

Operadores Espaciales no Lineales

Operadores espaciales No Lineales

- En PI un operador espacial no lineal realiza una operación no lineal en una vecindad de píxeles con el fin de:
 - Mejorar la imagen para su visualización por un ser humano
 - Facilitar la interpretación por una computadora
- Filtros de ordenamiento (Mediana, Moda)
- Filtro Kuwahara
- Filtro Bilateral



¿Qué es procesamiento de imágenes?

- El procesamiento de imágenes es el proceso computacional mediante el cual se transforma una o mas imágenes de entrada en una imagen de salida, **con el fin**:
 - Mejorar la imagen para su visualización por un ser humano
 - Facilitar la interpretación por una computadora
- Transformaciones comunes en PI
 - Operadores puntuales
 - Operadores espaciales
 - Operadores en frecuencia
 - Operadores espaciales no lineales
 - Operadores en escala



Filtros de ordenamiento

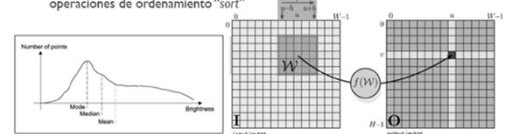
Operadores espaciales No Lineales

- En PI un operador espacial lineal (filtro lineal) computa una salida mediante una combinación lineal de los píxeles que rodean al centro del operador (filtro)
 - En algunos casos el procesamiento no necesita ser uniforme
- En PI un operador espacial no lineal realiza una operación no lineal en una vecindad de píxeles con el fin de:
 - Mejorar la imagen para su visualización por un ser humano
 - Facilitar la interpretación por una computadora



Operadores espaciales No Lineales

- Filtros de ordenamiento (Mediana, Moda):
 - La ventana W escogida se ordena de manera creciente y el valor del píxel resultante es el r -avo píxel
 - Ejemplo: $lre = 1; Sto = 6;$
 - Útil para la remoción de ruido puntual y localizado (salt and pepper noise)
 - Casos especiales del filtro de ordenamiento son la mediana y la moda.
 - La complejidad computacional del filtro es alta al utilizar operaciones de ordenamiento "sort"



Operadores espaciales No Lineales

Filtros de Mediana

- La ventana W escogida se ordena de manera creciente y el valor del pixel resultante es la mediana de la ventana
- Útil para la remoción de ruido puntual y localizado (salt and pepper noise) – Trata de conserva características (bordes)
- La complejidad computacional del filtro es alta al utilizar operaciones de ordenamiento "sort"

			1
			2
			4
			5
8	4	11	6
1	2	6	7
7	5	8	8
			8
			11

Imagen original con ruido salt and pepper alto

Filtrado Gaussiano

Filtrado mediana



Operadores espaciales No Lineales

Aspectos prácticos – Filtro de mediana

- Ver Ex3Med
- `dst = cv.medianBlur(src, ksize[, dst])`
- Blurs an image using the median filter.
- The function smooths an image using the median filter with the $ksize \times ksize$ aperture. Each channel of a multi-channel image is processed independently. In-place operation is supported.

» `[medianBlur()]` https://docs.opencv.org/3.4.3/d4/d86/group__imgproc__filter.html#ga564869aa33e58769b4469101aac458f9

Operadores espaciales No Lineales

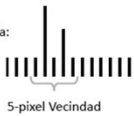
Imagen original con ruido salt and pepper alto

Filtrado mediana

Ejemplos Filtros de Mediana

Mediana([1 7 1 5 1]) = 1
Media([1 7 1 5 1]) = 2.8

Entrada:



Salida:



Ruido de sal y pimienta se remueve

Entrada:



Salida:



Borde simples (monotónicos) se mantiene

Operadores espaciales No Lineales

Filtros de Moda

- La ventana W escogida se ordena de manera creciente y el valor del pixel resultante es la moda de la ventana
- Útil para la remoción de ruido puntual y localizado (salt and pepper noise) – Trata de conserva características (bordes)
- La complejidad computacional del filtro es alta al utilizar operaciones de ordenamiento "sort" – En algunos casos no se puede calcular la moda [1 2 3 4]

