

Restauración

Definición del problema

- La imagen captada z es una versión degradada de la imagen x
- Objetivo: estimar x lo mejor posible.

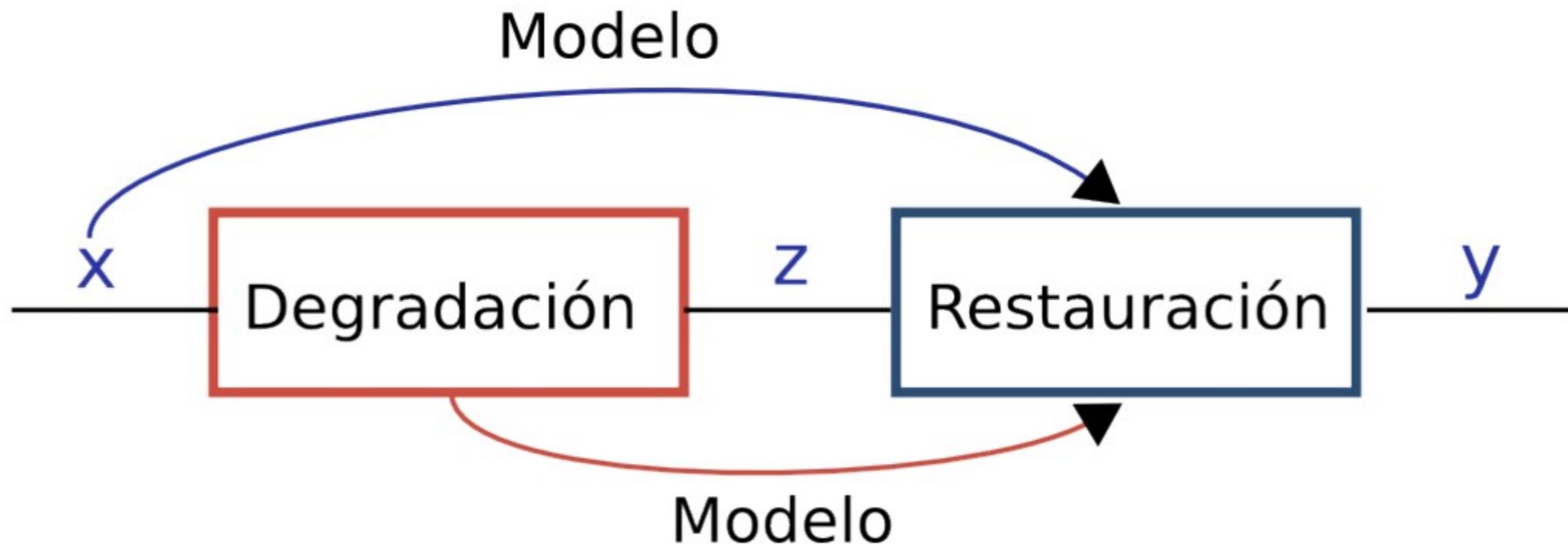


Restauración vs. Mejoramiento

- Mejoramiento
 - Resaltar aspectos de la imagen observada a los efectos de facilitar su visualización
 - No se busca fidelidad respecto a datos originales.
- Restauración
 - Recuperar la imagen original a partir de la versión degradada que se observa
 - Se utiliza información a priori del proceso de degradación y de propiedades conocidas de la imagen original.
 - Se busca máxima fidelidad respecto a datos “verdaderos”.

Información útil para la restauración

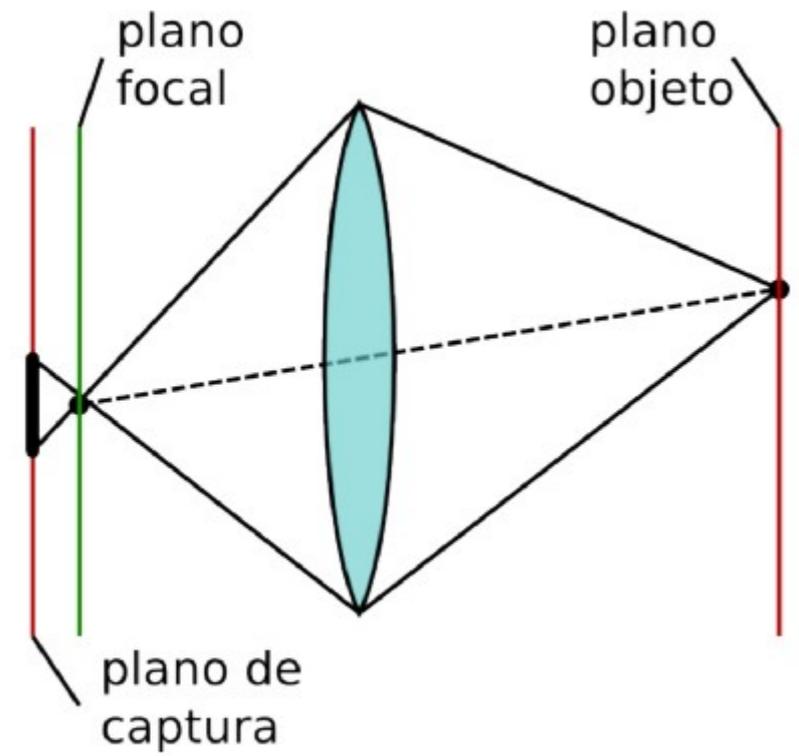
- Es fundamental aprovechar todo el conocimiento a priori disponible para obtener un buen resultado.
 - Modelo de degradación
 - Modelo de la imagen no degradada.



Modelos de degradación

- Modelo físico del proceso de captura
- Estructura
- Propiedades estadísticas
- Ejemplos: desenfoque, borroneo por movimiento (motion blur), ruido térmico.

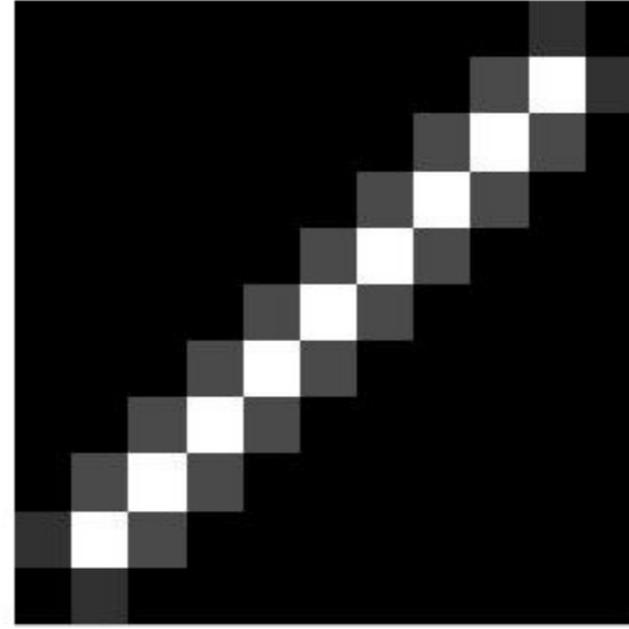
Desenfoque



Borroneo por movimiento lineal



Original image



Original PSF



Blurred image

Ruido



$$y_{i,j} = x_{i,j} + n_{i,j}, n_{i,j} \sim \mathcal{N}(0, \sigma^2)$$

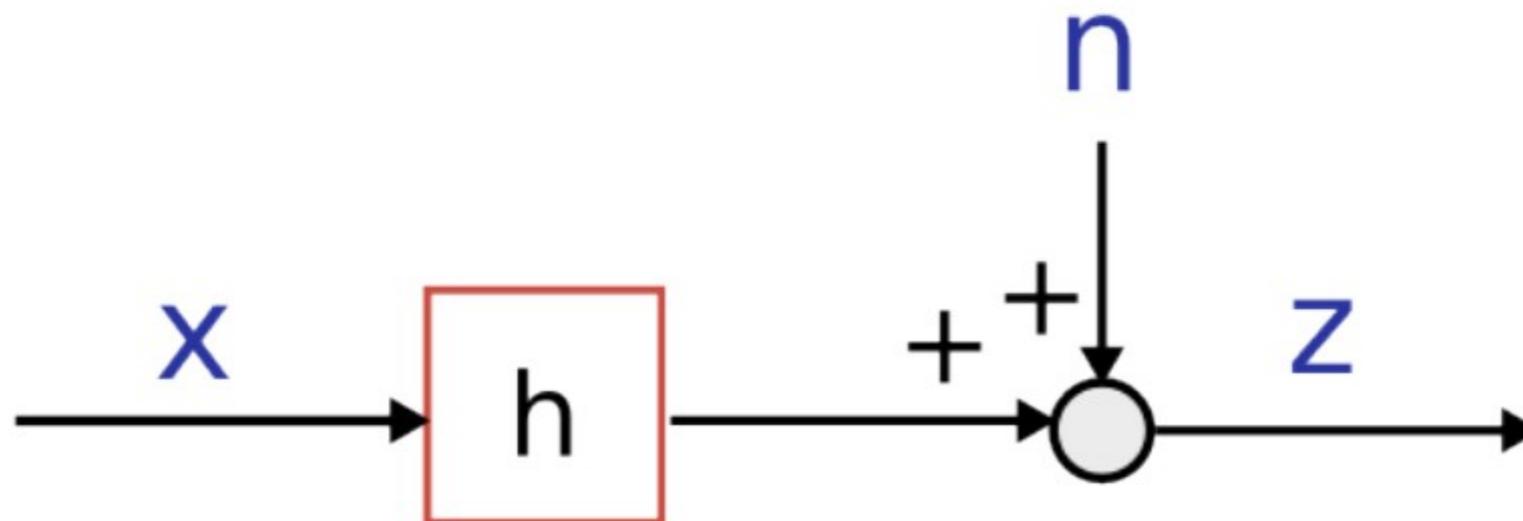
Ruido

- Independiente de la señal
 - Aditivo
 - Multiplicativo
- Dependiente de la señal
- Ej. Ruido electrónico, ruido de cuantización, speckle.

