Filtrado de Imágenes

Dr. Álvaro Pardo
Departamento de Ingeniería Eléctrica
Universidad Católica del Uruguay

Indice

Modelo

Modelo de ruido aditivo:

$$I(\mathbf{p}) = Z(\mathbf{p}) + N(\mathbf{p})$$

Si se tienen n muestras *independientes* de la misma imagen

$$\{I_i(\mathbf{p}) = Z(\mathbf{p}) + N_i(\mathbf{p})\}_{i=1,\ldots,n}$$

Reducción de ruido:

$$\frac{1}{k}\sum_{i=1}^{n} I_i(\mathbf{p}) = Z(\mathbf{p}) + \frac{1}{n}\sum_{i=1}^{n} N_i(\mathbf{p})$$

La varianza del ruido decae como $\sigma_N/\sqrt{n}$. Si la imagen varía lentamente:

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} I(\mathbf{p}_i) &= \frac{1}{n}\sum_{i=1}^{n} Z(\mathbf{p}_i) + \frac{1}{n}\sum_{i=1}^{n} N(\mathbf{p}_i) \\ &\approx Z(\mathbf{p}) + \frac{1}{n}\sum_{i=1}^{n} N(\mathbf{p}_i)\end{aligned}$$

Filtros Lineales: Filtros de Promedio

$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$
$\mathbf{p}_4$	$\mathbf{p}_5 = \mathbf{p}$	$\mathbf{p}_6$
$\mathbf{p}_7$	$\mathbf{p}_8$	$\mathbf{p}_9$

$$\hat{I}(\mathbf{p}) = \frac{I(\mathbf{p}_2) + I(\mathbf{p}_4) + I(\mathbf{p}_6) + I(\mathbf{p}_8)}{4}$$

$$\hat{I}(\mathbf{p}) = \frac{I(\mathbf{p}_2) + I(\mathbf{p}_4) + I(\mathbf{p}_5) + I(\mathbf{p}_6) + I(\mathbf{p}_8)}{5}$$

$$\hat{I}(\mathbf{p}) = \frac{\sum_{i=1}^{9} I(\mathbf{p_i})}{9}$$

Filtros Lineales: Respuesta al impulso

$$\frac{1}{4}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline\end{array} \qquad \frac{1}{5}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline\end{array} \qquad \frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

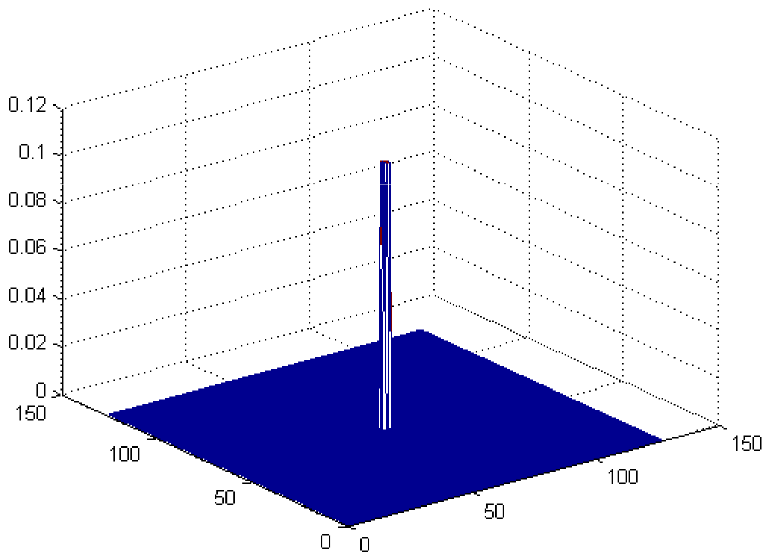
$$\begin{aligned}\hat{I}(n,m) &= \sum_p \sum_q I(p,q)h(n-p,m-q) \\ &= \sum_p \sum_q h(p,q)I(n-p,m-q) \\ &= h(n,m) * I(n,m) = I(n,m) * h(n,m)\end{aligned}$$

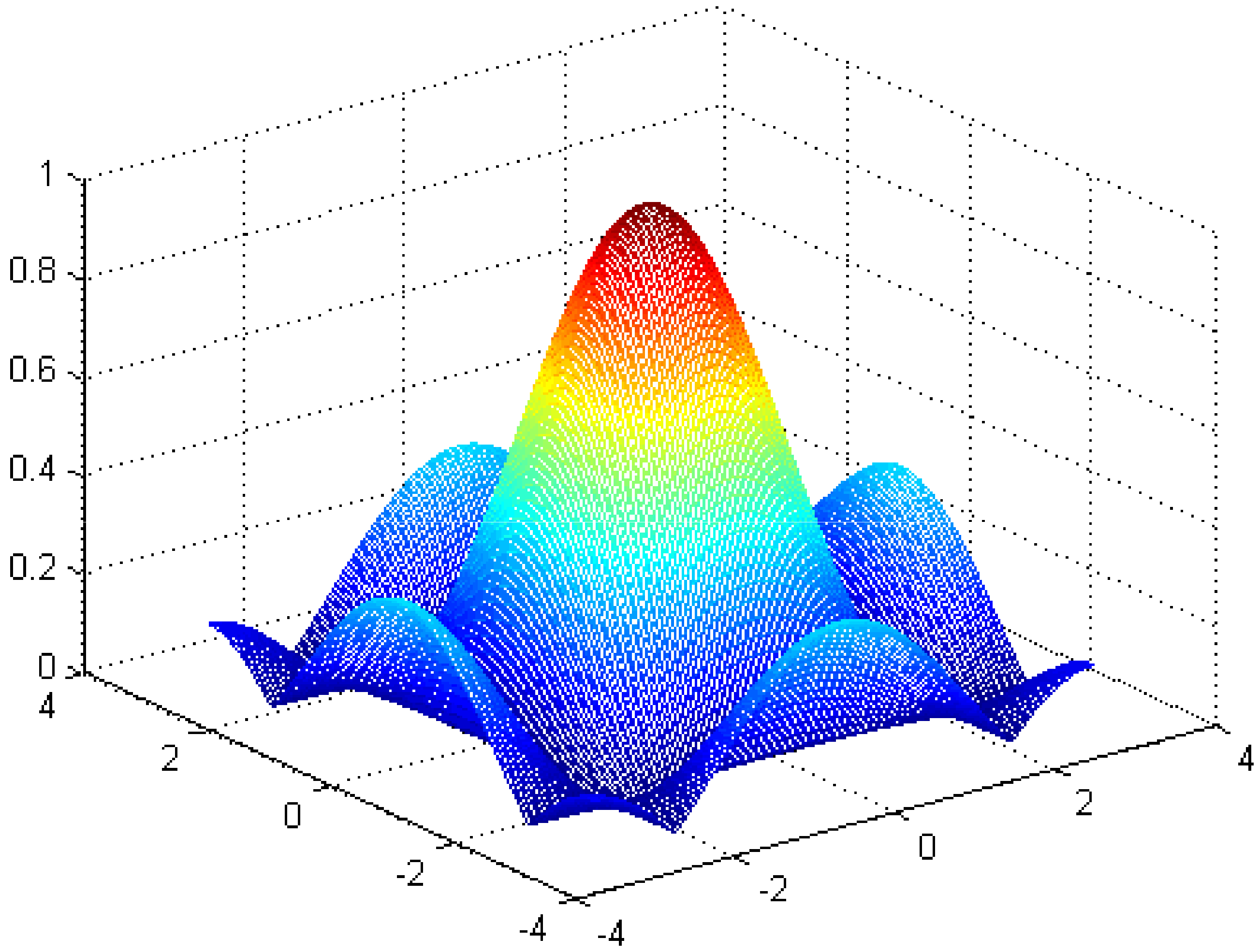
Simplificación:

$$\hat{I}(\mathbf{p}) = \sum_{\mathbf{q}} I(\mathbf{q})h(\mathbf{p}-\mathbf{q}).$$