Imagen original con ruido salt and pepper muy alto

Filtrado mediana

Filtrado moda



Operadores espaciales No Lineales

Ejemplos Filtros de Mediana

3x3

5x5

7x7

Filtrado Gaussiano

Imagen original con ruido salt and pepper



Filtrado mediana



Filtros Kuwahara

Operadores espaciales No Lineales

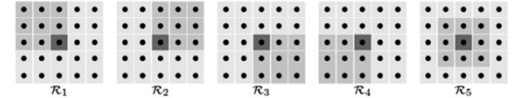
- En PI un operador espacial no lineal realiza una operación no lineal en una vecindad de píxeles con el fin de:
 - Mejorar la imagen para su visualización por un ser humano
 - Facilitar la interpretación por una computadora
- Filtros de ordenamiento (Mediana, Moda)
- Filtro Kuwahara
- Filtro Bilateral



Operadores espaciales No Lineales

- Filtro Kuwahara - variante Tomita-Tsuji 1977 → La ventana W escogida se divide en subregiones de las cuales se cuantifica la media y la varianza. La región mas homogénea (menor varianza) será el valor del pixel de salida mediante su media

Se adiciona una subregión en el centro y las subregiones son impar



$$\mu_k(u, v) = \frac{1}{|\mathcal{R}_k|} \cdot \sum_{(i,j) \in \mathcal{R}_k} I(u+i, v+j)$$

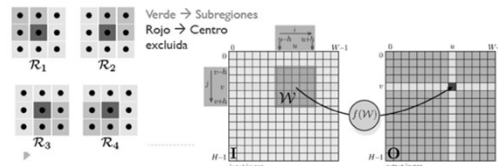
$$\sigma_k^2(u, v) = \frac{1}{|\mathcal{R}_k|} \cdot \sum_{(i,j) \in \mathcal{R}_k} (I(u+i, v+j) - \mu_k(u, v))^2$$

$$I'(u, v) \leftarrow \mu_{k'}(u, v), \quad \text{with } k' = \underset{k=1, \dots, K}{\operatorname{argmin}} \sigma_k^2(u, v).$$

Problema de optimización que define la salida

Operadores espaciales No Lineales

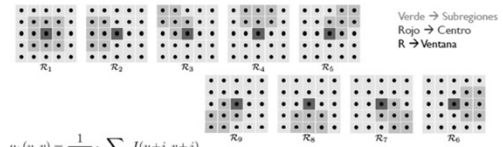
- Filtro Kuwahara 1976 → La ventana W escogida se divide en subregiones de las cuales se cuantifica la media y la varianza. La región mas homogénea (menor varianza) será el valor del pixel de salida mediante su media
- Útil para eliminar ruidos si eliminar características de la imágenes (bordes)



Operadores espaciales No Lineales

- Filtro Kuwahara - variante Nagao-Matsuyama 1979 → La ventana W escogida se divide en subregiones de las cuales se cuantifica la media y la varianza. La región mas homogénea (menor varianza) será el valor del pixel de salida mediante su media

Las subregiones no incluyen el pixel central y tiene formas irregulares para tener en cuenta los posibles bordes (diagonales)



$$\mu_k(u, v) = \frac{1}{|\mathcal{R}_k|} \cdot \sum_{(i,j) \in \mathcal{R}_k} I(u+i, v+j)$$

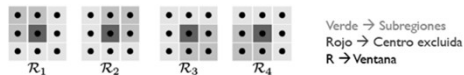
$$\sigma_k^2(u, v) = \frac{1}{|\mathcal{R}_k|} \cdot \sum_{(i,j) \in \mathcal{R}_k} (I(u+i, v+j) - \mu_k(u, v))^2$$

$$I'(u, v) \leftarrow \mu_{k'}(u, v), \quad \text{with } k' = \underset{k=1, \dots, K}{\operatorname{argmin}} \sigma_k^2(u, v).$$