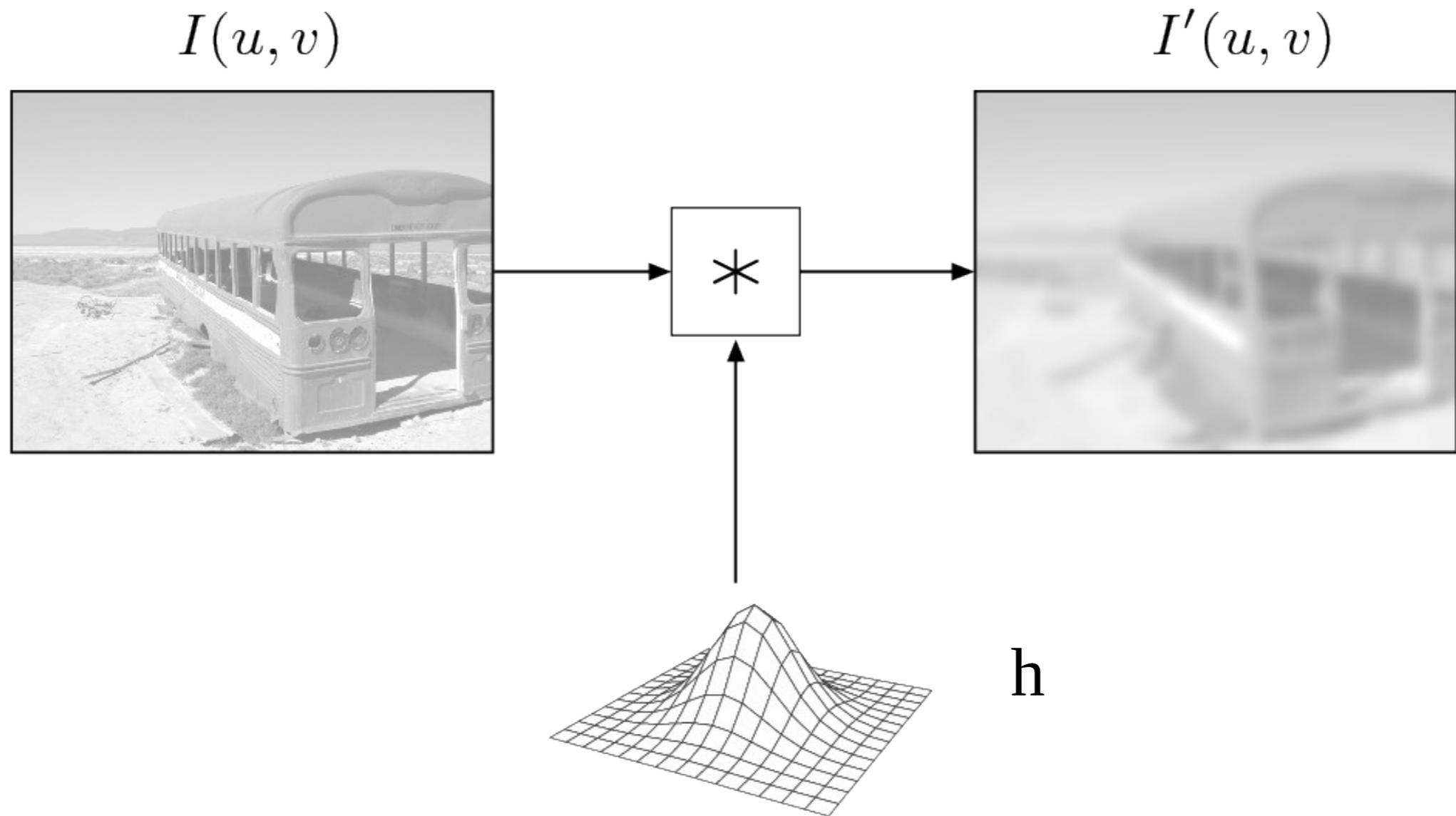
Modelo combinado lineal

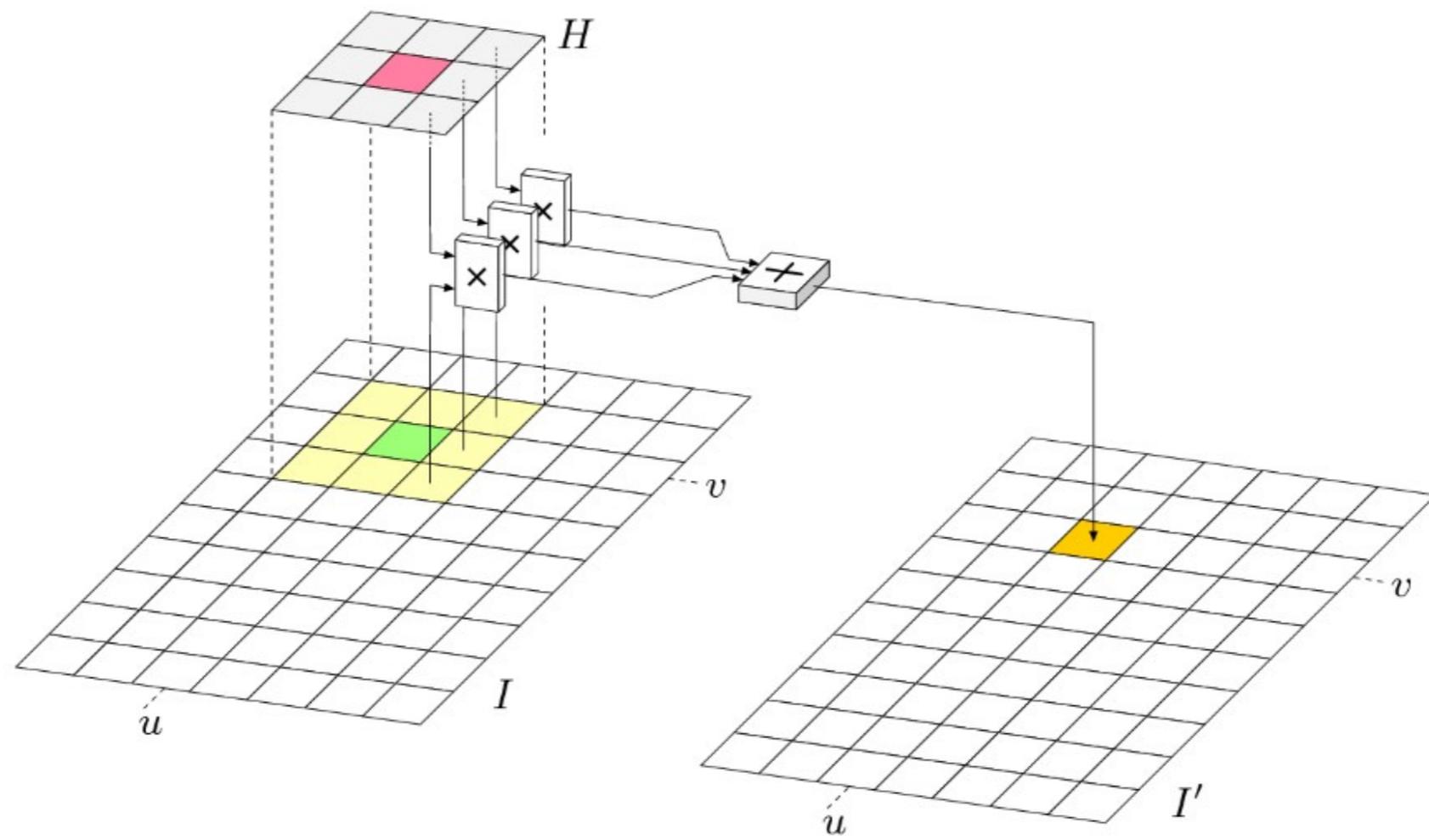
- borroneo: degradación lineal e invariante en el espacio
 - función de transferencia invariante en el espacio, h , también llamada PSF (Point Spread Function).
 - Respuesta frecuencial $H(u, v)=F[h]$ se la suele llamar MTF (Modulation Transfer Function)
- ruido: aditivo, usualmente i.i.d. y de media nula.

