Hx\|_2^2 + \alpha \sum_{i=1}^N x_i \log(x_i) \right)$$

- Total Variation Regularization:

$$\tilde{x}(\alpha)_{TV} = \underset{x}{\operatorname{argmin}} \left(\|y - Hx\|_2^2 + \alpha \sum_{i=1}^N |[\Delta x]_i| \right)$$

- l_p – norms:

$$J(z) = \|z\|_p^p = \sum_{i=1}^N |z_i|^p, \quad 1 \leq p \leq 2$$

Constrained Least Square Methods



Original Image



Blurred Image with Additional Noise



Reconstructed Image (CLS with $\alpha=0.1$)



Reconstructed Image (CLS with $\alpha=0.5$)

CLS vs. LS



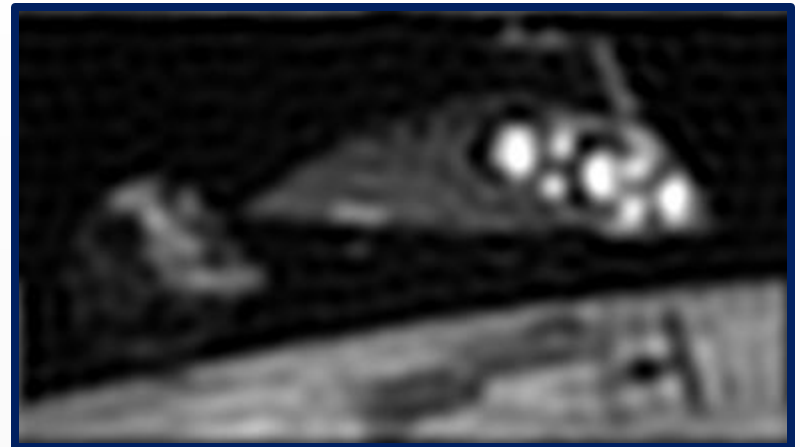
Reconstructed Image (CLS with $\alpha=0.1$)



Reconstructed Image (CLS with $\alpha=0.5$)



Reconstructed Image (LS with $T=0.02$)



Reconstructed Image (LS with $T=0.1$)

Iterative Restoration Algorithms

◎ Pros:

- ***Don't need to have the inverse*** of the degradation system explicitly
- The process can be monitored as it progresses, the ***number of iteration can be used as*** some kind of ***regularization*** (noise amplification can be controlled)
- Can be applied to ***spatially varying degradations*** and blind degradations

◎ Cons:

- Convergence of the algorithm is not always guaranteed
- Possibly takes more time

Iterative Restoration Algorithms

⊙ Using Successive Approximation Algorithm:

- Find a root for $\phi(x)$, by taking a reasonable initial point and iteratively go toward the root:

$$x_0 = 0$$

$$x_{k+1} = x_k + \beta\Phi(x_k)$$

⊙ In restoration:

$$\Phi(x) = y - Hx$$

$$x_{k+1} = x_k + \beta(y - Hx_k) = \beta y + (I - \beta H)x_k$$

- If the degradation system is assumed to be LSI (H is block circulant), then in the frequency domain we'll have the following:

$$X_{k+1}(\omega_1, \omega_2) = \beta Y(\omega_1, \omega_2) + (1 - \beta H(\omega_1, \omega_2))X_k(\omega_1, \omega_2)$$

Iterative Restoration Algorithms

- What do we know about the convergence?

$$X_{k+1}(\omega_1, \omega_2) = \beta Y(\omega_1, \omega_2) + (1 - \beta H(\omega_1, \omega_2))X_k(\omega_1, \omega_2)$$

$$X_0 = 0; \quad X_1 = \beta Y; \quad X_2 = \beta Y + (1 - \beta H)X_1 = \beta Y + (1 - \beta H)\beta Y$$

$$X_3 = \beta Y + (1 - \beta H)(\beta Y + (1 - \beta H)\beta Y) =$$

$$= \beta Y + (1 - \beta H)\beta Y + (1 - \beta H)^2 \beta Y = \beta Y(1 + (1 - \beta H) + (1 - \beta H)^2)$$

$$X_k(\omega_1, \omega_2) = R_k(\omega_1, \omega_2)Y(\omega_1, \omega_2)$$

$$R_k(\omega_1, \omega_2) = \beta \sum_{l=0}^{k-1} (1 - \beta H(\omega_1, \omega_2))^l$$