Problema de optimización que define la salida

Operadores espaciales No Lineales

- Filtro Kuwahara 1976 → La ventana W escogida se divide en subregiones de las cuales se cuantifica la media y la varianza. La región mas homogénea (menor varianza) será el valor del pixel de salida mediante su media



$$\mu_k(u, v) = \frac{1}{|\mathcal{R}_k|} \cdot \sum_{(i,j) \in \mathcal{R}_k} I(u+i, v+j)$$

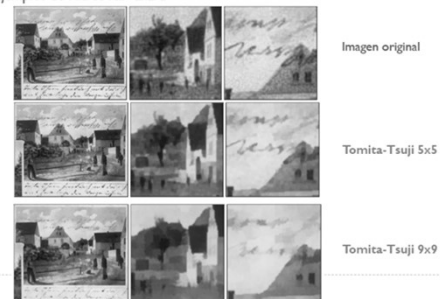
$$\sigma_k^2(u, v) = \frac{1}{|\mathcal{R}_k|} \cdot \sum_{(i,j) \in \mathcal{R}_k} (I(u+i, v+j) - \mu_k(u, v))^2$$

$$I'(u, v) \leftarrow \mu_{k'}(u, v), \quad \text{with } k' = \underset{k=1, \dots, K}{\operatorname{argmin}} \sigma_k^2(u, v).$$

Problema de optimización que define la salida

Operadores espaciales No Lineales

- Ejemplos de Filtros Kuwahara



Operadores espaciales No Lineales

Ejemplos de Filtros Kuwahara



Imagen original

Tomita-Tsuji 5x5

Nagao-Matsuyama 5x5

Conserva mejor los bordes en diagonal

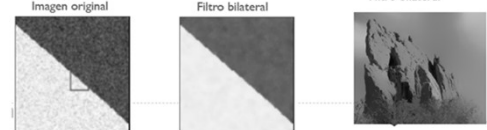
Operadores espaciales No Lineales

► Filtro Bilateral Tomasi-Manduchi 1997: Es un filtro que combina el filtrado espacial de distancia y el filtrado espacial de rango con el fin de eliminar ruido y conservar características en imágenes (bordes)...Matriz de filtrado es adaptativa

► Filtro bilateral = Filtro de distancia + Filtro de rango

► F. Distancia → El valor es proporcional a la distancia del pixel

► F. Rango → El valor es proporcional a la diferencia de intensidad



Filtros bilateral

Operadores espaciales No Lineales

► Filtro Bilateral Tomasi-Manduchi 1997:

► F. Distancia → El valor es proporcional a la distancia del pixel...lo usual

$$I'_d(u,v) \leftarrow \sum_{i,j=-\infty}^{\infty} I(i,j) \cdot H_d(i-u, j-v),$$

► F. Rango → El valor es proporcional a la diferencia de intensidad...lo nuevo

$$I'_r(u,v) \leftarrow \sum_{i,j=-\infty}^{\infty} I(i,j) \cdot H_r(I(i,j) - I(u,v))$$

► Filtro bilateral = Filtro de distancia + Filtro de rango

$$I''(u,v) \leftarrow \frac{1}{W_{u,v}} \cdot \sum_{i,j=-\infty}^{\infty} \sum_{k,l=-\infty}^{\infty} I(i,j) \cdot \underbrace{H_d(i-u, j-v) \cdot H_r(I(i,j) - I(u,v))}_{w_{i,j}},$$

$$\text{Filtro bilateral gaussiano} \leftarrow \frac{\exp\left(-\frac{d^2}{2\sigma_d^2}\right)}{2\pi\sigma_d^2} \cdot \exp\left(-\frac{(I(u+m, v+n) - I(u,v))^2}{2\sigma_r^2}\right),$$



