جَامعة
المَنارة
MANARA UNIVERSITY

Computer vision

المحاضرة الثامنة

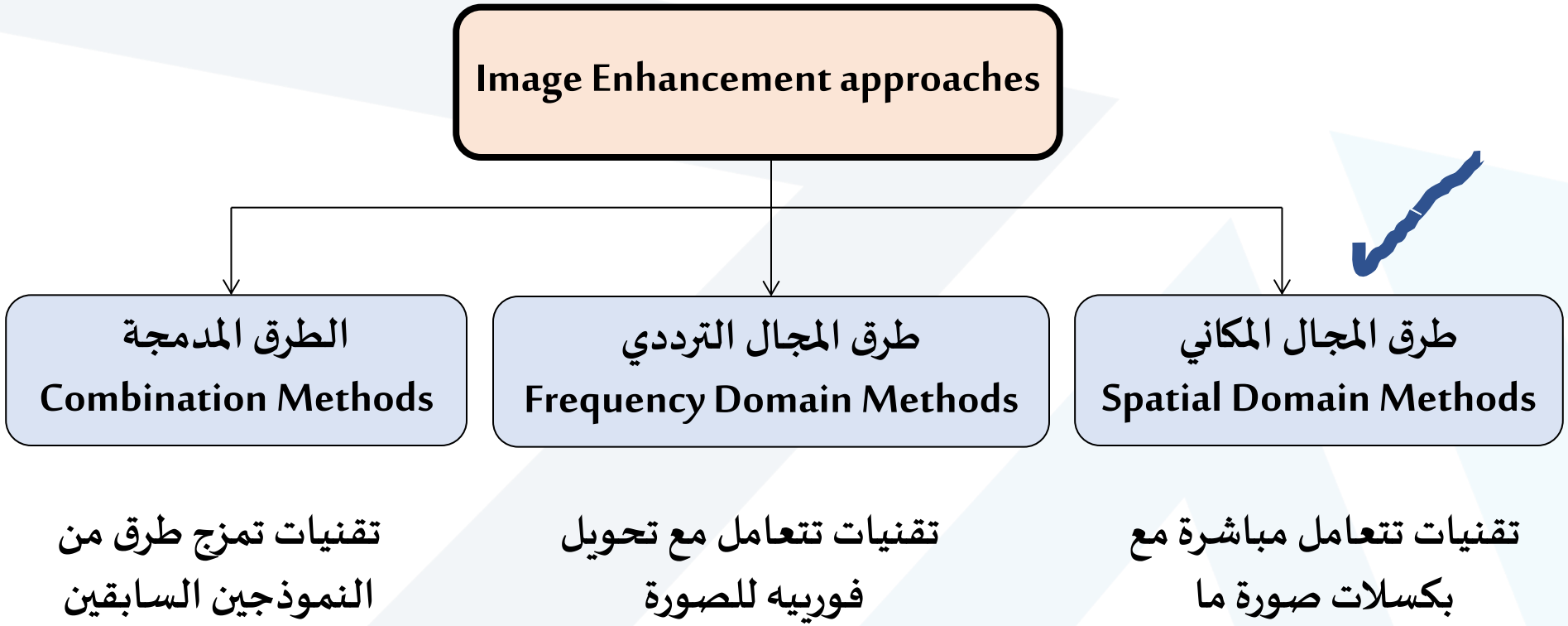
Image Enhancement

Spatial Domain Methods

العمليات على مستوى القناع أو جيران البكسل باستخدام المرشحات المكانية

د. عيسى الغنام د. إياد حاتم

طرق تحسين الصور



تحسين الصورة في المجال المكاني

مفهوم ترشيح الصورة بالاعتماد على النوافذ المحلية جيران البكسل التي تستخدم عادة في إزالة التشويش أو للكشف عن الحواف واظهارها.