$$z = h * x + n$$

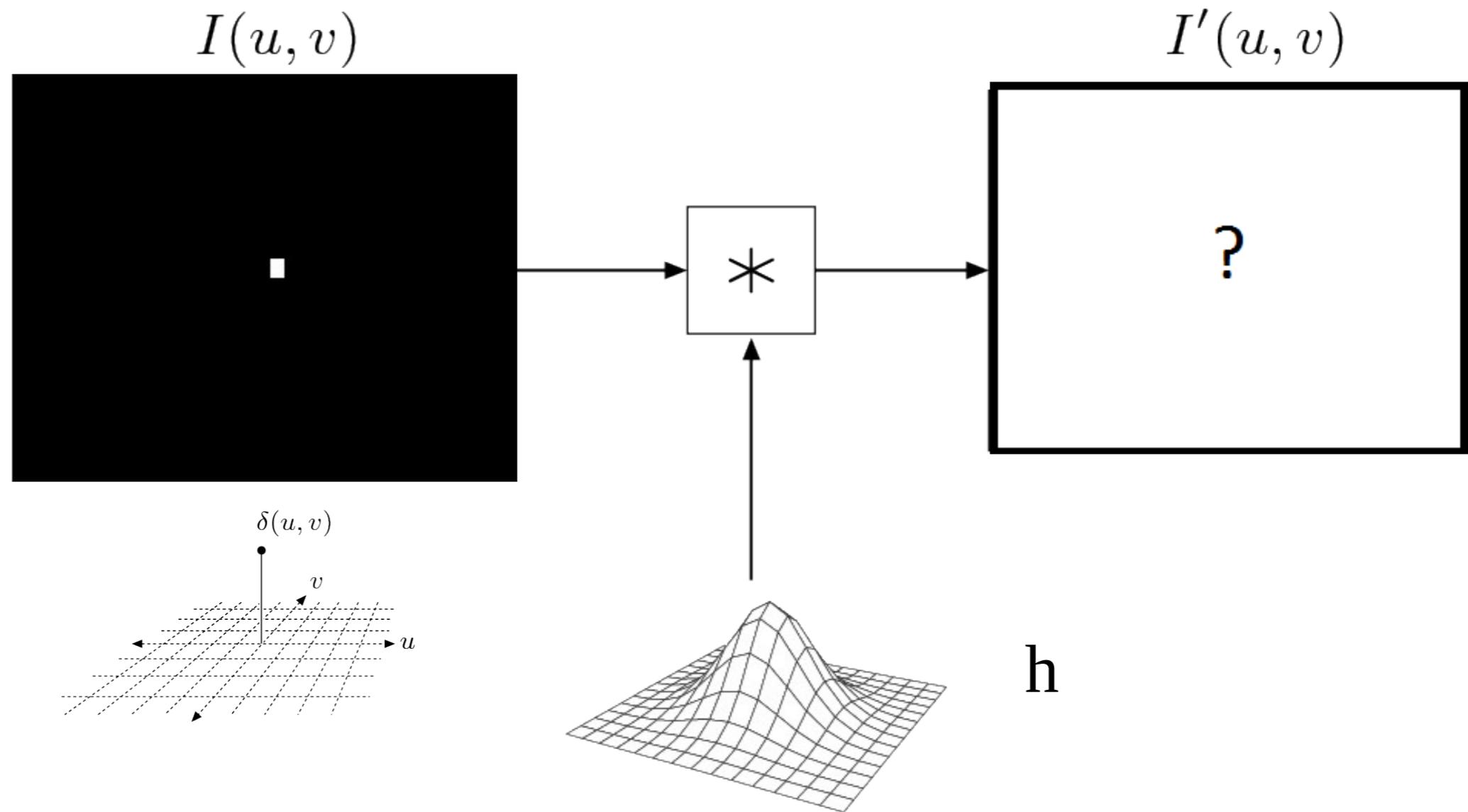


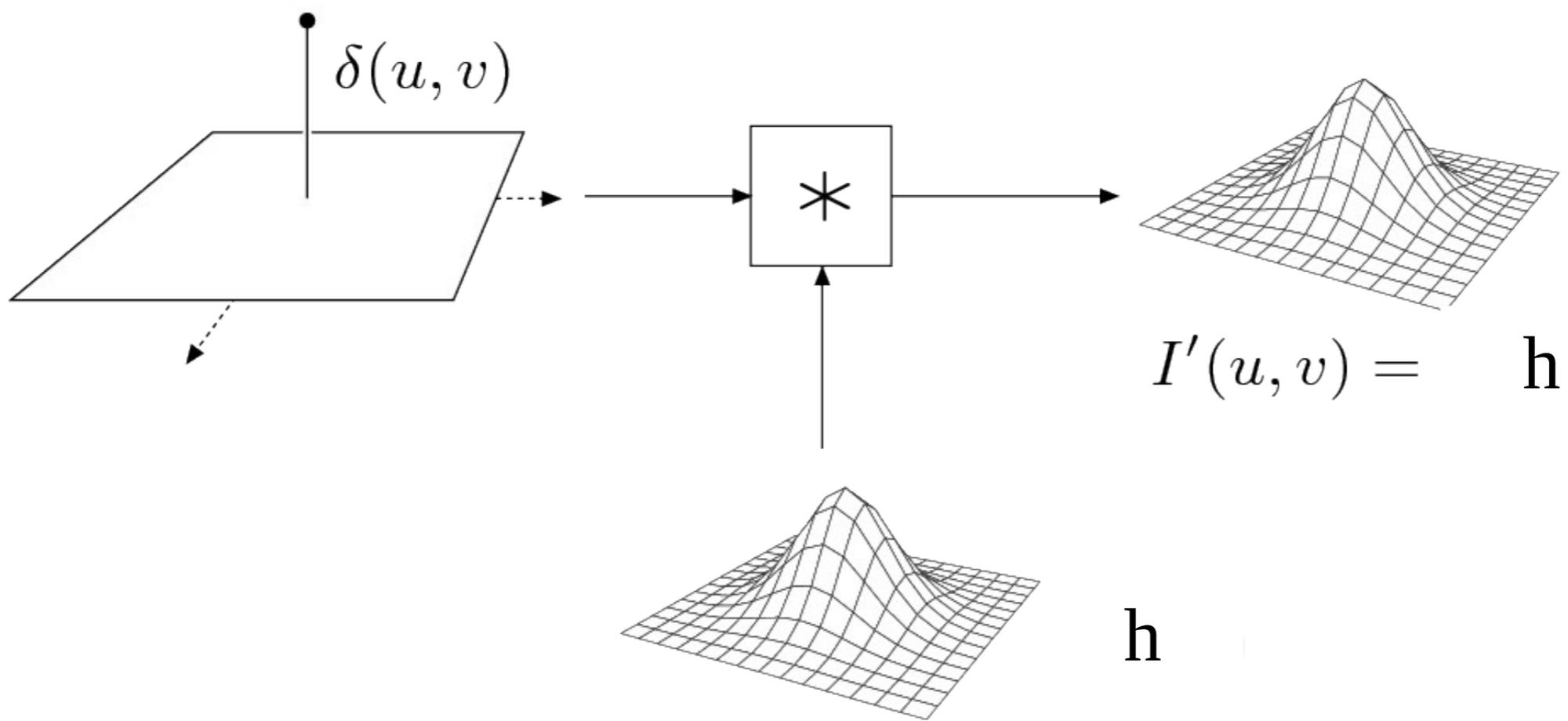
- Degradación por borroneo





- Cómo es la salida degradada si la imagen de entrada ideal es una delta





Modelos de imagen

- Modelo físico del proceso de captura
- Estructura
- Propiedades estadísticas
- Ejemplos: regiones uniformes, bordes, texturas regulares, patrones repetidos.