Filtros Lineales: Box Filter

$$h(\mathbf{p}-\mathbf{q}) = h(|\mathbf{p}-\mathbf{q}|_0) = \begin{cases} \frac{1}{9} & |\mathbf{p}-\mathbf{q}|_0 \leq 1 \\ 0 & \text{else} \end{cases}$$

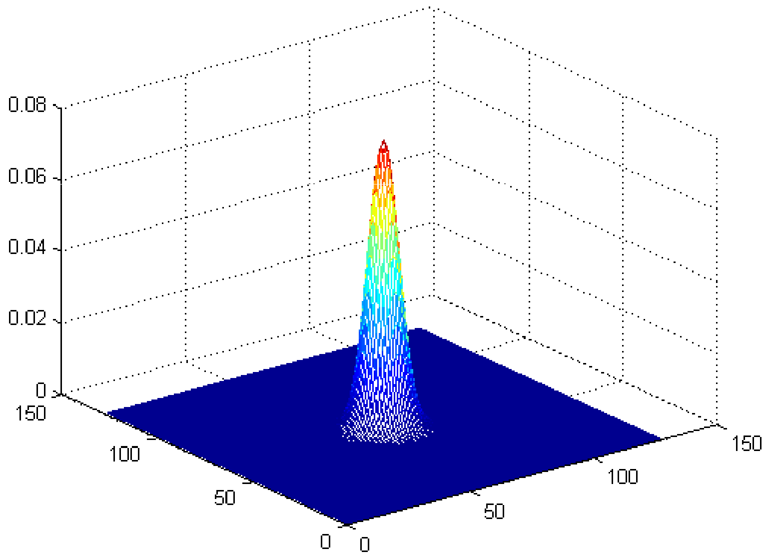


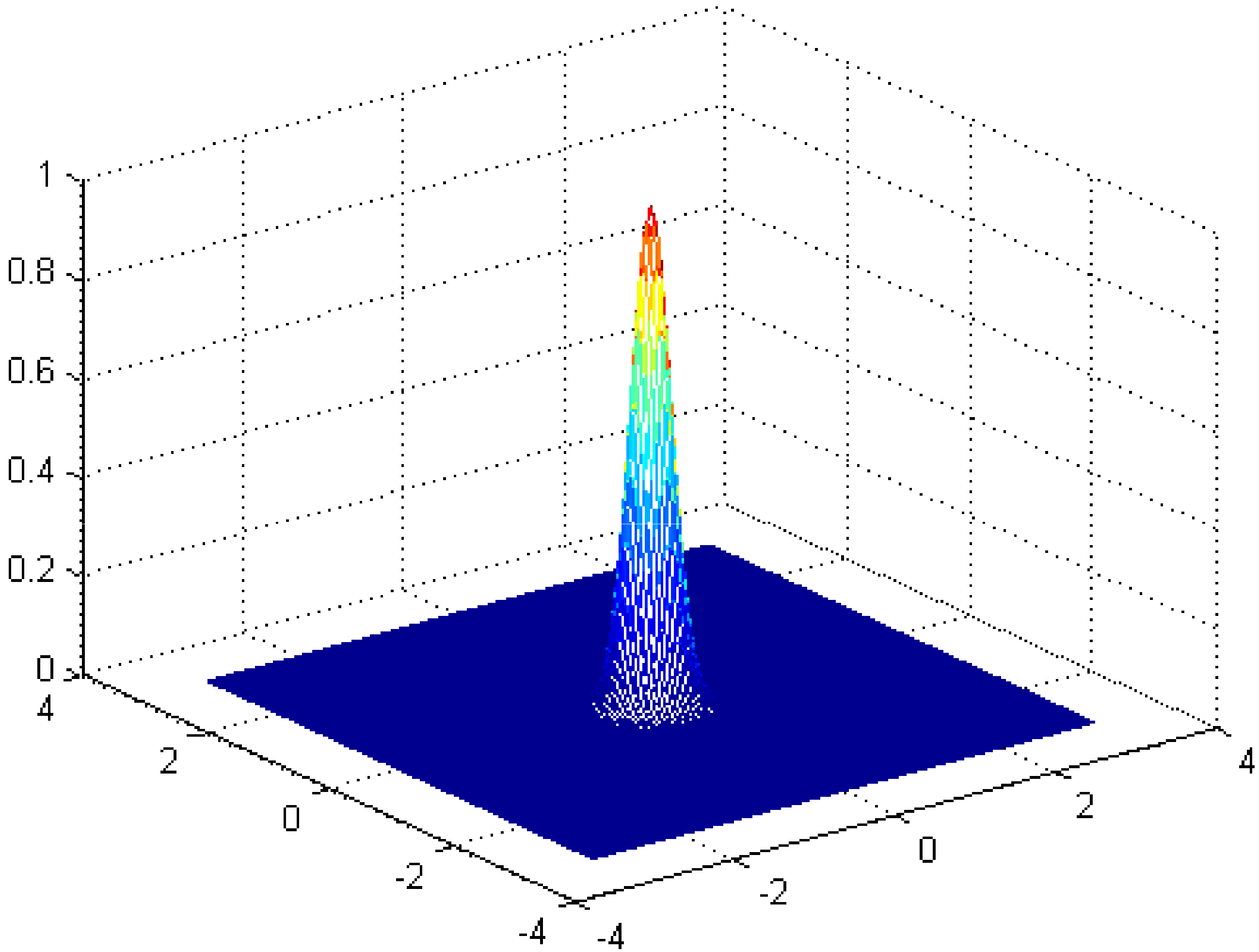


Espectro del filtro Box.