Operadores espaciales No Lineales

► En PI un operador espacial no lineal realiza una operación no lineal en una vecindad de píxeles con el fin de:

- Mejorar la imagen para su visualización por un ser humano
- Facilitar la interpretación por una computadora



- Filtros de ordenamiento (Mediana, Moda)
- Filtro Kuwahara
- Filtro Bilateral



Operadores espaciales No Lineales

► Filtro Bilateral Tomasi-Manduchi 1997:

► Filtro bilateral = Filtro de distancia + Filtro de rango

► Implementado con Gaussianas

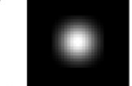
$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_d}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q$$

Factor de normalización

Pesos de distancia

Pesos de rango

Se controlan dos parámetros además del tamaño de la ventana W_p , el sigma de distancia "d" y el sigma de rango "r".



Operadores espaciales No Lineales - Filtro Bilateral Tomasi-Manduchi 1997:

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_p - I_q\|) I_q$$

Salida

Entrada

reproduced from [Durand 02]

Operadores espaciales No Lineales

► Ejemplos de Filtro bilateral

Imagen original

Bilateral d=2 y r=50

Bilateral d=2 y r=100

Operadores espaciales No Lineales

► Filtro Bilateral Tomasi-Manduchi 1997:

► Filtro Gaussiano vs Filtro Bilateral

Kernel/matriz de filtrado igual en cada caso

Se controlan dos parámetros además del tamaño de la ventana W, el sigma de distancia "d" y el sigma de rango "r".

Operadores espaciales No Lineales

► Ejemplos de Filtro bilateral

Imagen original

Filtro Bilateral 7x7

Filtro mediana 5x5

Operadores espaciales No Lineales

► Filtro Bilateral Tomasi-Manduchi 1997:

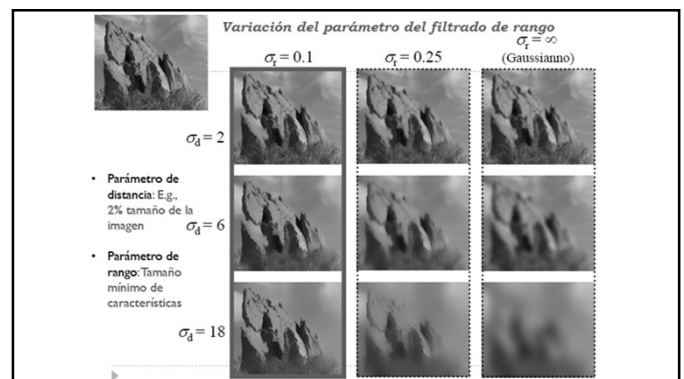
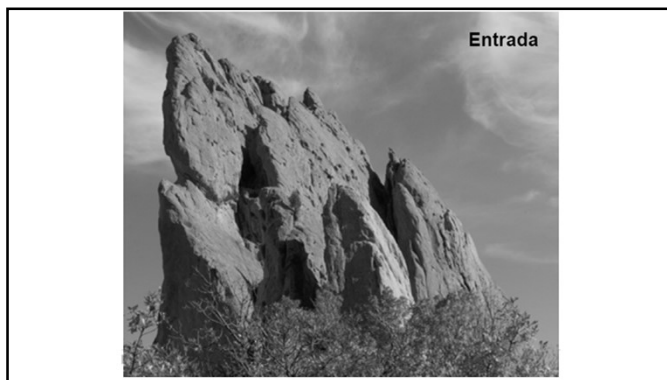
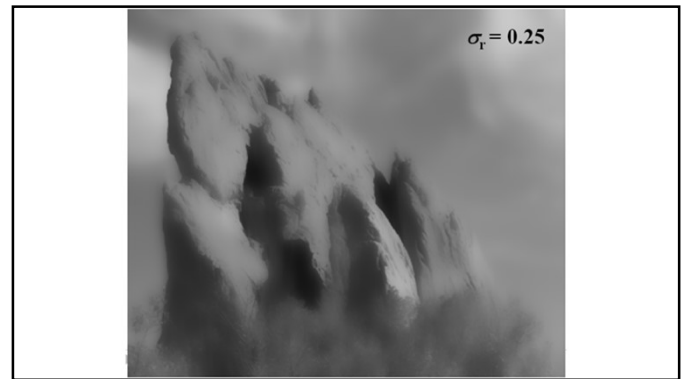
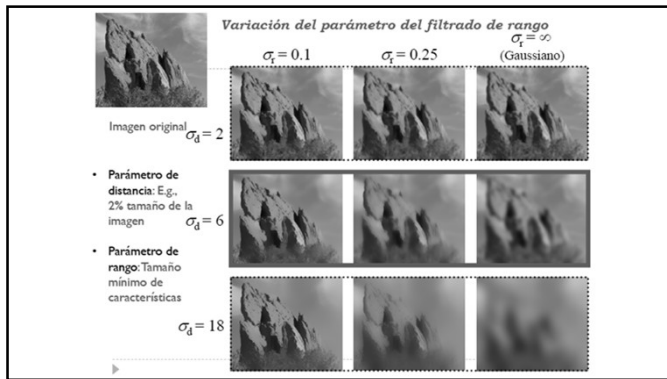
► Filtro Gaussiano vs Filtro Bilateral