X_k converges if :

$$|1 - \beta H(\omega_1, \omega_2)| < 1$$

Iterative Restoration Algorithms

◉ Iterative Least-Squares:

$$\Phi(x) = \frac{1}{2} \nabla_x \|y - Hx\|^2$$

$$x_{k+1} = x_k + \beta H^T (y - Hx_k) = \beta H^T y + (I - \beta H^T H)x_k$$

- In the frequency domain (assuming H is block circular):

$$X_{k+1}(\omega_1, \omega_2) = \beta H^*(\omega_1, \omega_2) Y(\omega_1, \omega_2) + \left(1 - \beta |H(\omega_1, \omega_2)|^2\right) X_k(\omega_1, \omega_2)$$

◉ Iterative Constrained Least-Squares:

$$\Phi(x) = \frac{1}{2} \nabla_x \left(\|y - Hx\|^2 + \alpha \|Cx\|^2 \right)$$

$$x_{k+1} = \beta H^T y + \left(I - \beta (H^T H + \alpha C^T C) \right) x_k$$

- In the frequency domain (assuming H is block circular):

$$X_{k+1}(\omega_1, \omega_2) = \beta H^*(\omega_1, \omega_2) Y(\omega_1, \omega_2) + \left(1 - \beta \left(|H(\omega_1, \omega_2)|^2 + \alpha |C(\omega_1, \omega_2)|^2 \right) \right) X_k(\omega_1, \omega_2)$$

Iterative Restoration Algorithms



Original Image



Blurred Image with Additional Noise



Reconstructed with LS



Reconstructed with ILS

Iterative Restoration Algorithms



Original Image



Blurred Image with Additional Noise



Reconstructed with CLS



Reconstructed with ICLS

Wiener Filter

● Stochastic restoration approach:

- Treat the image as a sample from a 2D random field.
- The image is part of a class of samples (an ensemble), realizations of the same random field.

$$R(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2) \cdot P_{xx}(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2 \cdot P_{xx}(\omega_1, \omega_2) + P_{NN}(\omega_1, \omega_2)}$$

- Autocorrelation:

$$R_{ff}(n_1, n_2, n_3, n_4) = E[f(n_1, n_2) f^*(n_3, n_4)]$$

Expected value over many realizations of the 2D random field

- Power-Spectrum:

$$P_{ff}(\omega_1, \omega_2) = F\{R_{ff}(d_1, d_2)\}$$

Fourier transformation

Wiener Filter and CLS Filter

- Wiener filter:

$$\begin{aligned} R(\omega_1, \omega_2) &= \frac{H^*(\omega_1, \omega_2) \cdot P_{xx}(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2 \cdot P_{xx}(\omega_1, \omega_2) + P_{NN}(\omega_1, \omega_2)} = \\ &= \frac{H^*(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2 + \frac{P_{NN}(\omega_1, \omega_2)}{P_{xx}(\omega_1, \omega_2)}} = \frac{H^*(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2 + \frac{\sigma_N^2}{P_{xx}(\omega_1, \omega_2)}} \end{aligned}$$

Assuming white noise

- CLS filter:

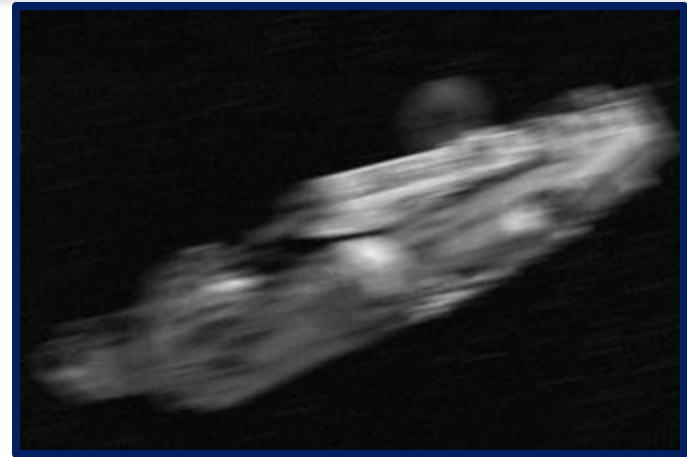
$$R(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2 + \alpha |C(\omega_1, \omega_2)|^2}$$

- With the right choice of C and α , CLS filter is the same as the Wiener filter.