Filtros Lineales: Filtro Gaussiano

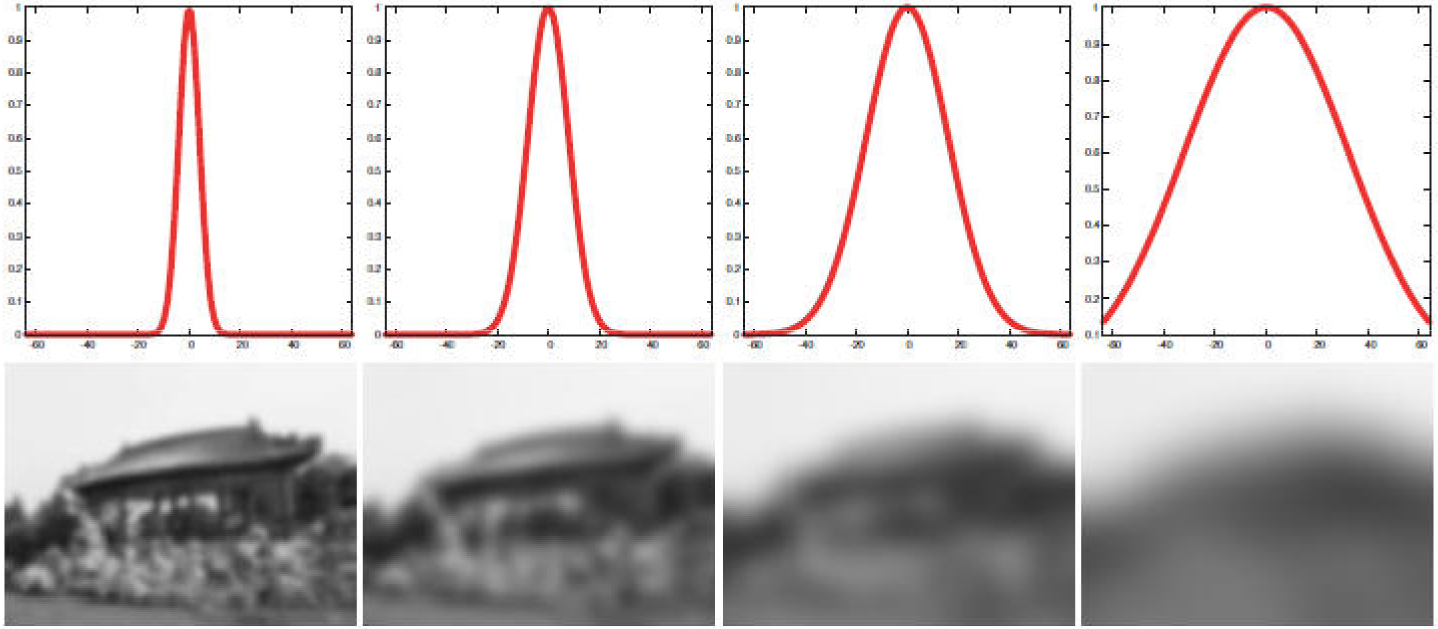
$$h(\mathbf{p}-\mathbf{q}) = h(|\mathbf{p}-\mathbf{q}|_2) = \frac{1}{C} e^{-|\mathbf{p}-\mathbf{q}|_2^2/\sigma^2}$$





Espectro del filtro Gaussiano.

Filtro Gaussiano: Resultados



$\sigma = 4$ $\sigma = 8$ $\sigma = 16$ $\sigma = 32$

Extraída de: Bilateral Filtering: Theory and Applications, Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Fredo Durand.

Filtros de máscaras variables

- La idea es encontrar entorno a cada pixel ***p*** a restaurar el conjunto de pixeles “cercanos” en nivel de gris (rango).
- Los primeros algoritmos de este tipo utilizaban un conjunto predefinido de máscaras.
- La máscara a utilizar para el promedio se selecciona mediante un criterio de homogeneidad.
- Es un filtro lineal pero espacialmente variante.

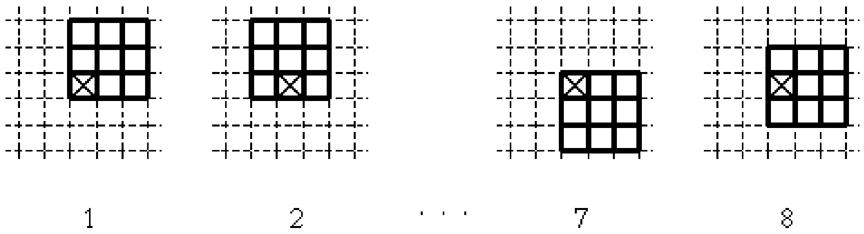


Figure 4.11 8 possible rotated 3×3 masks.

$$\sigma^2 = \frac{1}{n}\left(\sum_{(i,j)\in R}\left(g(i,j) - \frac{1}{n}\sum_{(i,j)\in R} g(i,j)\right)^2\right) \qquad (4.32)$$

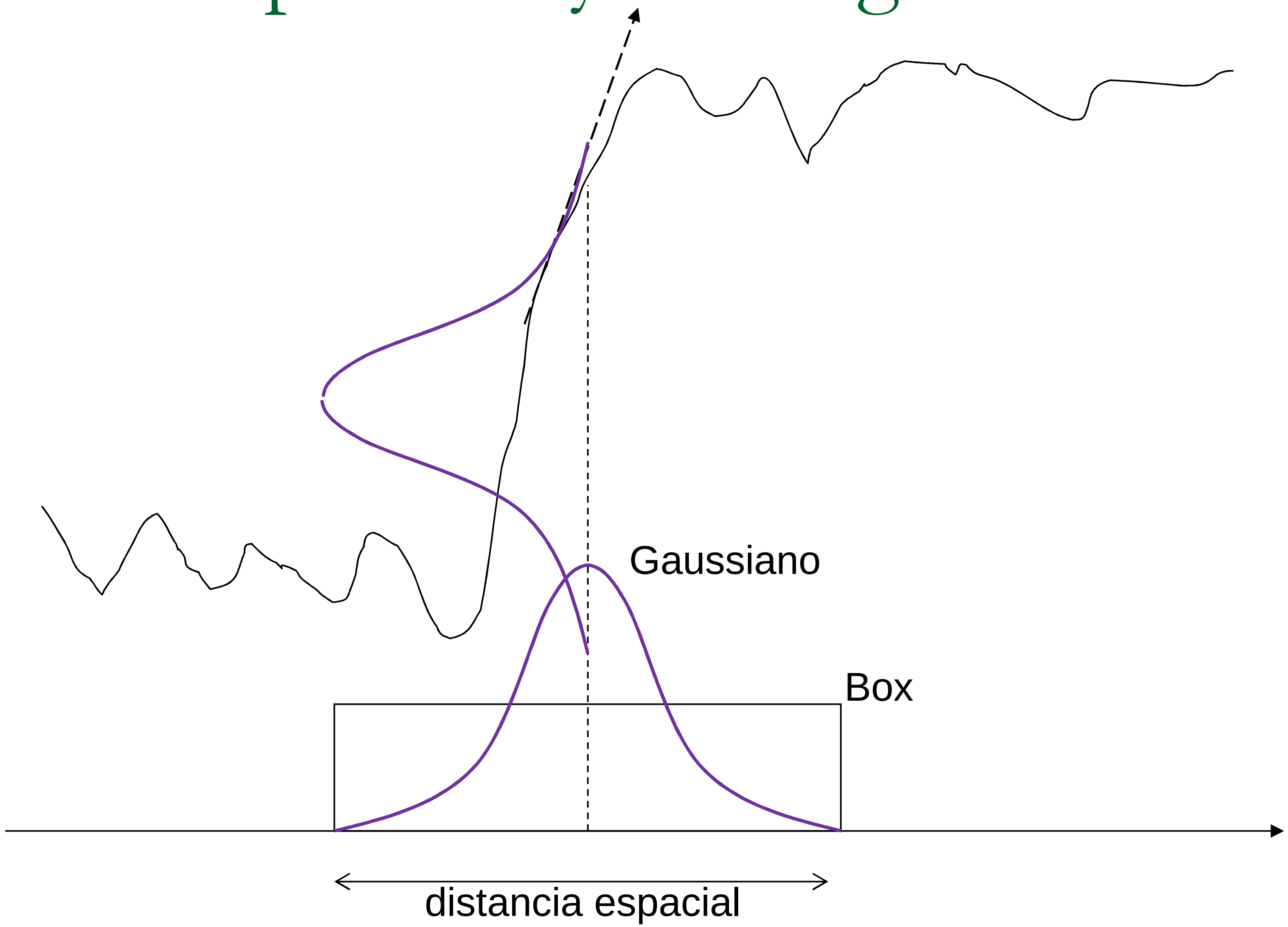
Figuras extraídas de M. Sonka, V. Hlavac, R. Boyle, “Image Processing, Analysis, and Machine Vision”, Thomson 1999.