Spatial Domain Methods

- **مستوى القناع جيران البكسل**
  Mask Level
- **مستوى البكسل**
  Pixel Level
- **العمليات على هيستوغرام الصورة**

العمليات على البكسلات

باستخدام هيستوغرام الصورة

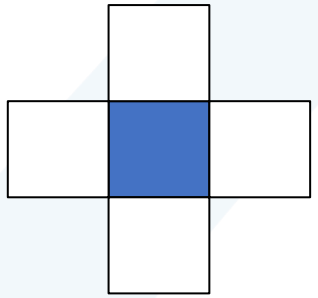
معالجة الرسم البياني (الهيستوغرام)

مساواة Histogram Processing

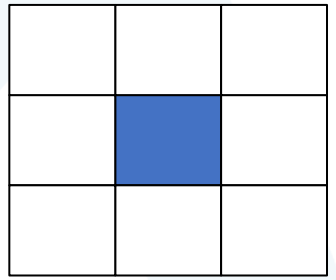
الهيستوغرام

$N_0(x,y)$

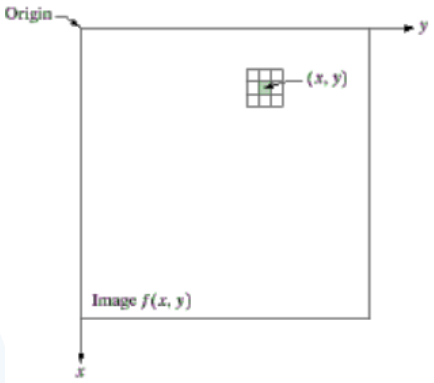
تحسين الصورة باستخدام المرشحات المكانية



$N_4(x,y)$

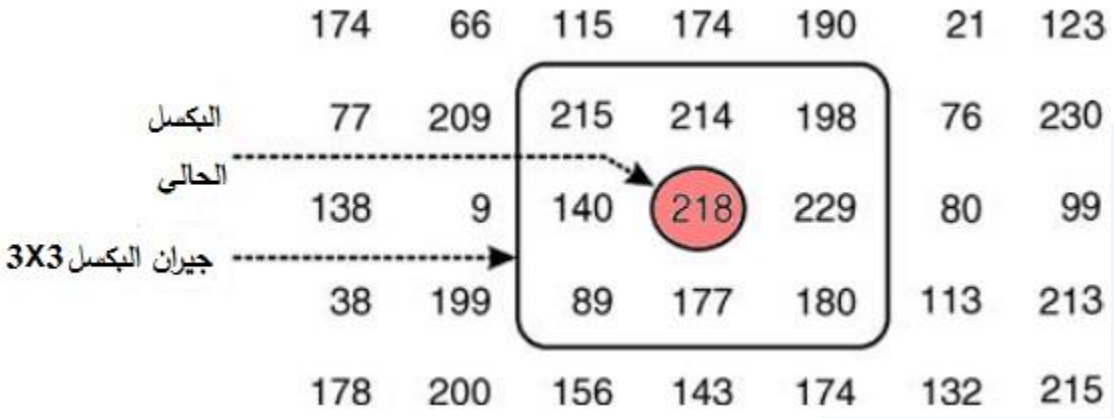


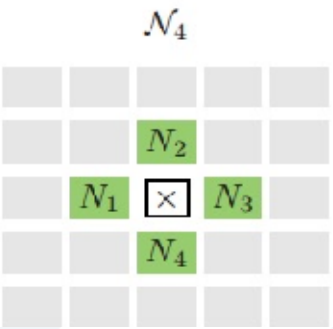
$N_8(x,y)$

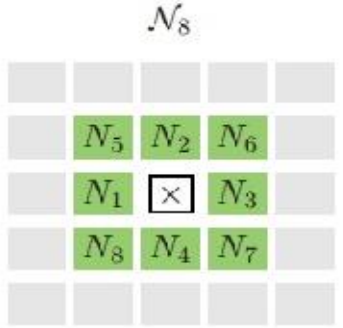


تعاريف

- جيران البكسل: البكسلات المحيطة به (أصغر مصفوفة يقع هذا البكسل في مركزها)
- نافذة جيران البكسل (النافذة المحلية): هي مصفوفة مربعة ذات أبعاد فردية (قد تكون مستطيلة أو دائرية)