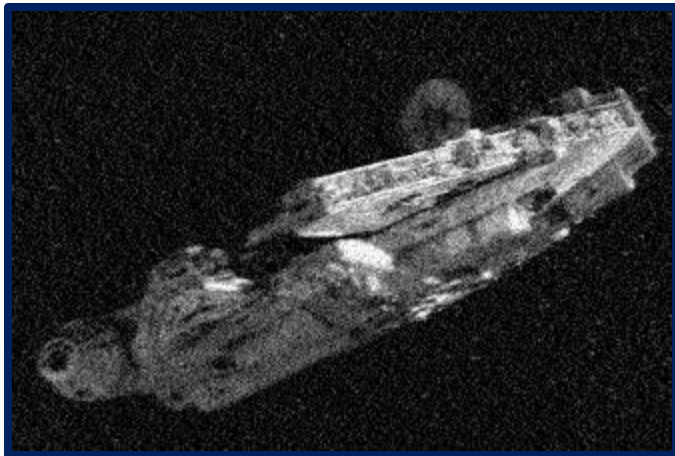
Wiener Filter



Original Image



Motion Blurred Noisy Image



Reconstructed with Wiener filter



Reconstructed with CLS filter

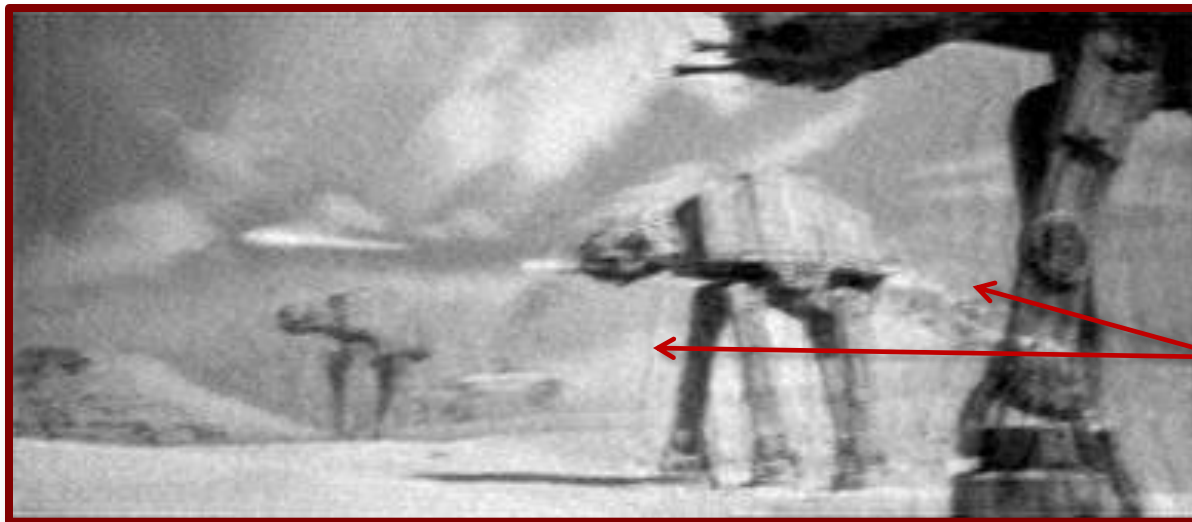
Ringling Artifact



Original Image



Motion Blurred Noisy Image



Reconstructed with CLS filter

← ← Ringing artifact

Ringling Artifact

- If we convolve the degradation and the restoration filters:

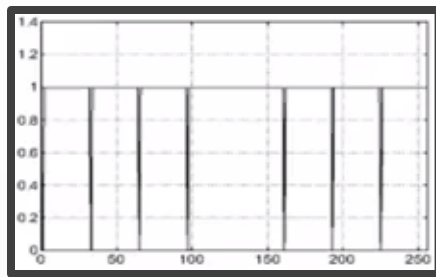
$$s_{all}(n_1, n_2) = h(n_1, n_2) * r(n_1, n_2)$$

$$\hat{x}(n_1, n_2) = s_{all} * x(n_1, n_2)$$

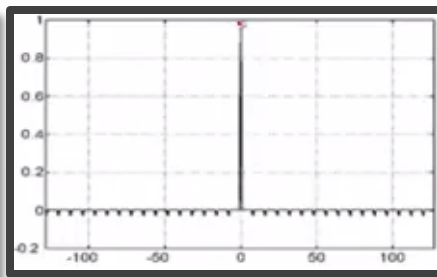
- In an ideal case:

$$s_{all}(n_1, n_2) = \delta(n_1, n_2)$$

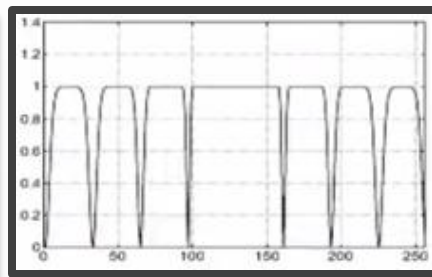
$$S_{all}(\omega_1, \omega_2) = 1, \quad \text{for } \forall \omega_1, \omega_2$$