Filtro de Promedio Ponderado

$$\hat{I}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\mathcal{N}} w(\mathbf{p},\mathbf{q}) I(\mathbf{q})}{\sum_{\mathbf{q}\in\mathcal{N}} w(\mathbf{p},\mathbf{q})}.$$

- Los filtros anteriores se pueden generalizar promediando los píxeles dentro de una ventana *N* con pesos *w(.).*
- Los pesos *w(.)* pueden ser fijos e independientes de la posición (dependientes únicamente de p-q) en cuyo caso tenemos filtros lineales e invariantes.
- Si los *w(.)* se calculan utilizando características locales de la imagen y/o utilizando las relaciones entre *I(p)* e *I(q)* entonces resultan filtros no lineales.

Filtros espaciales y en rango



Filtros Espaciales y en Rango

Yarolavsky:

$$\hat{I}(\mathbf{p}) = \frac{1}{C(\mathbf{p})} \int_{\mathcal{N}} I(\mathbf{q}) e^{\frac{-|I(\mathbf{p})-I(\mathbf{q})|_2^2}{\sigma_r^2}} d\mathbf{q}$$

$\mathcal{N}$ es un vecindario entorno al pixel $\mathbf{p}$, σ es el ancho del filtro y C una constante de normalización.

SUSAN y Filtro Bilateral:

Ambos consideran pesos espaciales y en nivel de intensidad.

$$\hat{I}(\mathbf{p}) = \frac{1}{C(\mathbf{p})} \int_{\mathcal{N}} I(\mathbf{q}) e^{\frac{-|I(\mathbf{p})-I(\mathbf{q})|_2^2}{\sigma_r^2}} e^{\frac{-|\mathbf{p}-\mathbf{q}|_2^2}{\sigma_s^2}} d\mathbf{q}$$

Similitud entre los pixeles

Filtro Bilateral: Ecuaciones

$$\hat{I}(p) = \frac{1}{W_p} \sum_{q \in \mathcal{N}} w(p,q) I(q)$$

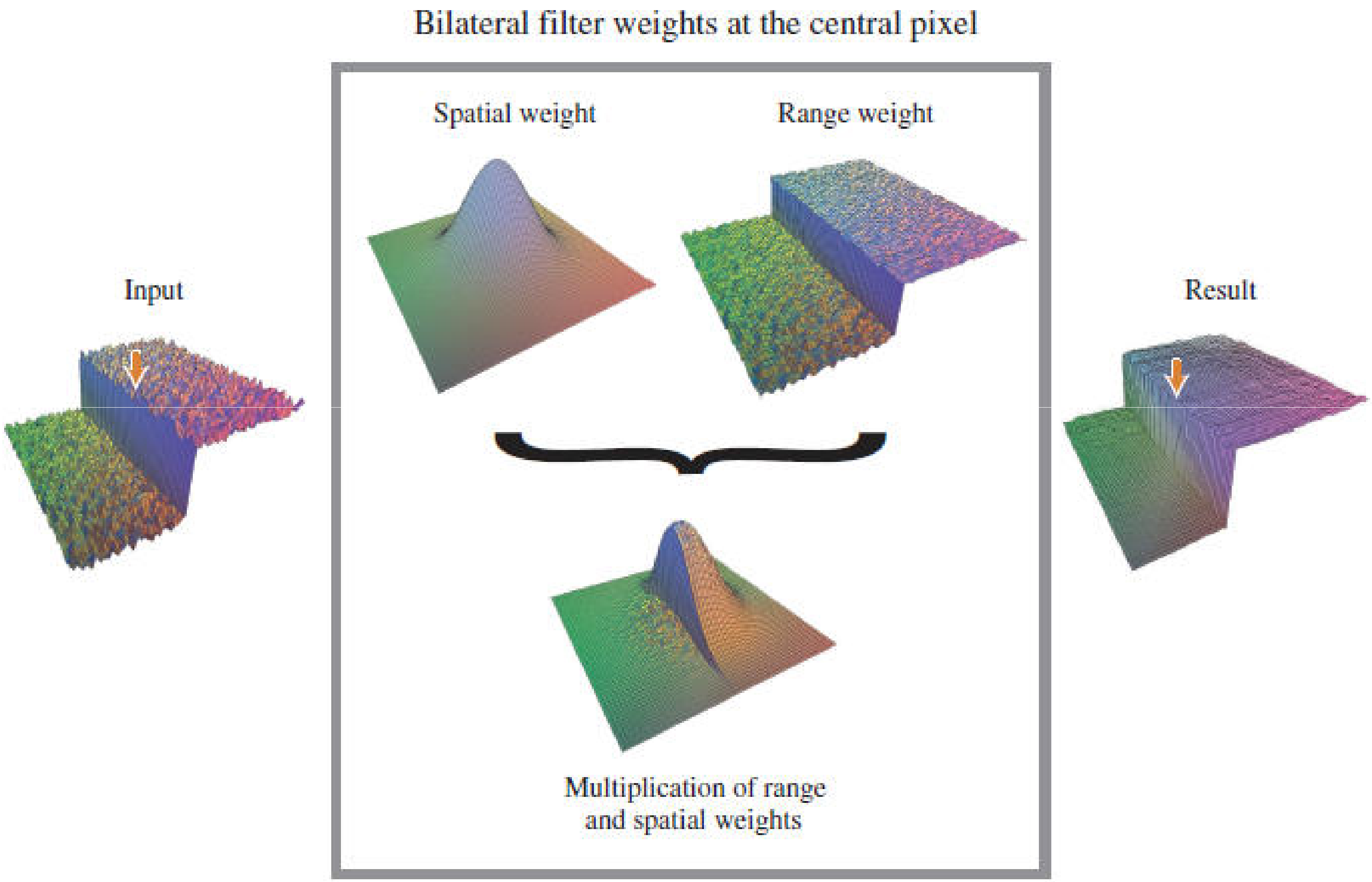
$$w(p,q) = G_s(|p-q|) G_r(|I(p) - I(q)|)$$

$$W_p = \sum_{q \in \mathcal{N}} w(p,q)$$

Los pesos combinan la diferencia espacial, |p-q|, y la diferencia en rango, |I(p)-I(q)|.