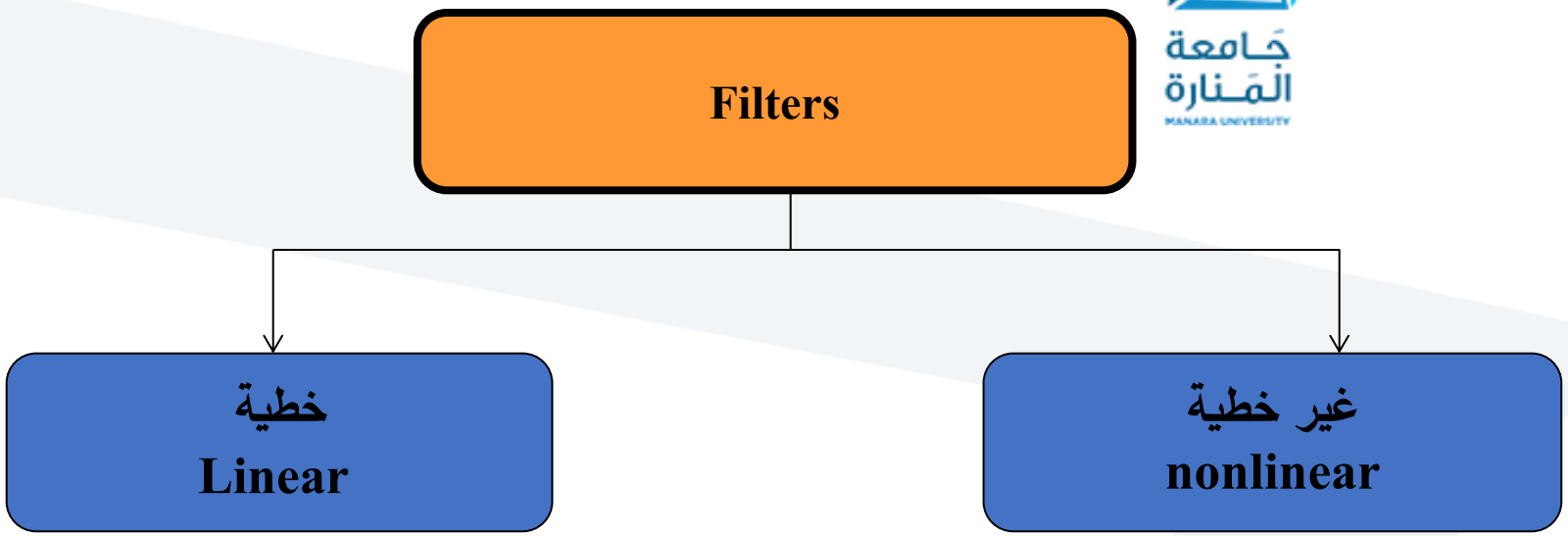






فإذا كانت إحداثيات البكسل المرجعي هي u,v
فعندها تتراوح إحداثيات جوارهذا البكسل ضمن المجال $u,v \pm k$