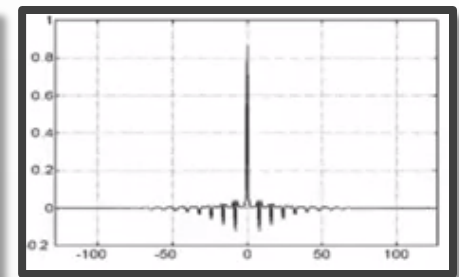
S_{all} for 1D motion blur,
with direct Inverse Filter



s_{all} for 1D motion blur,
with direct Inverse Filter

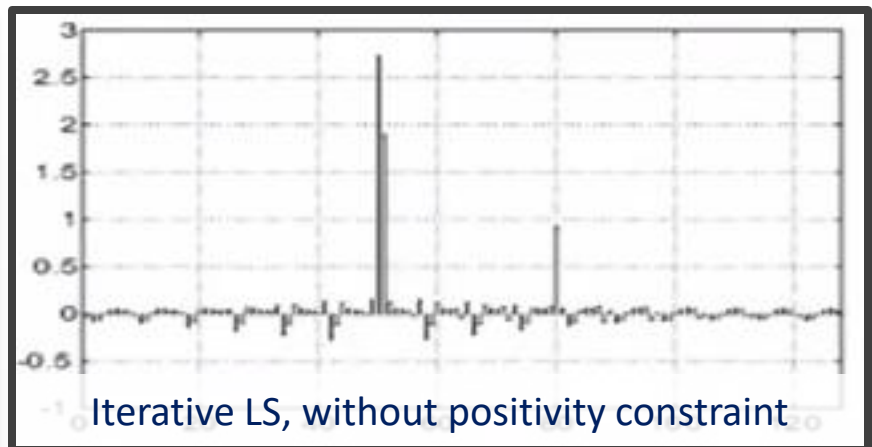
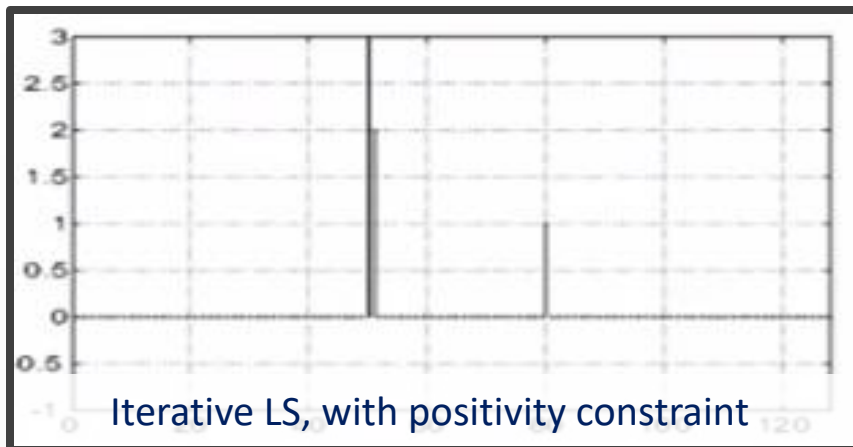
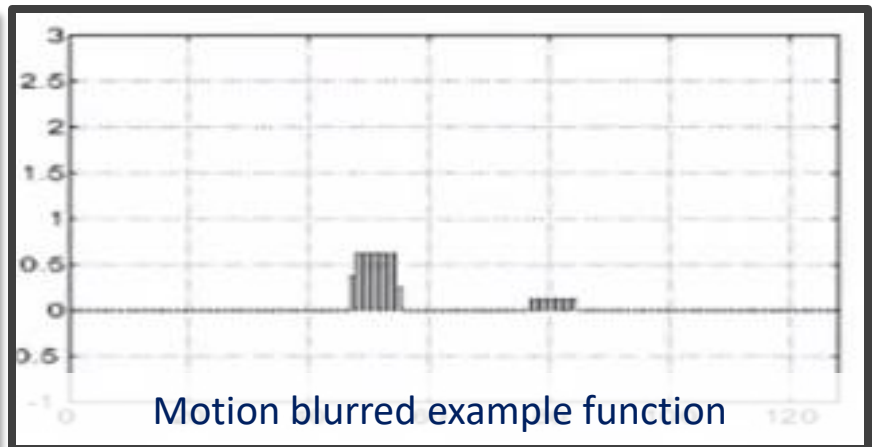
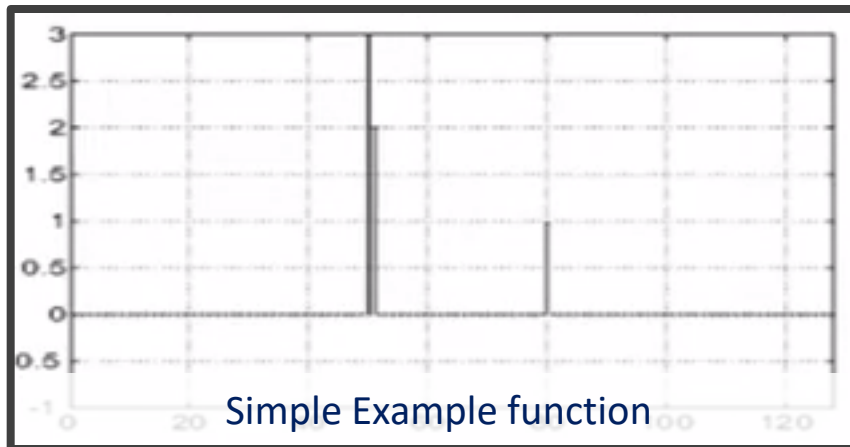