Características a tener en cuenta de la imagen

Bordes

Variaciones suaves de luminosidad

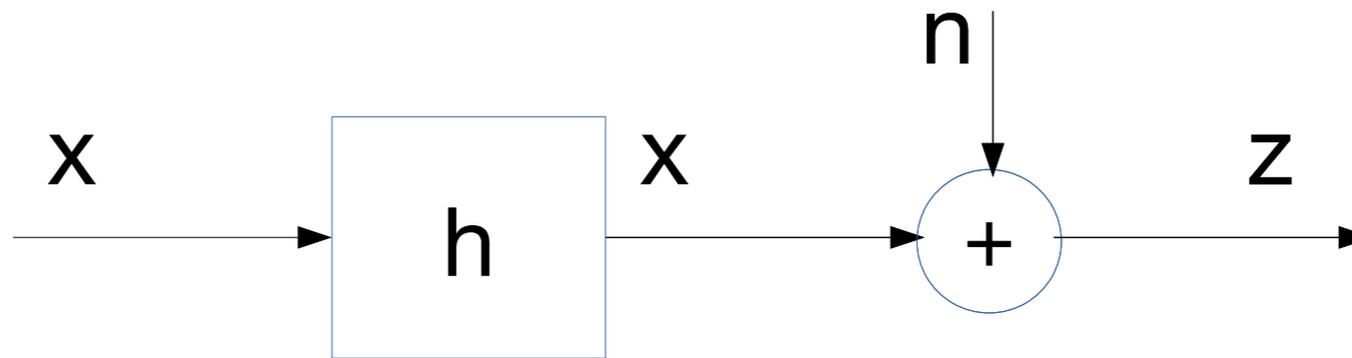


Textura

Patrones que se repiten

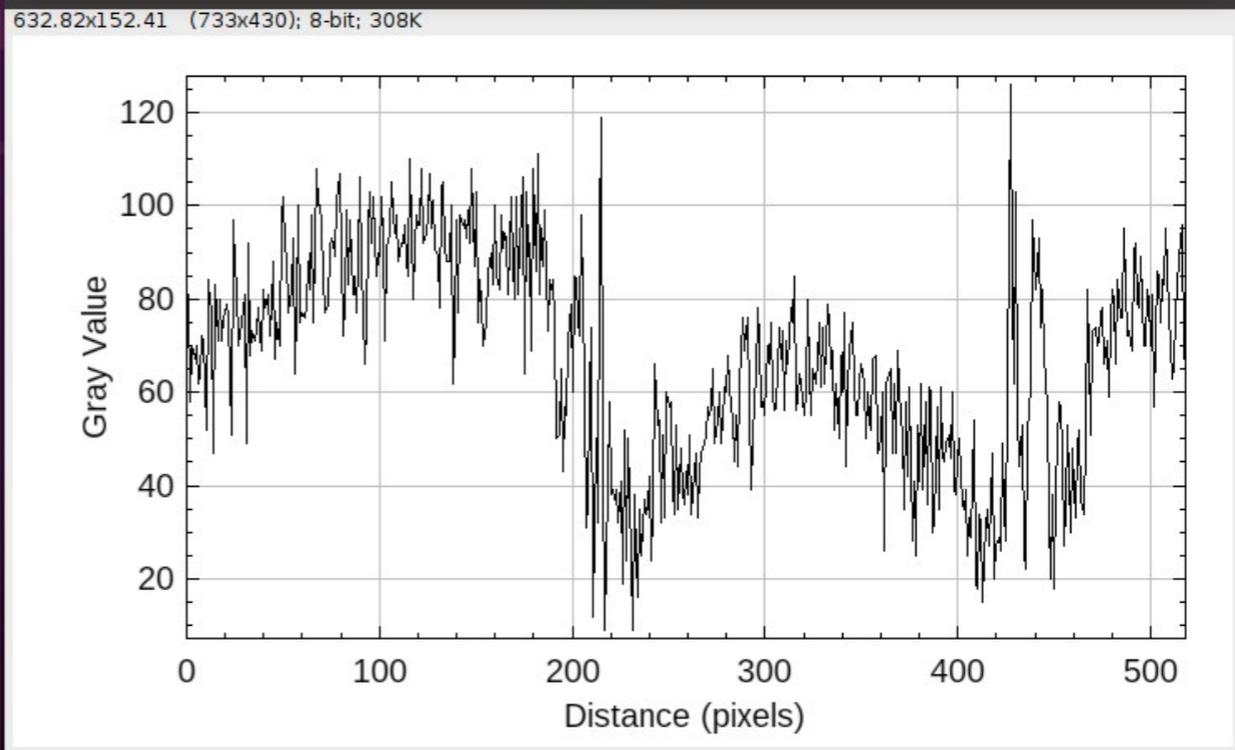
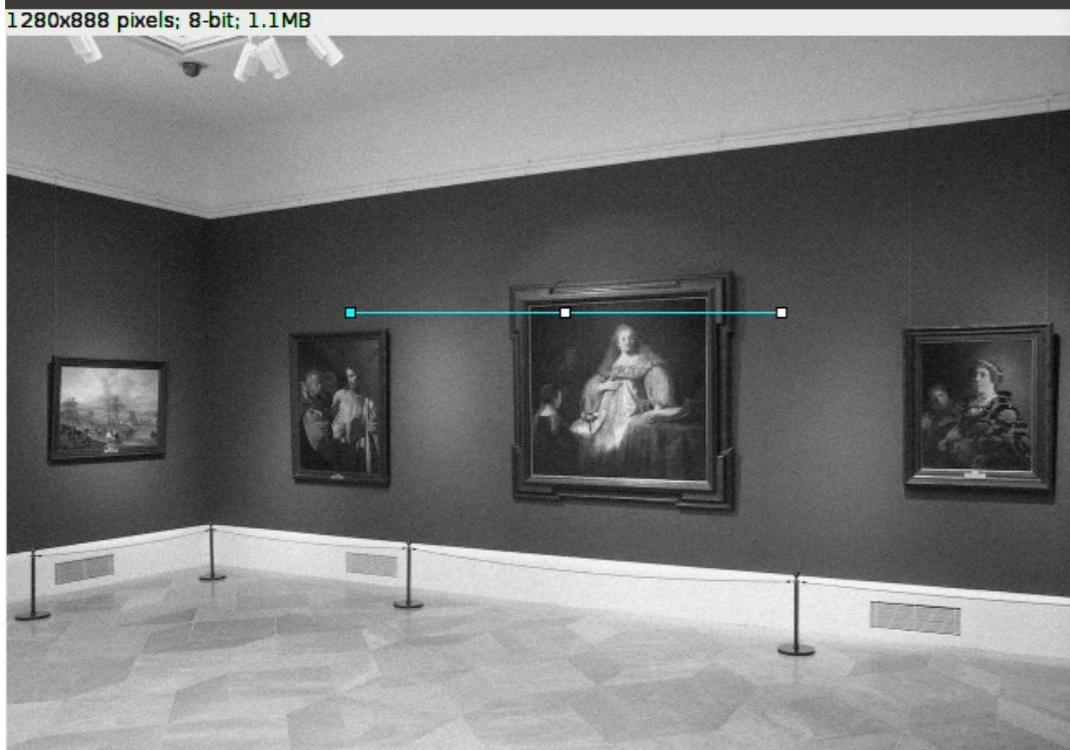
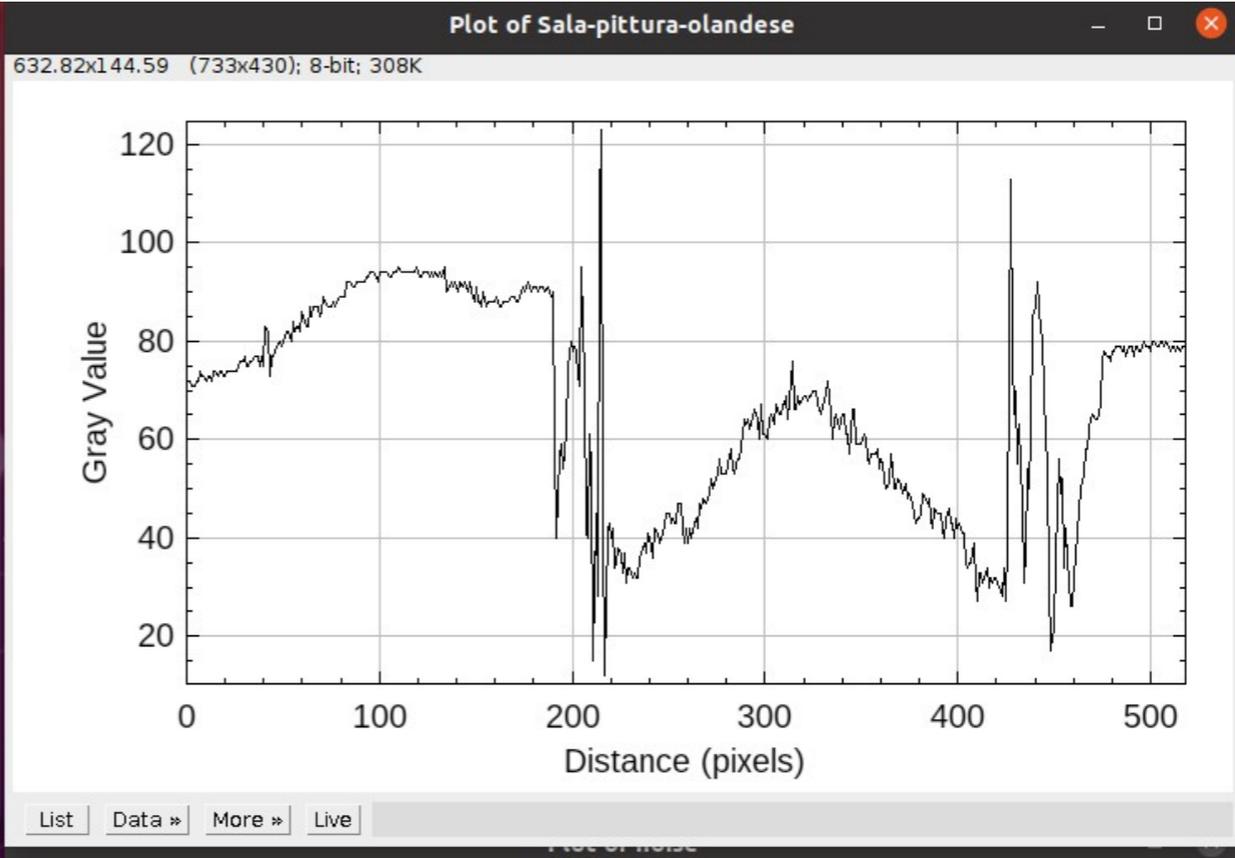
Ruido

- Suponemos que del modelo general sólo tenemos ruido aditivo independiente de la señal



Ruido aditivo

- En muchos caso se modela el ruido aditivo como gaussiano e independiente de la imagen original
- Suposiciones usuales sobre el ruido
 - Sin memoria (i.i.d.)
 - Media nula
 - Independiente de la imagen
- Técnicas de restauración
 - Filtros lineales
 - Filtros adaptivos
 - Otros: NLM, Wavelets



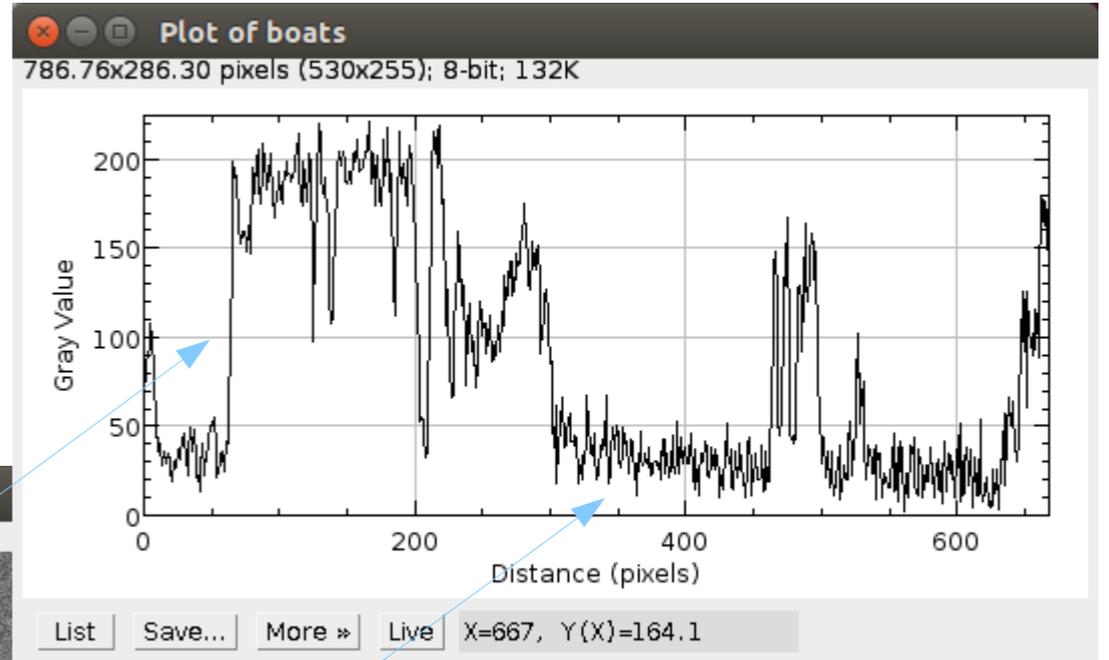
Restauración de ruido aditivo

- Filtro de media
 - Los pixeles cercanos son similares mientras que el ruido varía en forma aleatoria. Al promediar, el ruido se reduce respecto a la señal.
- Filtro de media adaptivo
 - Se requiere conocer la varianza del ruido
 - Sólo se promedia si la relación señal-ruido es baja.

$$y_{i,j} = \mu_{i,j} + \frac{\sigma_{i,j}^2 - \sigma_n^2}{\sigma_{i,j}^2} (z_{i,j} - \mu_{i,j})$$

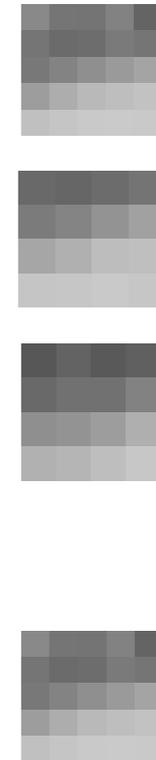
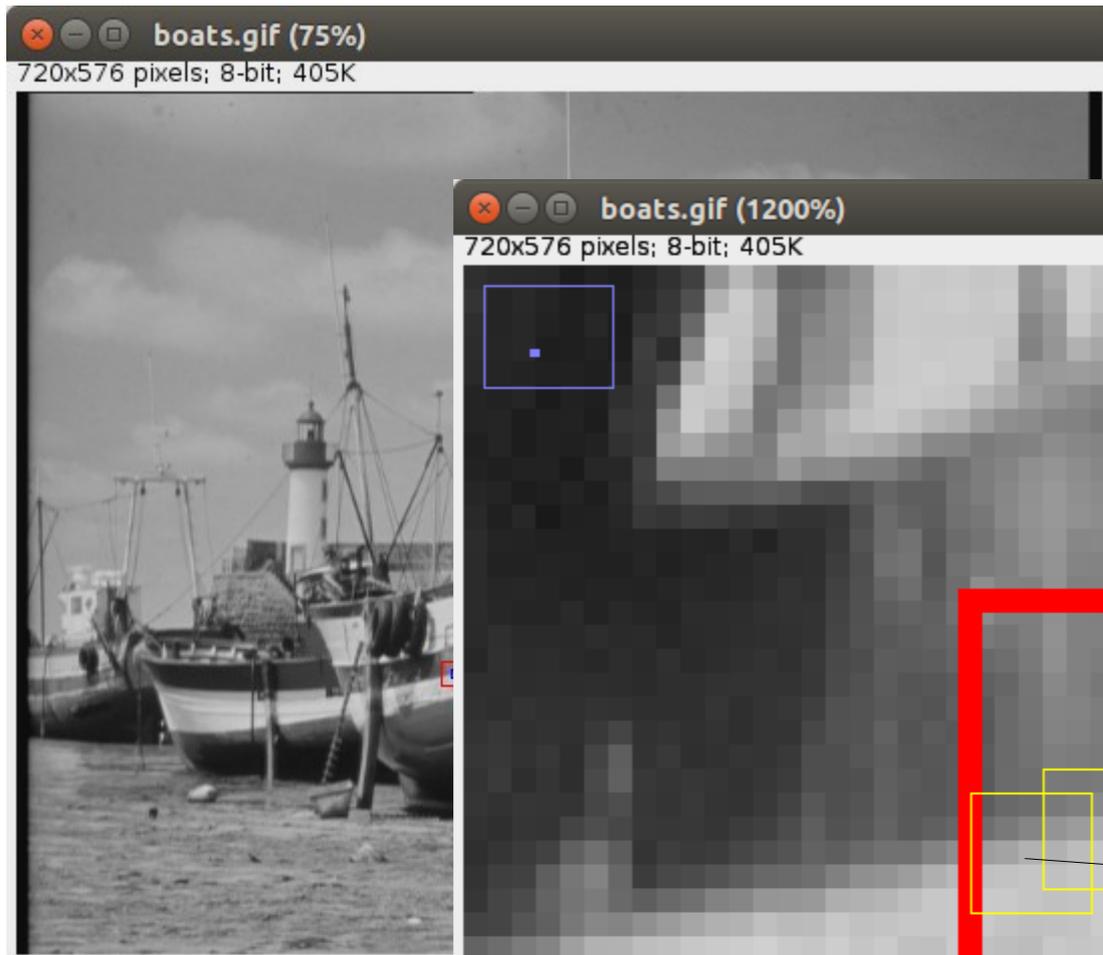
- Non Local Means
 - Explotar el hecho que hay patrones repetidos en la imagen
 - Promediar patrones repetidos
 - http://www.ipol.im/pub/art/2011/bcm_nlm/