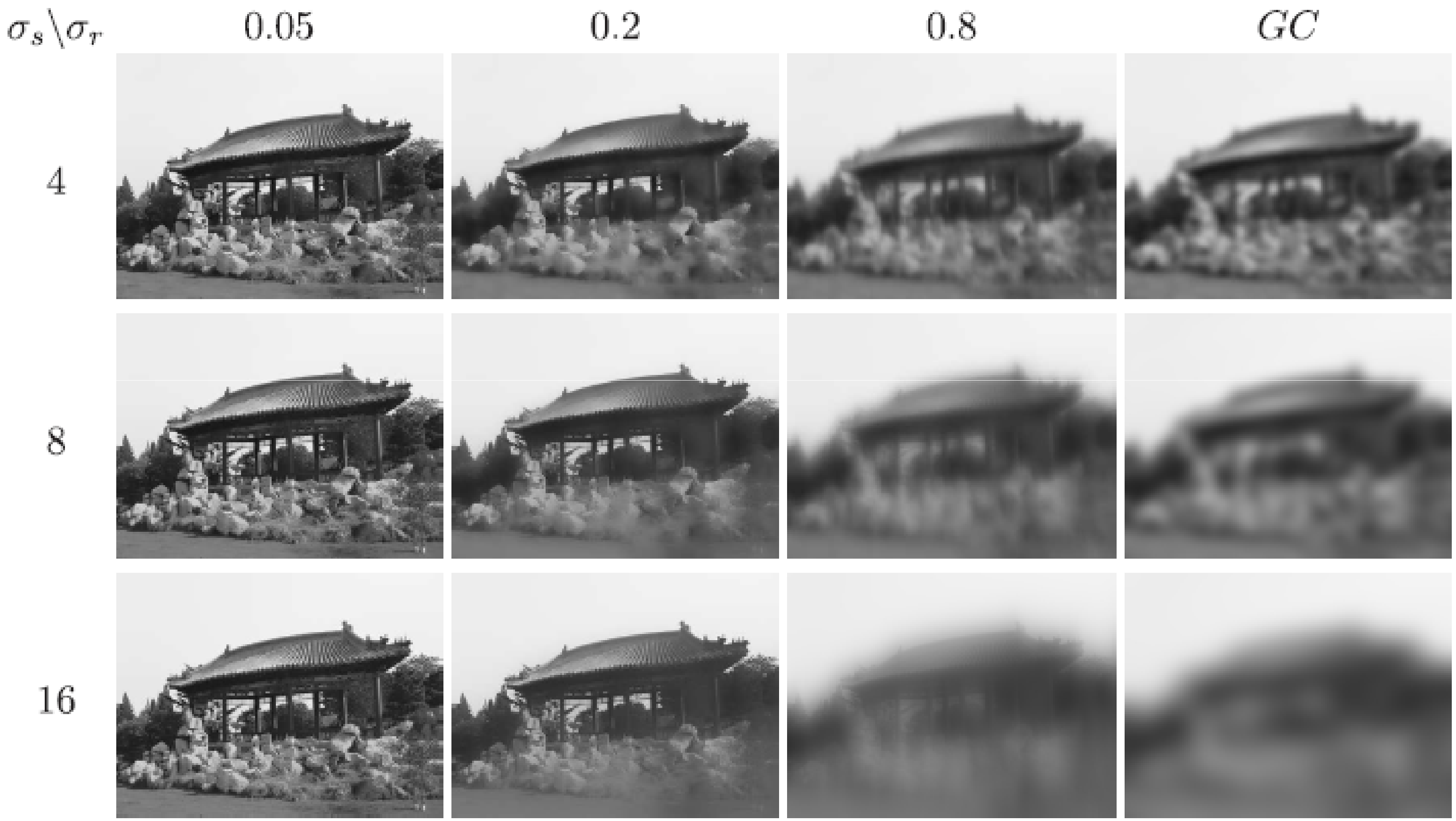
Es un filtro no lineal y variante a desplazamientos (no es LTI).

Filtro Bilateral: Idea



Extraída de: Bilateral Filtering: Theory and Applications, Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Fredo Durand.

Filtro Bilateral: Funcionamiento



Extraída de: Bilateral Filtering: Theory and Applications, Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Fredo Durand.

Filtro Bilateral: Residuo

(a) Input (b) Bilateral filter (c) Residual

- El residuo luego del filtrado contiene ruido y/o textura dependiendo de los parámetros usados.
- El residuo se puede utilizar en diversas aplicaciones:
  - Separación y procesamiento en componentes
  - Realce de detalles
  - HDR,
  - Etc.

Extraída de: Bilateral Filtering: Theory and Applications, Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Fredo Durand.

Filtrado: Formulación General

- Los pesos del filtro bilateral depende de *p,q,I(p),I(q)*:

$$w(p,q,I(p),I(q)) = G_s(|p-q|)G_r(|I(p)-I(q)|)$$

- entonces:

$$\hat{I}(p) = \frac{1}{W_p}\sum_{q\in\mathcal{N}} w(p,q,I(p),I(q))I(q)$$

- En general, se puede formular el proceso de estimación:

$$I^*(p) = \arg\min_{I(p)} \sum_q w(p,q,I(p),I(q))[I(q)-I(p)]^2$$

Non local Means

- La similitud, *w(.)*, se mide en función de la similitud de vecindades entorno a cada pixel.
- Además la búsqueda de píxeles similares se hace “idealmente” a lo largo de toda la imagen.
  - Esto es en teoría porque en la práctica muchas veces se trabaja sobre una región menor.

$$\hat{I}(\mathbf{p}) = \frac{1}{C(\mathbf{p})}\int_{\Omega} I(\mathbf{q}) e^{\frac{-||w(\mathbf{p})-w(\mathbf{q})||_a^2}{\sigma^2}} d\mathbf{q}$$

$$||w(\mathbf{p})-w(\mathbf{q})||_a^2 = \int G_a(\mathbf{z})|I(\mathbf{p}+\mathbf{z})-I(\mathbf{q}+\mathbf{z})|^2 d\mathbf{z}$$

Usando los pesos $w(\mathbf{p},\mathbf{q}) = e^{\frac{-||w(\mathbf{p})-w(\mathbf{q})||_a^2}{\sigma^2}}$

$$\hat{I}(\mathbf{p}) = \frac{1}{C(\mathbf{p})}\int_{\Omega} I(\mathbf{q}) w(\mathbf{p},\mathbf{q}) d\mathbf{q} \qquad \hat{I}(\mathbf{p}) = \frac{1}{C(\mathbf{p})}\sum_{\mathbf{q}} I(\mathbf{q}) w(\mathbf{p},\mathbf{q})$$