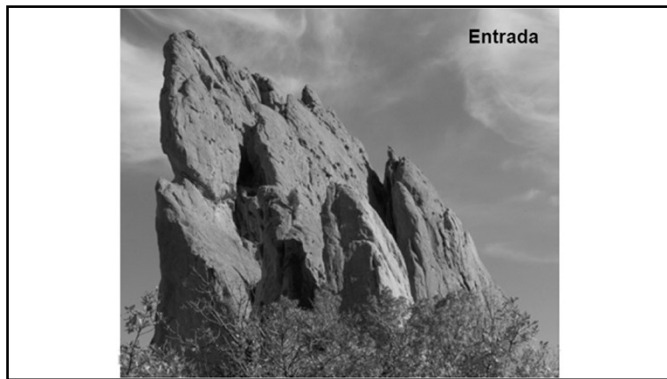
Kernel/matriz de filtrado adaptativa debido al filtro de rango

Operadores espaciales No Lineales

Filtro Bilateral 7x7

Filtro mediana 5x5





Operadores espaciales No Lineales

- Filtro Bilateral Tomasi-Manduchi 1997:
 - Filtro bilateral = Filtro de distancia + Filtro de rango
 - Implementado con Gaussianas

$$BF[I]_p = \frac{1}{\sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|)} I_q$$

Factor de normalización Pesos de distancia Pesos de rango

Se controlan dos parámetros además del tamaño de la ventana W , el sigma de distancia " d " y el sigma de rango " r ".



Operadores espaciales No Lineales

- Aspectos prácticos – Filtro bilateral
 - Ver Ex4Bil
- `dst = cv.bilateralFilter(src, d, sigmaColor, sigmaSpace[, dst[, borderType]])`
 - Applies the bilateral filter to an image. `BilateralFilter` can reduce unwanted noise very well while keeping edges fairly sharp. However, it is very slow compared to most filters.
 - Sigma values: For simplicity, you can set the 2 sigma values to be the same. If they are small (< 10), the filter will not have much effect, whereas if they are large (> 150), they will have a very strong effect, making the image look "cartoonish".
 - Filter size: Large filters ($d > 5$) are very slow, so it is recommended to use $d=5$ for real-time applications, and perhaps $d=9$ for offline applications that need heavy noise filtering.

`[bilateralFilter()]`
https://docs.opencv.org/3.4.3/d4/d86/group__imgproc__filter.html#gx9d7064d478c95d60003c839430737ed