Filtrado adaptivo

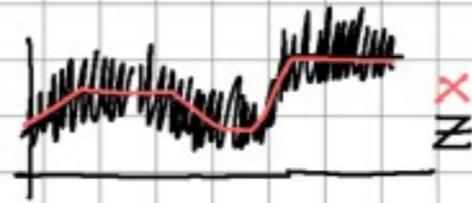


$$y_{i,j} = \mu_{i,j} + \frac{\sigma_{i,j}^2 - \sigma_n^2}{\sigma_{i,j}^2} (z_{i,j} - \mu_{i,j})$$

Non local means



Total variation denoising



$$V(z) = \sum_n |z[n+1] - z[n]| \quad \text{variación total de } z$$

Reducir el ruido de z es encontrar una señal y → "parecida" a z (a)
↳ de menor variación total (b)

$$(a) \quad E(y|z) = \sum_n (y[n] - z[n])^2 \quad \text{debe ser chico}$$

$$(b) \quad V(y) \quad \text{debe ser chico}$$

Las condiciones son antagónicas!

(a) Fidelidad a los datos

(b) Regularidad del resultado

Hay que balancear los términos

$$\min_y [V(y) + \lambda E(y|z)]$$

$$\begin{aligned} \boxed{2D} \quad v(y) &= \sum_{i,j} \sqrt{|y(i+1,j) - y(i,j)|^2 + |y(i,j+1) - y(i,j)|^2} \\ &= \sum_{i,j} |\nabla y \cdot (i,j)| \\ \min_y [v(y) + \lambda \mathbb{E}(y, z)] \end{aligned}$$

Total variation denoising

- Rudin, Osher y Fratermi (ROF),

$$\min_{u \in \text{BV}(\Omega)} \|u\|_{\text{TV}(\Omega)} + \frac{\lambda}{2} \int_{\Omega} (f - u)^2 dx$$

$$\min_{u \in \text{BV}(\Omega)} \int_{\Omega} \left[\|\nabla u\| + \frac{\lambda}{2} (f - u)^2 \right] dx$$

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{\nabla u}{\|\nabla u\|} \right) + \lambda(f - u)$$

Ruido no aditivo

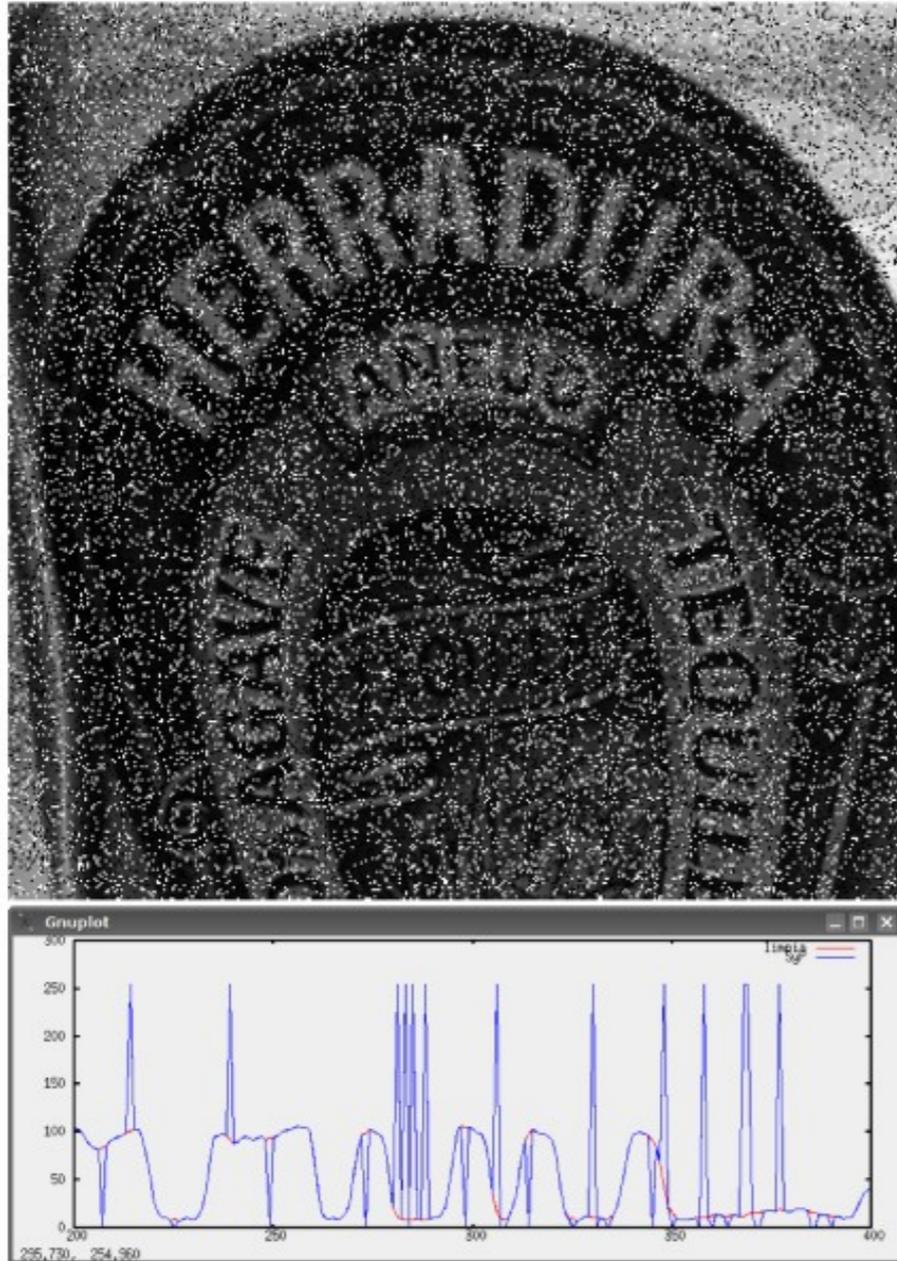


Figura : Ruido Sal y Pimienta

Multiplicativo:

$$z_{i,j} = x_{i,j} \alpha_{i,j}, \quad \alpha_{i,j} \sim P$$

Impulsivo (sal y pimienta):

$$z_{i,j} = \begin{cases} \{0, M - 1\} & , \quad p = \delta/2 \\ x_{i,j} & , \quad p = 1 - \delta \end{cases}$$

Ruido impulsivo (no aditivo)

- Filtro de mediana



¿Cuál restauración es mejor ?

y1

x

z



y2

Medidas de desempeño de la restauración

- Subjetivas
 - Calidad percibida subjetivamente.
 - Inteligibilidad del resultado.
 - Aspecto agradable.
- Objetivas (clásicas)
 - Error cuadrático medio MSE, NMSE
 - Relación señal a ruido: SNR, PSNR
 - Fáciles de medir y de formular matemáticamente.
 - Marco de referencia común.
 - No necesariamente se condicen con las subjetivas.

$r(x,y)$ imagen de referencia, $t(x,y)$ imagen de test

SNR	Signal-to-noise ratio expressed in dB	$\text{SNR} = 10 \cdot \log_{10} \left[\frac{\sum_0^{n_x-1} \sum_0^{n_y-1} [r(x,y)]^2}{\sum_0^{n_x-1} \sum_0^{n_y-1} [r(x,y) - t(x,y)]^2} \right]$
PSNR	Peak signal-to-noise ratio expressed in dB	$\text{PSNR} = 10 \cdot \log_{10} \left[\frac{\max(r(x,y))^2}{\frac{1}{n_x \cdot n_y} \cdot \sum_0^{n_x-1} \sum_0^{n_y-1} [r(x,y) - t(x,y)]^2} \right]$
RMSE	Root mean square error	$\text{RMSE} = \sqrt{\frac{1}{n_x n_y} \cdot \sum_0^{n_x-1} \sum_0^{n_y-1} [r(x,y) - t(x,y)]^2}$
MAE	Mean absolute error	$\text{MAE} = \frac{1}{n_x n_y} \cdot \sum_0^{n_x-1} \sum_0^{n_y-1} r(x,y) - t(x,y) $