S_{all} for 1D motion blur,
with iterative LS

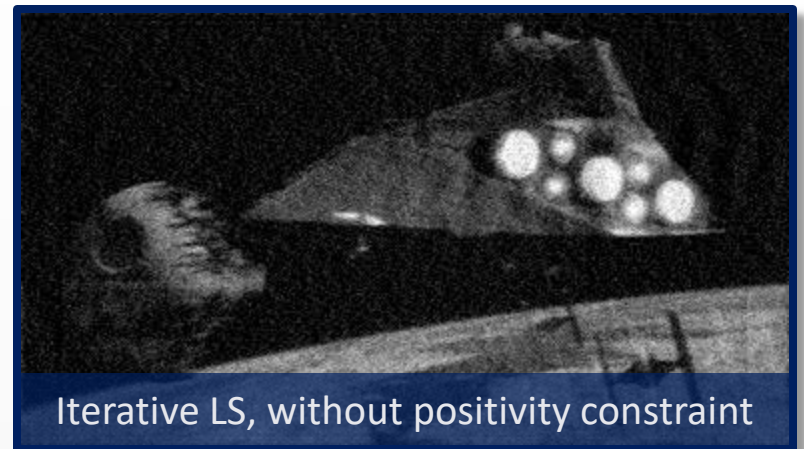


s_{all} for 1D motion blur,
with iterative LS

Restoration with Positivity Constraint



Restoration with Positivity Constraint



Degradation/Restoration Metrics

- Signal-to-Noise Ratio:

$$SNR = 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_n^2} \right)$$

- Blurred Signal-to-Noise Ratio:

$$BSNR = 10 \log_{10} \left(\frac{\frac{1}{MN} \sum_i \sum_j [g(i, j) - \bar{g}(i, j)]^2}{\sigma_n^2} \right), \quad \text{where} \quad \begin{aligned} g(i, j) &= x(i, j) * h(i, j) \\ \bar{g}(i, j) &= E\{g(i, j)\} \\ \sigma_n^2 &: \text{variance of the noise} \end{aligned}$$

- Improvement in Signal-to-Noise Ratio:

- ISNR is computable only in a simulation environment, where the original image is available.

$$ISNR = 10 \log_{10} \left(\frac{\sum_i \sum_j [x(i, j) - y(i, j)]^2}{\sum_i \sum_j [x(i, j) - \tilde{x}(i, j)]^2} \right)$$

Main Sources and Further Readings

- ◉ Fundamentals of Digital Image and Video Processing lectures by Aggelos K. Katsaggelos
- ◉ Babacan, D. S., R. Molina, and A. K. Katsaggelos, "Total Variation Super Resolution Using A Variational Approach", IEEE International Conf. on Image Processing 2008, San Diego, USA, 10/10/2008.
- ◉ J. Rombaut, A. Pizurica, and W. Philips, "Locally adaptive passive error concealment for wavelet coded images," IEEE Signal Processing Letters, vol. 15, pp. 178-181, 2008.
- ◉ Zhaofu Chen; Derin Babacan, S.; Molina, R.; Katsaggelos, AK., "Variational Bayesian Methods For Multimedia Problems," *Multimedia, IEEE Transactions on* , vol.16, no.4, pp.1000,1017, June 2014
- ◉ <http://www.mathworks.com/company/newsletters/articles/applying-modern-pde-techniques-to-digital-image-restoration.html>
- ◉ S. Alireza Golestaneh, D. M. Chandler, "An Algorithm for JPEG Artifact Reduction via Local Edge Regeneration" *Journal of Electronic Imaging (JEI)*, Jan 2014