Non Local Means: Idea

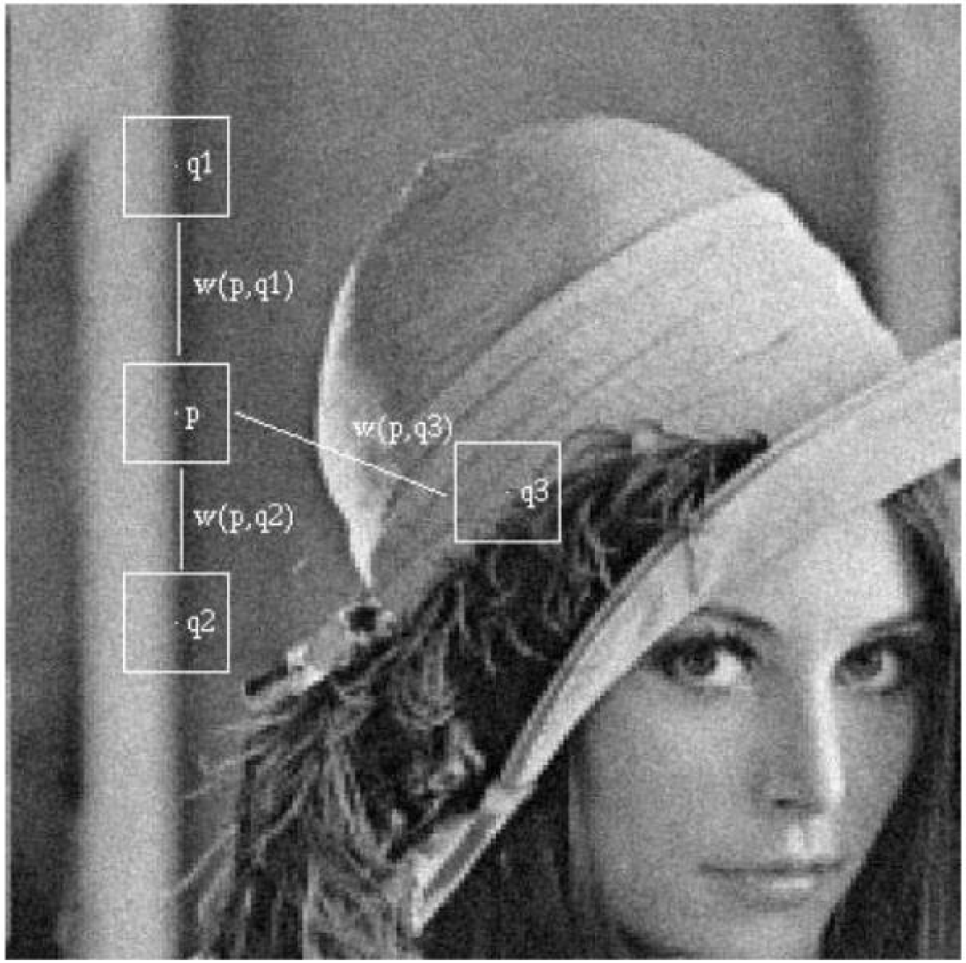


FIG. 6. *q1 and q2 have a large weight because their similarity windows are similar to that of p. On the other side the weight $w(p, q3)$ is much smaller because the intensity grey values in the similarity windows are very different.*

Figura extraída de Buades, Morel, Coll, SIAM, 2005.

Filtros Basados en “Patches”

- NLMeans es un ejemplo de un algoritmo de eliminación de ruido que utiliza “patches” para determinar las similitudes entre pixeles.
- La idea es que estos permiten capturar la geometría local y la textura dentro de la imagen.

$$\hat{I}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{q})} w(\mathbf{p},\mathbf{q}) I(\mathbf{q})}{\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{q})} w(\mathbf{p},\mathbf{q})}$$

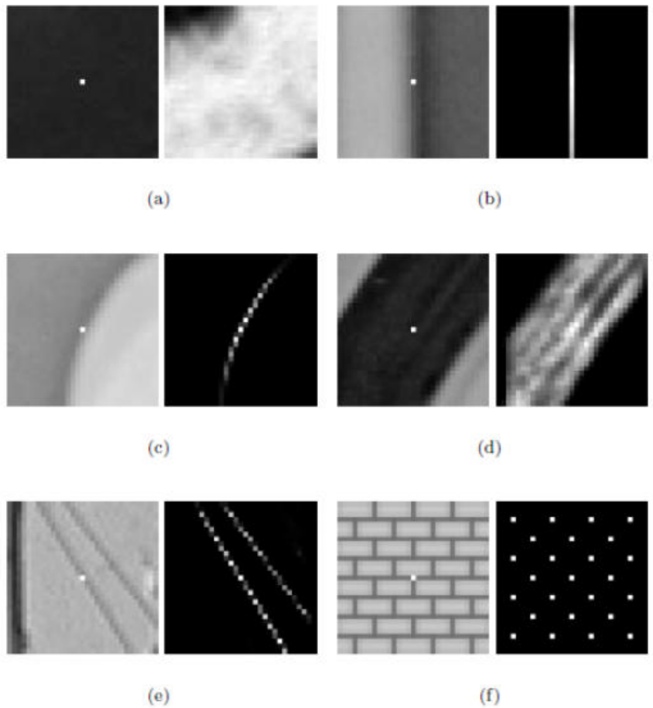


FIG. 13. *On the right-hand side of each pair, we display the weight distribution used to estimate the central pixel of the left image by the NL-means algorithm.* (a) *In flat zones, the weights are distributed as a convolution filter (as a Gaussian convolution).* (b) *In straight edges, the weights are distributed in the direction of the level line (as the mean curvature motion).* (c) *On curved edges, the weights favor pixels belonging to the same contour or level line, which is a strong improvement with respect to the mean curvature motion.* (d) *In a flat neighborhood, the weights are distributed in a grey-level neighborhood (as with a neighborhood filter). In the cases of* (e) *and* (f)*, the weights are distributed across the more similar configurations, even though they are far away from the observed pixel. This shows a behavior similar to a nonlocal neighborhood filter or to an ideal Wiener filter.*

Figura extraída de Buades, Morel, Coll, SIAM, 2005.

Propiedades de NLM

Let $I(x_i)$ be the original noisy image value at pixel x_i. Its denoised version using NLM can be obtained as:

$$\hat{I}(x_i) = \frac{\sum_j w(x_i, x_j) I(x_j)}{\sum_j w(x_i, x_j)} = \frac{\sum_j w_{ij} I(x_j)}{\sum_j w_{ij}}$$

where w_{ij} are computed using a gaussian kernel,

$$w_{ij} = \exp(-||N_i - N_j||^2/\sigma^2)$$

and N_i, N_j are image patches of size $(2K+1)\times(2K+1)$ centered at pixels i and j.

Propiedades de NLM

Let the matrix W be the one with entries w_{ij}, and D the diagonal matrix with entries $d_{ii} = \sum_j w_{ij}$. If we consider $\mathbf{I}$ as the vectorial version of the image, scanned in lexicographic order, NLM can be rewritten as:

$$\hat{\mathbf{I}} = D^{-1} W \mathbf{I}$$

- The matrix

  $$\boldsymbol{L = D^{-1}W}$$

  defines an operator which filters the image ***I,*** to obtain its denoised version.