Sala-pittura-olandese-1-crop

363x309 pixels; 8-bit; 110K



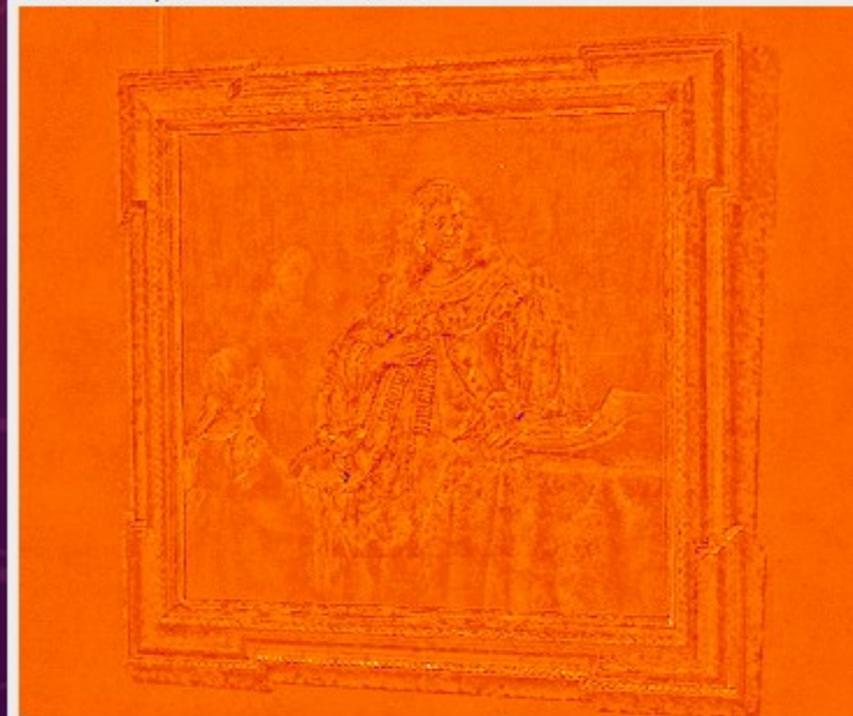
restauracion1-crop

363x309 pixels; 8-bit; 110K



resta-restauracion1-crop

363x309 pixels; 32-bit; 438K



noise-1-crop

363x309 pixels; 8-bit; 110K



restauracion2-crop

363x309 pixels; 8-bit; 110K



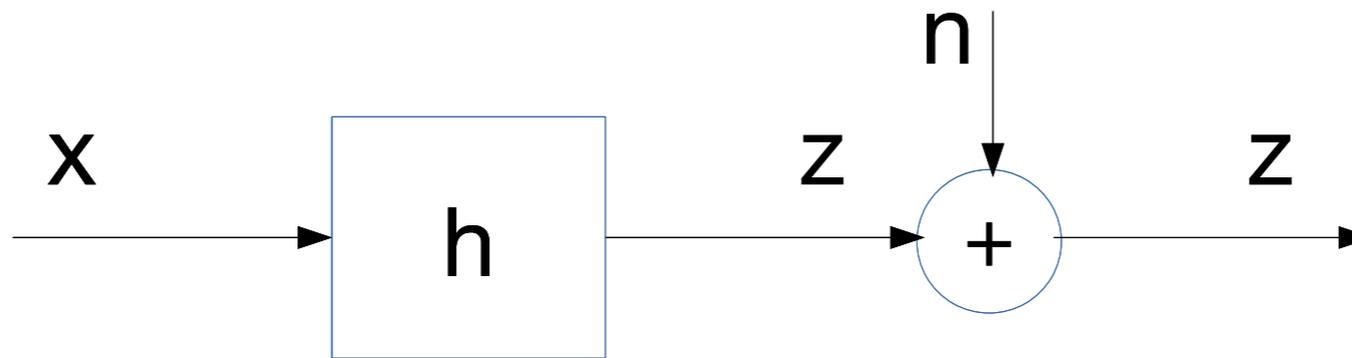
resta-restauracion2-crop

363x309 pixels; 32-bit; 438K



Borroneo (blur)

- Suponemos que del modelo general de degradación sólo tenemos el blur



Borroneo (Blur)

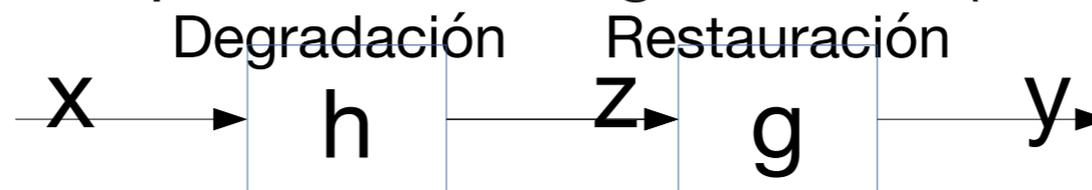
- Modelo

$$z = h * x$$

- h es conocido
- La imagen original x esta convolucionada por el núcleo de convolución h

- Técnicas

- Se busca deshacer el proceso de degradación (realizar la **deconvolución**)



- Se busca obtener un filtro g que aplicado a z me permita recuperar algo lo más parecido a x posible

-

Deconvolución

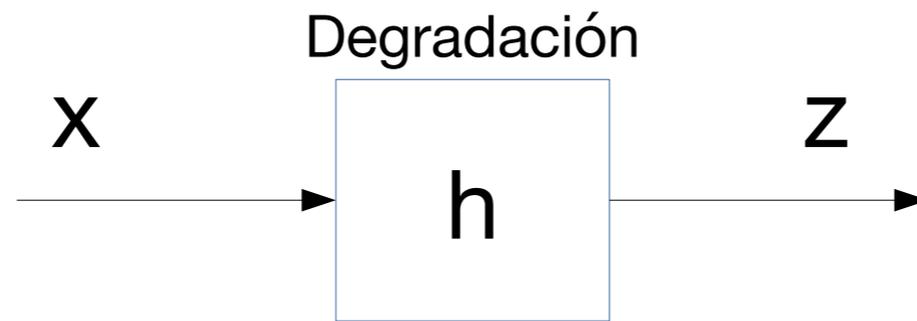
- Con PSF conocida
 - Filtrado (en frecuencia)
 - Cálculo de un filtro inverso al de degradación
 - Métodos iterativos
 - Secuencia de pasos repetitivos que (en ciertas condiciones) se aproximan a una buena restauración.
 - Más costosos computacionalmente, pero usualmente mejores (PSF variable en la imagen)
- Con PSF desconocida
 - Blind deconvolution
Recuperar simultáneamente la PSF y la imagen restaurada

Teorema de convolución

- Propiedad importante de la transformada de Fourier
 - A una operación de convolución en el espacio le corresponde una multiplicación puntual en frecuencia

$$\begin{array}{ccccc} \text{Image space: } & g(u, v) & * & h(u, v) & = & g'(u, v) \\ & \downarrow & & \downarrow & & \uparrow \\ & \text{DFT} & & \text{DFT} & & \text{DFT}^{-1} \\ & \downarrow & & \downarrow & & \uparrow \\ \text{Frequency space: } & G(m, n) & \cdot & H(m, n) & \longrightarrow & G'(m, n) \end{array}$$

Filtrado inverso



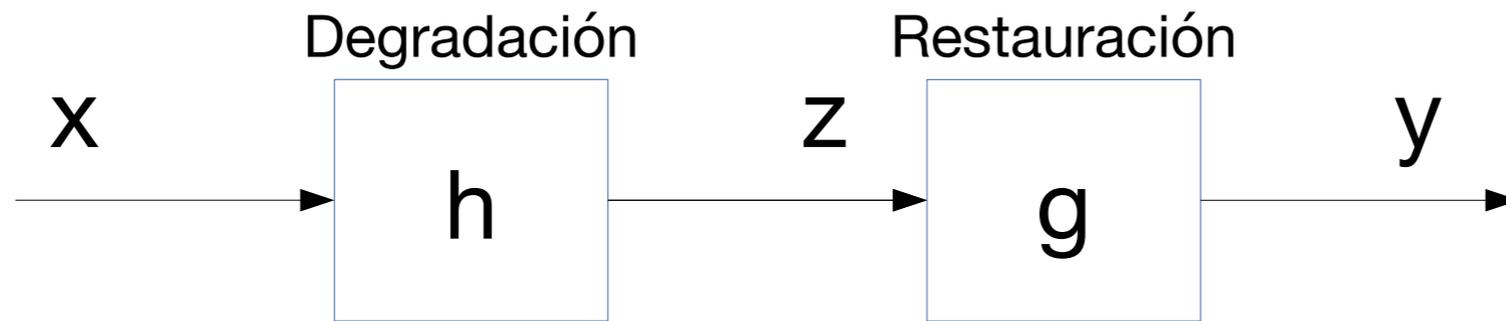
En el
espacio

$$z = x * h$$

En
frecuencia

$$Z = X \cdot H$$

Filtrado inverso



En el
espacio

$$y = z * g = (x * h) * g$$

En
frecuencia

$$Y = Z \cdot G = (X \cdot H) \cdot G$$

Queremos que **Y** sea parecido a **X**
Cómo debería ser **G** ?

$$G = \frac{1}{H}$$

Filtrado inverso

- Problemas:
 - Para las frecuencias que H es chico la inversión crece mucho
 - G no está definido para las frecuencias donde H se hace cero
 - Supusimos que no teníamos ruido pero eso no es cierto. Al invertir podemos amplificar mucho el ruido.
- En la práctica el filtrado inverso no funciona
- Alternativas

- Filtrado “pseudo inverso”