التصنيف العام للمرشحات

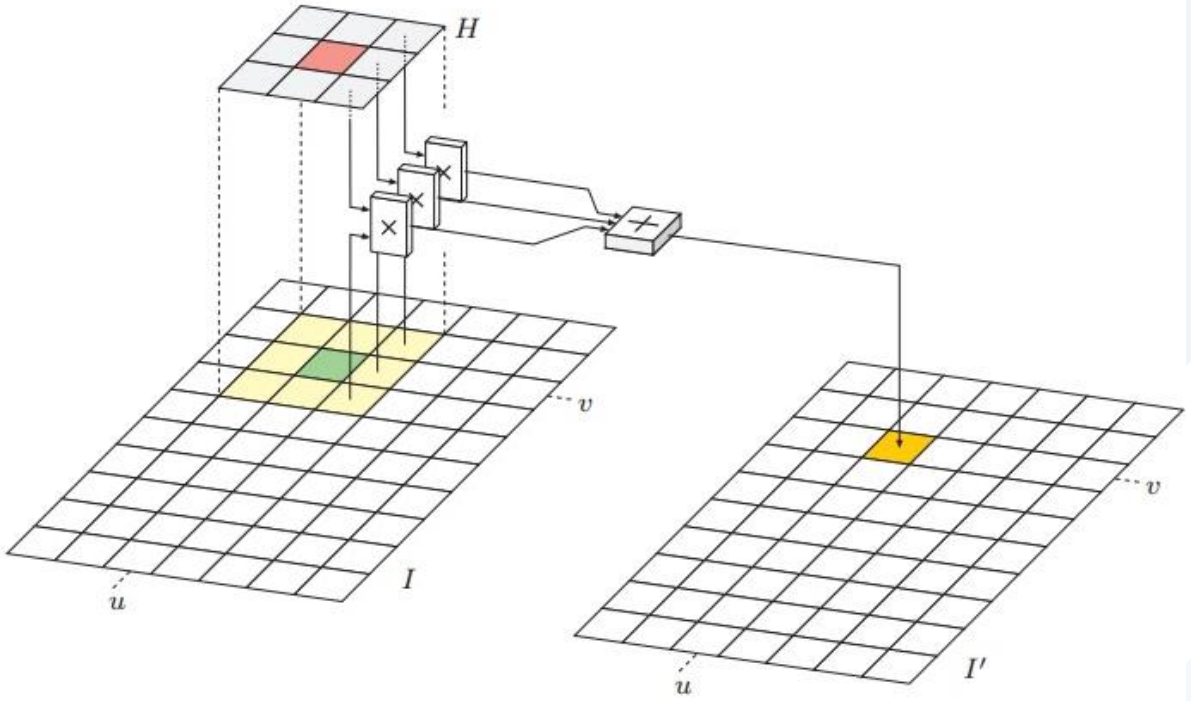


- تُطبق عمليات الترشيح على كامل الصورة وفق مبدأ النافذة المنزلقة:
- تسمى النافذة المنزلقة بـ mask أو kernels أو window أو filter.
- تتّم معالجة كل بكسل في الصورة بالاعتماد على عملية بكسلات هذه النافذة و تستخدم جيران هذا البكسل في الصورة الأصلية.
- تصنف عموماً (تبعاً للخصائص الرياضية لتابع المرشح) إلى مرشحات خطية وغير خطية .

❑ معظم الأقنعة الشائعة هي مصفوفات متناظرة حول مركزها

المرشحات الخطية

- المرشح الخطي هو الذي تعطى فيه القيم الجديدة للبكسلات عن طريق جمع خطي لقيم البكسل مع جيرانه
- تحدد عمليات الجمع الخطي لجيران البكسل عن طريق نواة المرشح أو ما يدعى بالقناع

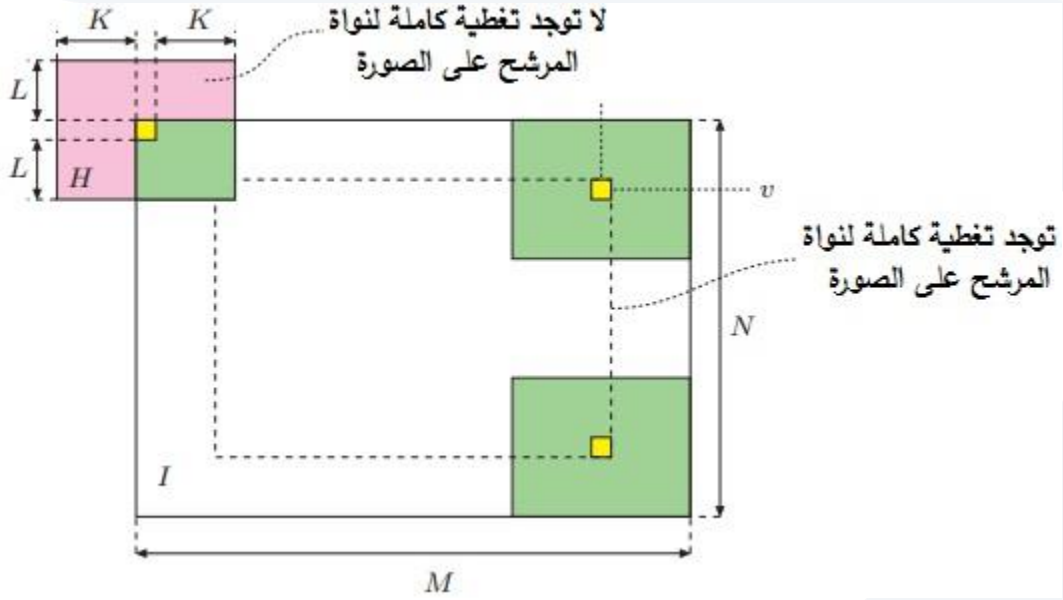


عندما تستخدم المرشحات الخطية في عمليات تنعيم الصورة أو إزالة التشويش فإنها لا تستهدف هذا التشويش فقط وانما تقوم بتنعيم كلّ محتويات الصورة بما فيها النقاط – والخطوط والحواف مما يسبب انخفاضاً في جودة الصورة