- The properties of the matrix ***L*** determine the denoising result.

1. $L = D^{-1}W$ defines an operator which filters the image.

2. L is an image adapted lowpass filter since the operator depends on the image itself.

3. Residual after denoising:

$$\mathbf{r} = \mathbf{x} - \hat{\mathbf{x}} = \mathbf{x} - D^{-1}W\mathbf{x} = (Id - D^{-1}W)\mathbf{x}.$$

4. $H = (Id - D^{-1}W)$ is the highpass operator associated to L.

5. If we consider pixels x_p as nodes of a graph connected with weights w_{pq}, H is the Normalized Laplacian of the graph.

NLM and Normalized Cuts

- If we write the residual after NLM we obtain:

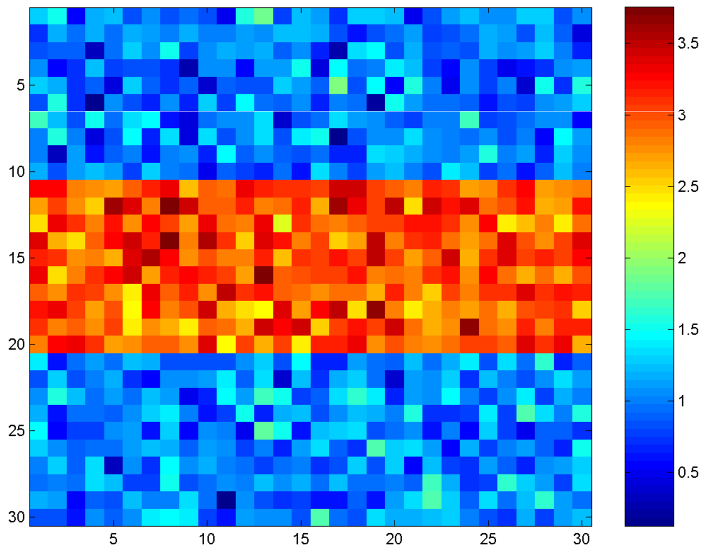
$$r(x) = I(x) - \hat{I}(x) = I - D^{-1}WI = (Id - D^{-1}W)I.$$

- **H = (Id - $\mathbf{D^{-1}}$W)** is the highpass operator associated with the lowpass operator defined by matrix **L**.
- Matrices **L** and **H** share the same eigenvectors
  - if ϕ_k is an eigenvector of **L** with eigenvalue λ_k then ϕ_k is an eigenvector of **H** with eigenvalue $1-\lambda_k$.
- If we view pixels $\mathbf{x_i}$ as nodes of a graph connected with weights $\mathbf{w_{ij}}$ the matrix **H** is the normalized Laplacian of the graph which is used in Normalized Cuts (NC).

NLM and Normalized Cuts

- Malik and Shi used the the Normalized Laplacian for segmentation.
- This shows the connection of NLM and NC.
- The eigenvectors of H are useful for segmentation. In fact, the second eigenvector can be used to split the image in two regions.
- It can be shown that the multiplicity of the eigenvalue 1 of H corresponds to the number of connected components in the graph.