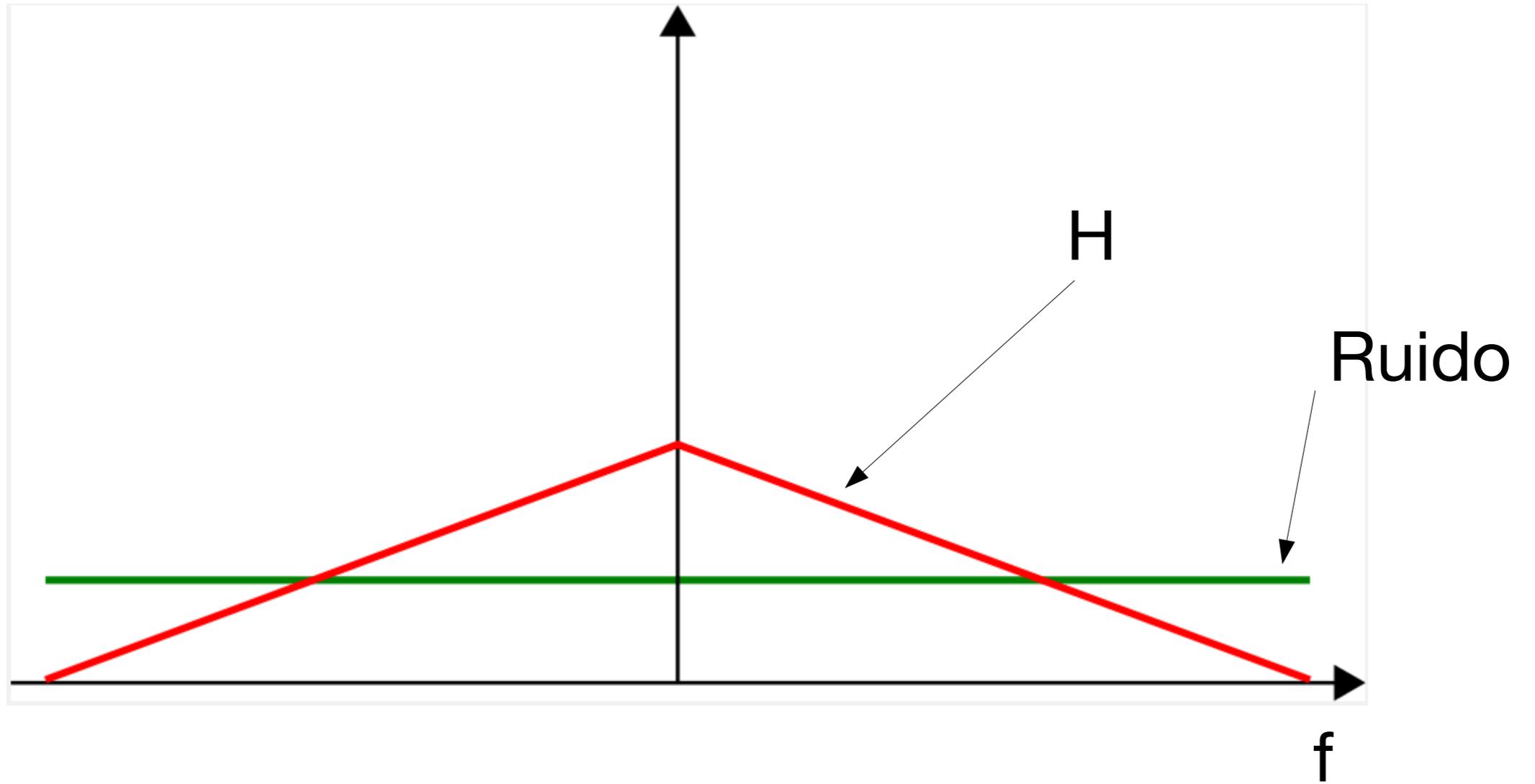
$$G = \frac{1}{H + cte}$$

- Filtrado de Wiener

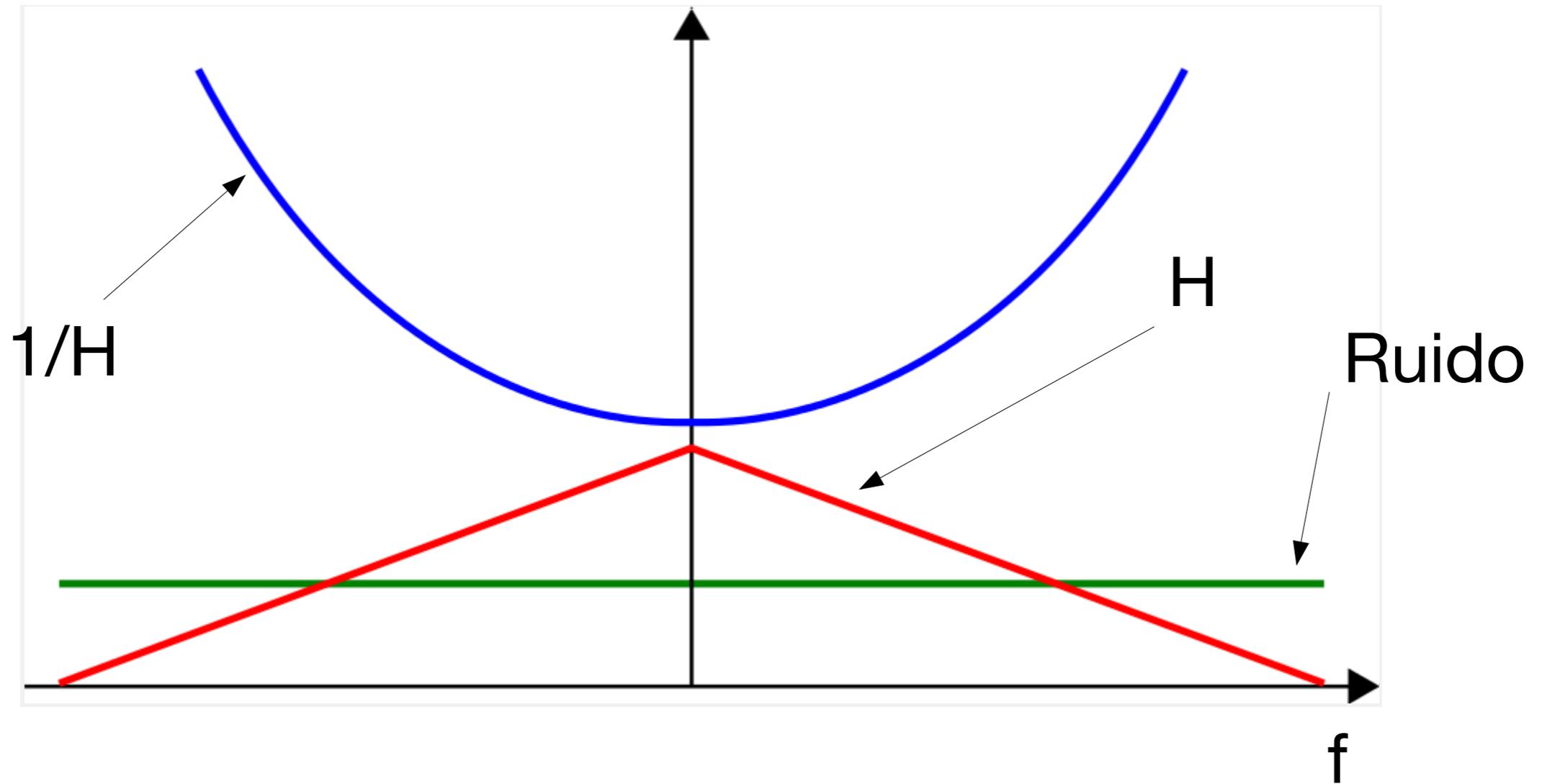
- Se invierte donde la relación señal/ruido es buena