وتعدّ هذه من أهم سيئات المرشحات الخطية التي لا يمكن تجاوزها ولهذا يتمّ عادة استبعاد المرشحات الخطية

التعامل مع حدود الصورة

- هناك عدّة طرق للتعامل مع حدود الصورة:



✓ تجاهل حدود الصورة

- الحفاظ على قيم البكسلات
- استبدال قيم البكسلات بقيمة ثابتة

✓ إضافة حدود صفرية إضافية للصورة الأصلية

✓ تكرار حدود الصورة الأصلية

✓ التعامل مع حدود صورة الدخل على أنها تابع دوري ثنائي الأبعاد يكرر قيمه في كلّ من الاتجاهين العمودي والأفقي

التشويش

- تنتج الكاميرات وقنوات بثّ الصورفي بعض الأحيان أنواعاً معيّنة من التشويش والتي تتواجد على شكل بكسلات عشوائية ومعزولة

تكون قيمة السويات الرمادية فيها خارج نطاق السويات الرمادية أي إن قيمتها قد تكون أكبر بكثير أو أصغر بكثير من قيم السويات الرمادية لجيرانها ويدعى هذا التشويش بتشويش "الملح والفلفل ". salt and pepper

يعدّ التحدي الأكبر لإزالة التشويش هو التمييز بينها وبين التفاصيل الصغيرة في الصورة كالحواف أو الخطوط أو بعض الملامح الأخرى.

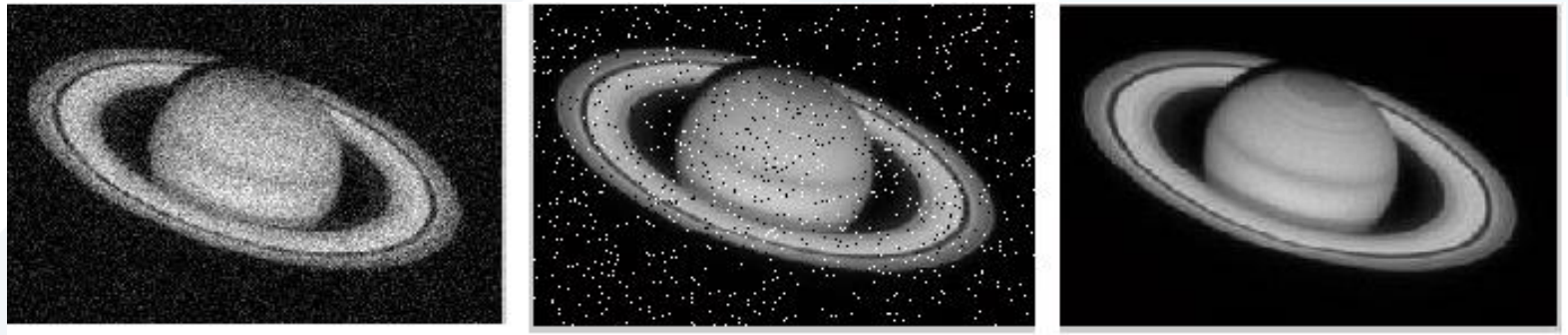
التشويش الغاوسي (: (Gaussian noise) أو الطبيعي) هو ضوضاء أو تشويش له دالة كثافة احتمالية مساوية لدالة الكثافة الاحتمالية للتوزيع الطبيعي.

التشويش

ولمقارنة وتحديد كفاءة المرشحات المختلفة نحتاج أولاً لإضافة التشويش إلى
صور ومن ثم اختبار قدرة
المرشحات على استعادة الصورة الأصلية.