Experiments with synthetic images



Noisy Image

1	1	1
1	1	1
1	1	1

1	1	1
1	1	1
3	3	3

1	1	1
3	3	3
3	3	3

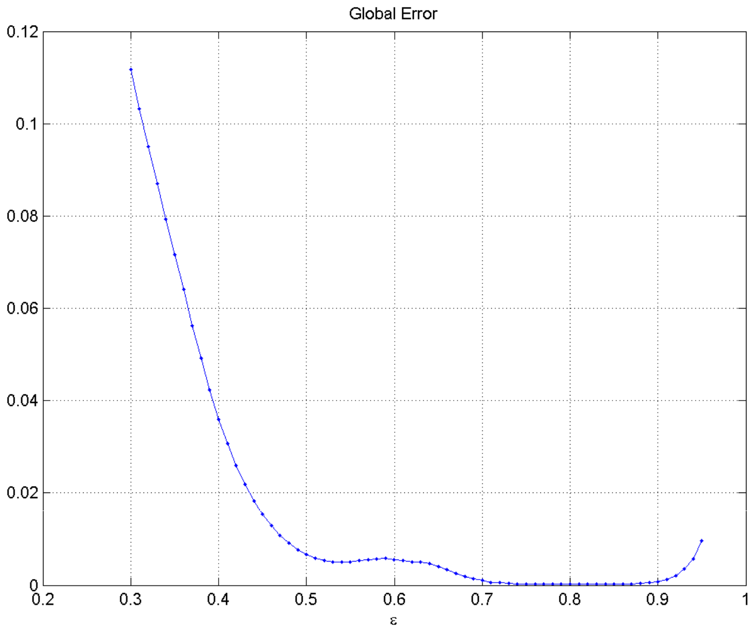
3	3	3
3	3	3
3	3	3

3	3	3
3	3	3
1	1	1

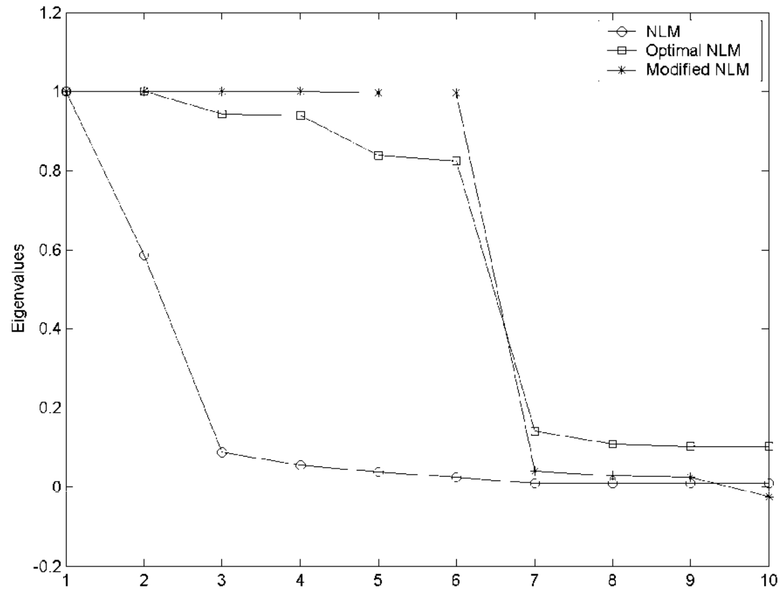
3	3	3
1	1	1
1	1	1

Patch configurations for 3x3 patches

Experiments with synthetic images



Total error



Eigenvalues

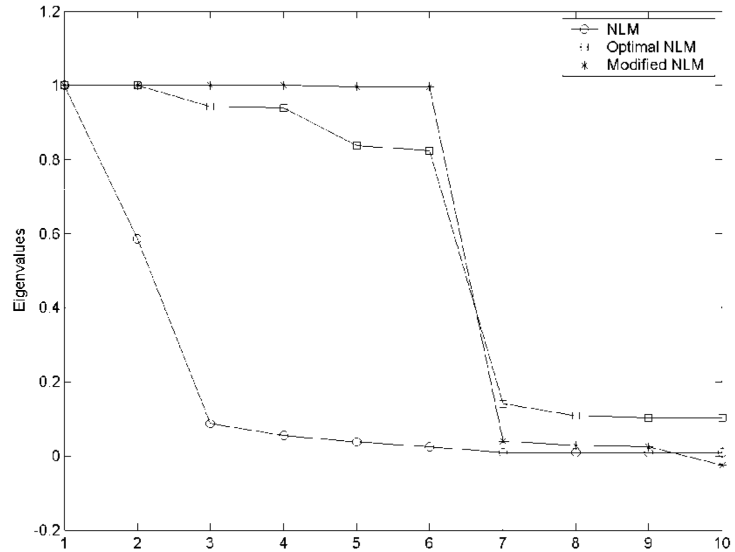
To study the performance of NLM with restrict the pixels used in the average considering only the ones with $w_{pq} > \varepsilon$.

Conclusions: The error depends on ε and therefore on σ.

NLM doesn't have a eigenvalue 1 of multiplicity 6.

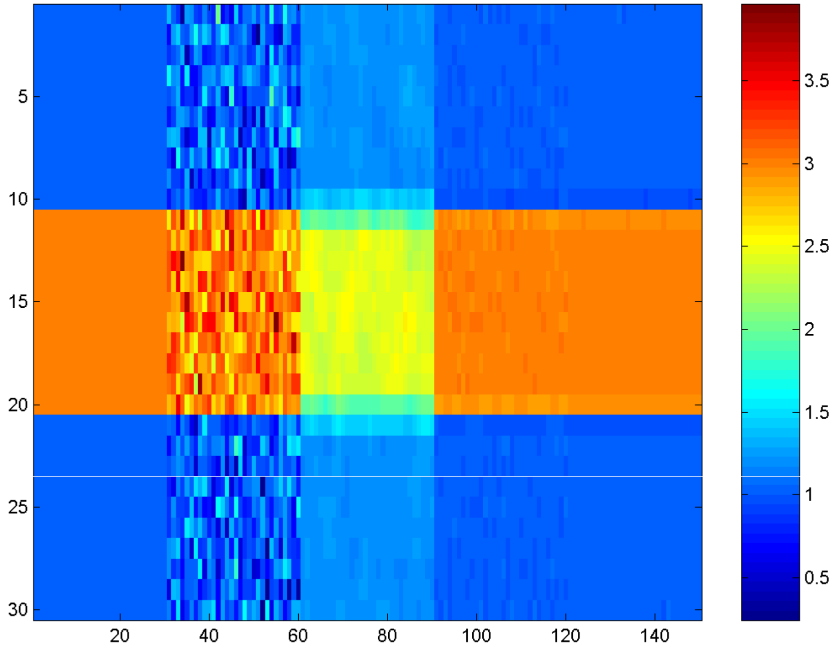
Proposal: Modified NLM

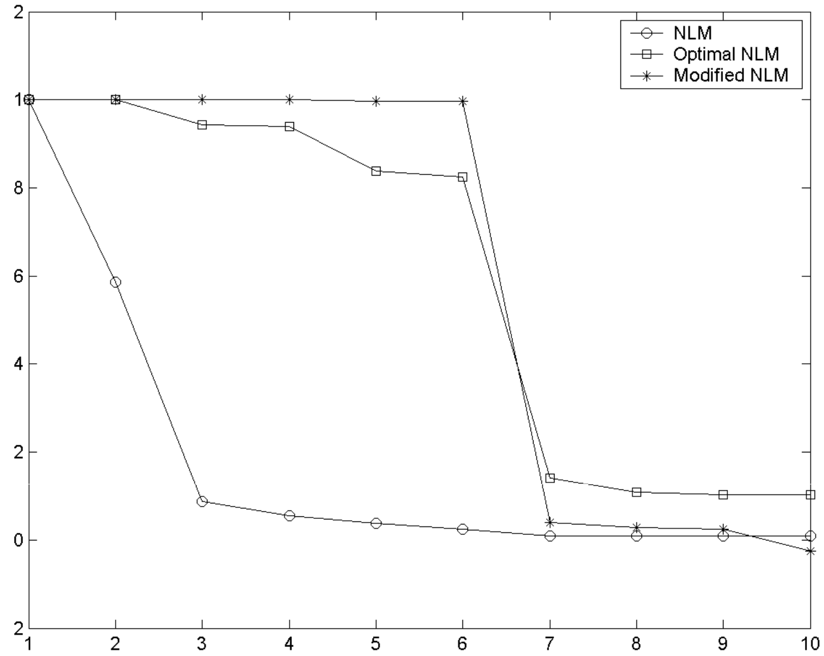
- **Observation**:
  - For the synthetic image we expect to have six different regions corresponding to the possible configurations, therefore we expect a eigenvalue 1 of multiplicity 6.



- **Goals:**
  - Find a way to automatically set the parameters of NLM to achieve minimum error.
  - Validate the results using the connection of NLM with the eigenvalues of H.

Results for synthethic images





Clean image, Noisy image, NLM (with parameters suggested by Baudes h = 12), optimal NLM (h=5), Modified NLM.

MNLM and Optimal NLM have six eigenvalues of value 0 corresponding to the six possible configurations.

MLNM performs slightly better that NLM.

MLNM preserves in a better way the edges.

Results for real images

Image	NLM			MNLM	
	$h*$	MSE	SSIM	MSE	SSIM
Barbara	3	30.11	0.913	30.10	0.922
Baboon	3	63.45	0.895	70.67	0.890
Couple	3	35.13	0.883	35.97	0.884
Einstein	3	35.03	0.859	36.01	0.858
Goldhill	3	33.78	0.868	34.61	0.869

The optimal ***h*** for NLM is lower than the one suggested by Buades et. at.

The results of NLM and MNLM are very similar using both MSE and SSIM.

MNLM: Estima automáticamente sus parámetros y alcanza resultados equivalentes al mejor NLM (obtenido manualmente conociendo la imagen original).

Referencias

- *Bilateral Filtering: Theory and Applications*, Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Frédo Durand, Foundations and Trends in Computer Graphics and Vision
  - http://people.csail.mit.edu/sparis/publi/2009/fntcgv/Paris_09_Bilateral_filtering.pdf
- *A Tour of Modern Image Processing*, PeymanMilanfar.
  - http://www.eecs.umich.edu/~fessler/course/556/r/milanfar--ato.pdf
- A. Buades, B. Coll, J.M Morel, "A review of image denoising algorithms, with a new one", Multiscale Modeling and Simulation (SIAM interdisciplinary journal), Vol 4(2), pp: 490-530, 2005.
  - http://www.mi.parisdescartes.fr/~buades/publicacions/2005_mms.pdf
- *Analysis of Non Local Image Denoising Methods*, A. Pardo, CIARP 2009.
- http://www.cs.tut.fi/~foi/GCF-BM3D/