$$G = \frac{1}{H + \frac{1}{SNR}}$$

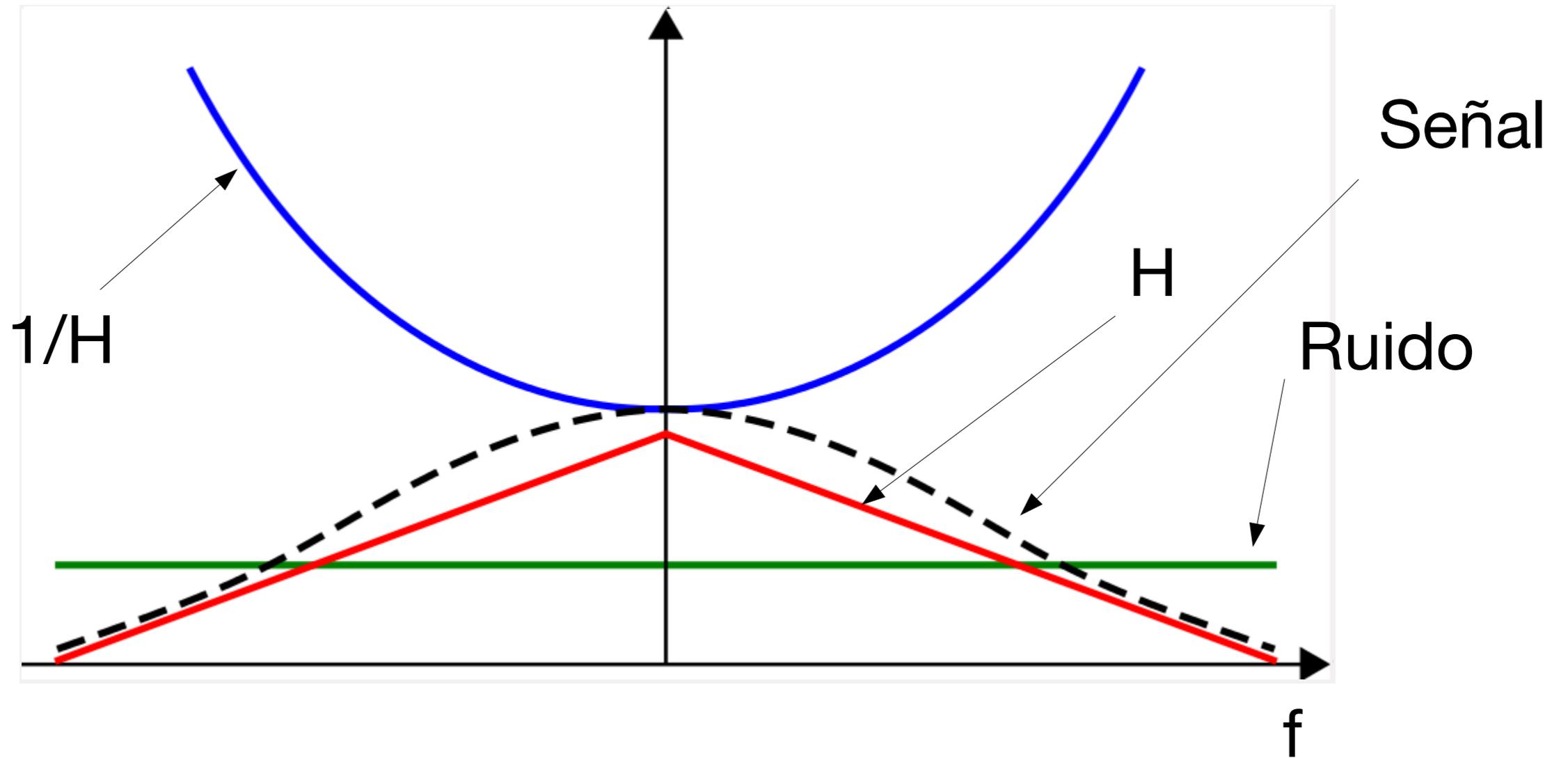
Filtrado inverso



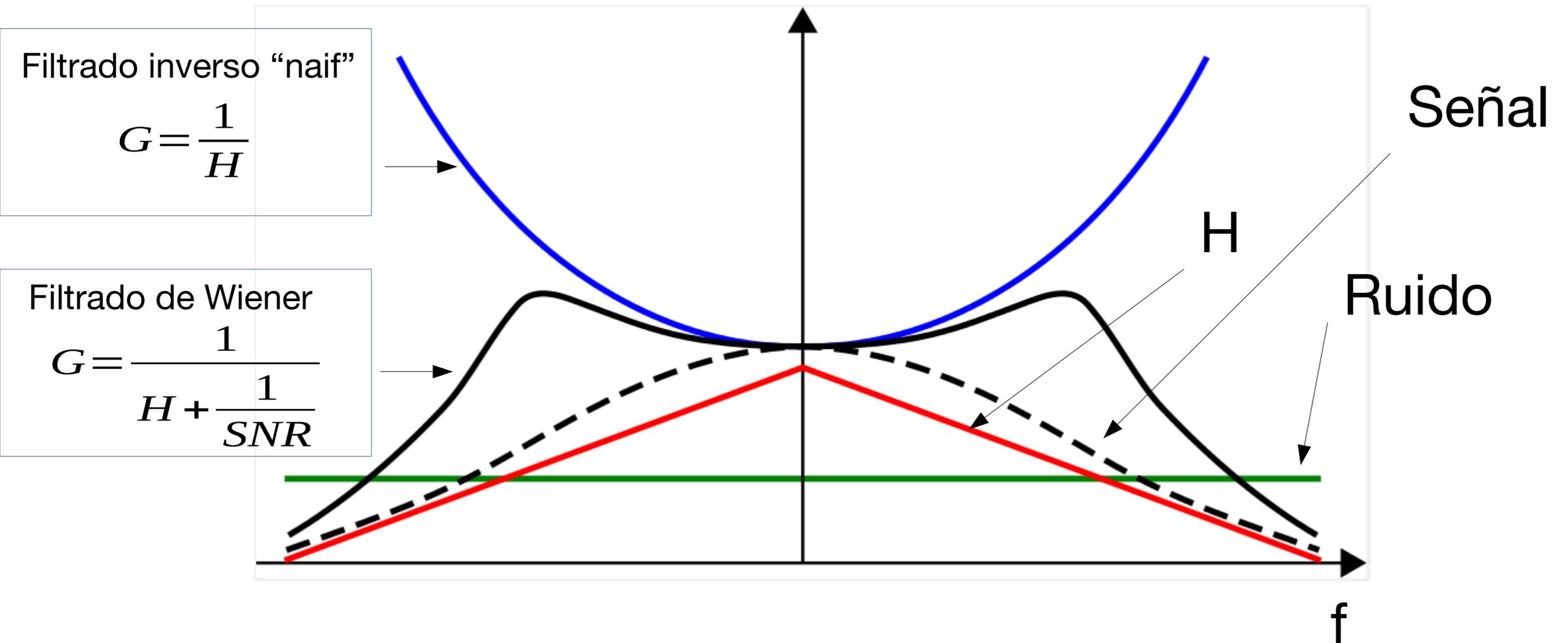
Filtrado inverso



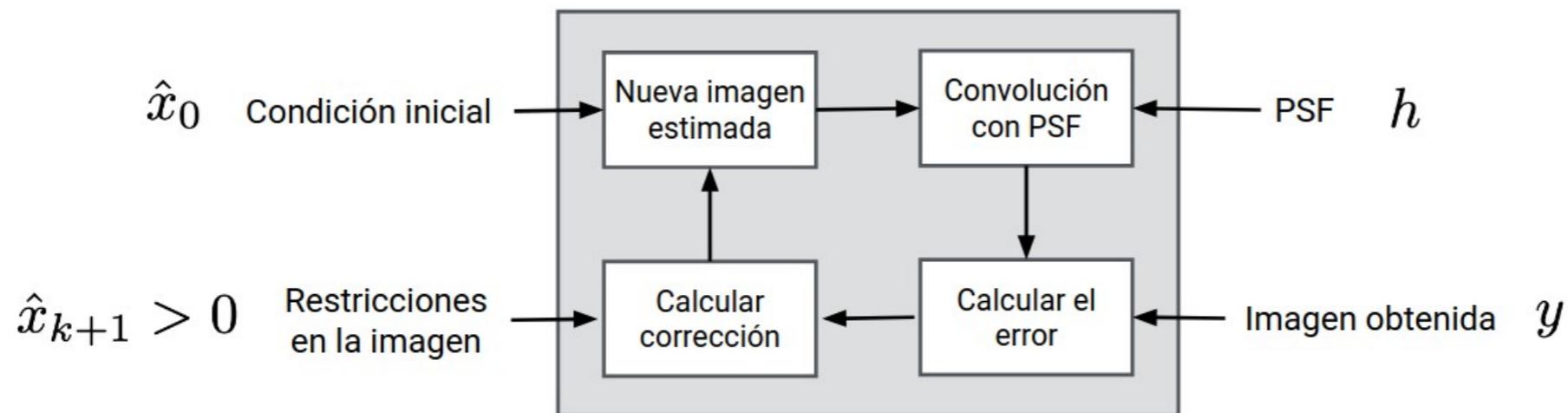
Filtrado inverso



Filtrado inverso



Deconvolución iterativa

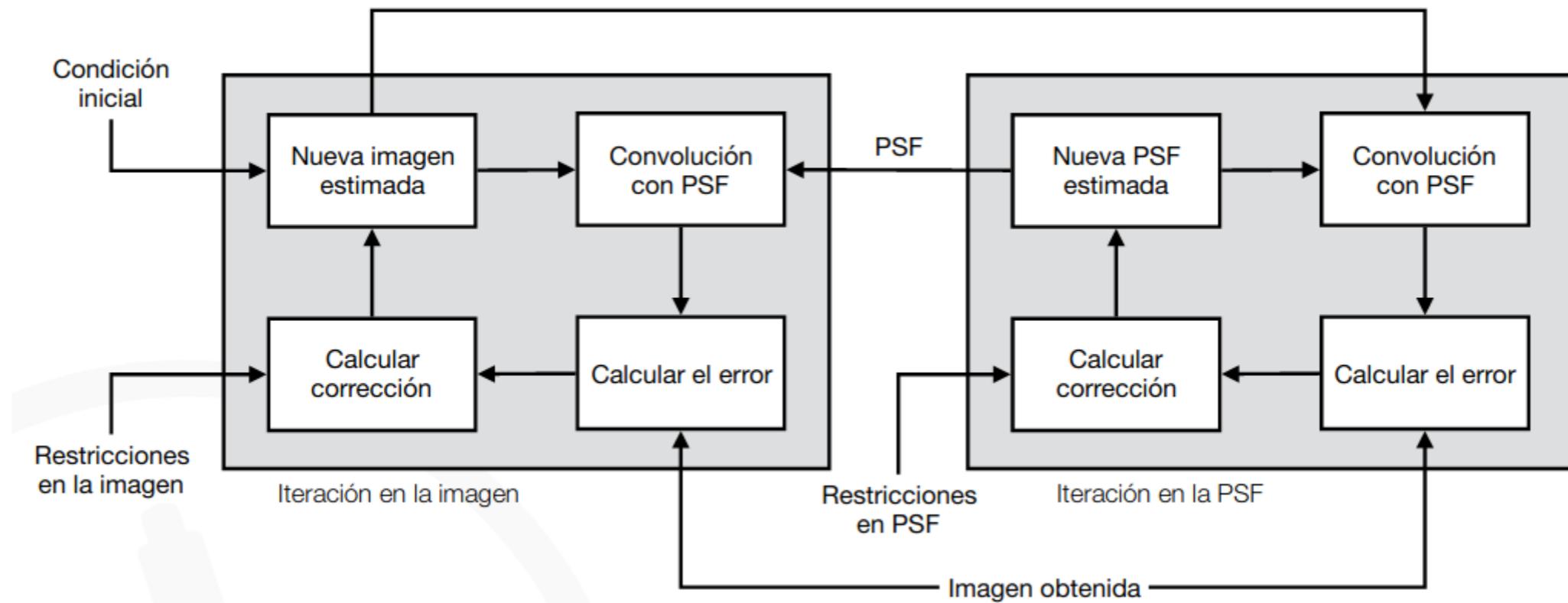


Alternativas de actualización iterativa:

$$\hat{x}_{k+1} = \hat{x}_k + (y - \hat{x}_k * h) \quad (\text{aditiva})$$

$$\hat{x}_{k+1} = \hat{x}_k \frac{y}{\hat{x}_k * h} \quad (\text{multiplicativa})$$

Blind deconvolution



IPOL

- Blur, Denoising, Turbulence, Inpainting

- **Blind Image Deblurring using the l0 Gradient Prior**
2019-03-12 · Jérémy Anger, Gabriele Facciolo, Mauricio Delbracio
- **Estimating an Image's Blur Kernel Using Natural Image Statistics, and Deblurring it: An Analysis of the Goldstein-Fattal Method**
2018-09-26 · Jérémy Anger, Gabriele Facciolo, Mauricio Delbracio
- **Implementation of Local Fourier Burst Accumulation for Video Deblurring**
2017-03-19 · Jérémy Anger, Enric Meinhardt-Llopis
- **Computing an Exact Gaussian Scale-Space**
2016-02-02 · Ives Rey Otero, Mauricio Delbracio
- **A Survey of Gaussian Convolution Algorithms**
2013-12-17 · Pascal Getreuer
- **Total Variation Deconvolution using Split Bregman**
2012-07-30 · Pascal Getreuer
- **Cross-comparison of the Performance of Sequential Summed Area Table and Box Filter Algorithms with respect to C/C++ Compilers**
PREPRINT · Ali Ozturk, Ibrahim Cayiroglu
- **Implementation of a Denoising Algorithm Based on High-Order Singular Value Decomposition of Tensors**
2019-06-13 · Fabien Feschet
- **An Analysis and Implementation of the FFDNet Image Denoising Method**
2019-01-06 · Matias Tassano, Julie Delon, Thomas Veit
- **EPLL: An Image Denoising Method Using a Gaussian Mixture Model Learned on a Large Set of Patches**
2018-12-23 · Samuel Hurault, Thibaud Ehret, Pablo Arias
- **Video Denoising with Optical Flow Estimation**
2018-07-23 · Antoni Buades, Jose-Luis Lisani
- **Multi-Scale DCT Denoising**
2017-10-29 · Nicola Pierazzo, Jean-Michel Morel, Gabriele Facciolo
- **A Fast Approximation of the Bilateral Filter using the Discrete Fourier Transform**
2017-05-24 · Pravin Nair, Anmol Popli, Kunal N. Chaudhury
- **Data Adaptive Dual Domain Denoising: a Method to Boost State of the Art Denoising Algorithms**
2017-05-24 · Nicola Pierazzo, Gabriele Facciolo
- **Analysis and Extension of the PCA Method, Estimating a Noise Curve from a Single Image**
2016-12-29 · Miguel Colom, Antoni Buades
- **PARIGI: a Patch-based Approach to Remove Impulse-Gaussian Noise from Images**
2016-06-29 · Julie Delon, Agnès Desolneux, Thierry Guillemot
- **On the Implementation of Collaborative TV Regularization: Application to Cartoon+Texture Decomposition**
2016-04-20 · Joan Duran, Michael Moeller, Catalina Sbert, Daniel Cremers
- **Small Neural Networks can Denoise Image Textures Well: a Useful Complement to BM3D**
2016-01-19 · Yi-Qing Wang