Matlab code

```
a=imread('saturn.jpg');
figure, imshow(a)
asp=imnoise(a, 'salt & pepper',0.05);
figure, imshow(asp)
ag=imnoise(a,'gaussian',0.02);
```



الصورة الأصلية | إضافة تشويش الملح والفلفل | إضافة تشويش غاوص

Adding Noise :

جَامعة المَنارة
MANARA UNIVERSITY

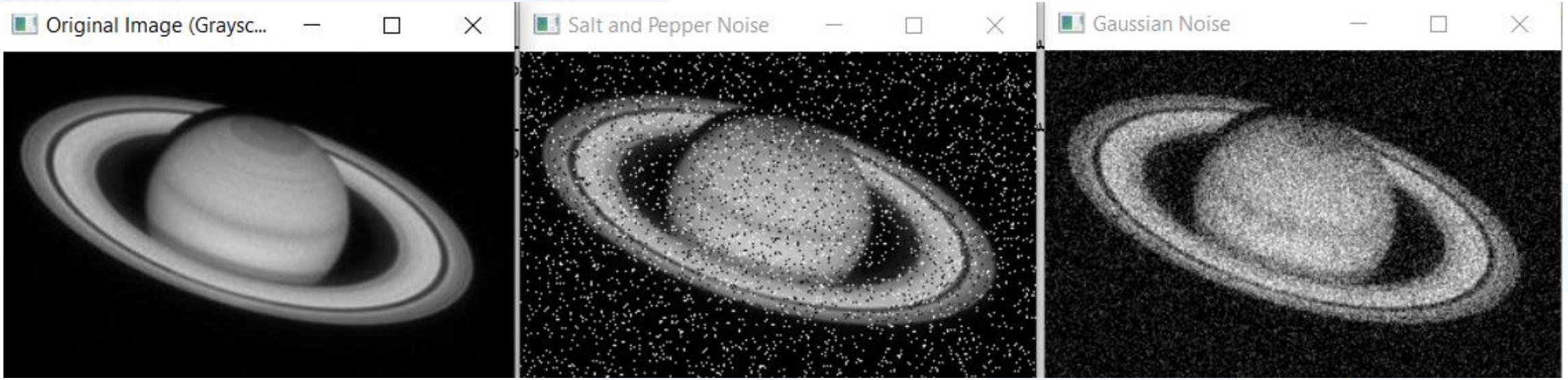
```
Noise = imnoise ( gray image , ' type ' ) ;
```

'gaussian'	Gaussian white noise with constant mean and variance
'localvar'	Zero-mean Gaussian white noise with an intensity-dependent variance
'poisson'	Poisson noise
'salt & pepper'	"On and Off" pixels
'speckle'	Multiplicative noise

```
import cv2
import numpy as np
import os # Import the 'os' module

# Load the image (replace 'saturn.jpg' with your image path)
image_path = 'saturn.jpg'
a = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

# Check if the image loaded successfully
if a is None:
    print("Error: Could not load image. Check the file path.")
    exit()

cv2.imshow('Original Image (Grayscale)', a)
cv2.waitKey(0)
```

```
# Add salt and pepper noise (intensity = 0.05)
asp = a.copy()
noise_prob = 0.05
for i in range(asp.shape[0]):
    for j in range(asp.shape[1]):
        rdn = np.random.random()
        if rdn < noise_prob:
            # Salt - set to white (255)
            asp[i, j] = 255
        elif rdn > 1 - noise_prob:
            # Pepper - set to black (0)
            asp[i, j] = 0

# Save the salt and pepper noisy image
output_path_sp =
os.path.join(os.path.dirname(image_path), 'nspsaturn.jpg')
# Construct path using original directory
cv2.imwrite(output_path_sp, asp)
print(f"Salt and pepper noise image saved to:
{output_path_sp}")
# Display the salt and pepper noisy image
cv2.imshow('Salt and Pepper Noise', asp)
cv2.waitKey(0)
```

```
# Add Gaussian noise (variance = 0.02)
ag = a.copy()
mean = 0
sigma = np.sqrt(0.02 * 255 * 255) # Standard deviation
gaussian_noise = np.random.normal(mean, sigma, ag.shape)
ag = np.clip(ag + gaussian_noise, 0, 255).astype(np.uint8)
# Save the Gaussian noisy image
output_path_gn = os.path.join(os.path.dirname(image_path), 'ngsaturn.jpg') # Construct path using original directory
cv2.imwrite(output_path_gn, ag)
print(f"Gaussian noise image saved to: {output_path_gn}")
# Display the Gaussian noisy image
cv2.imshow('Gaussian Noise', ag)
cv2.waitKey(0)

cv2.destroyAllWindows()
```


Original Image (Graysc...
Salt and Pepper Noise
Gaussian Noise

المرشحات المستخدمة للتقليل من التشويش
مرشح القيمة المتوسطة
Mean Filter

- أبسط مرشح خطي على الإطلاق

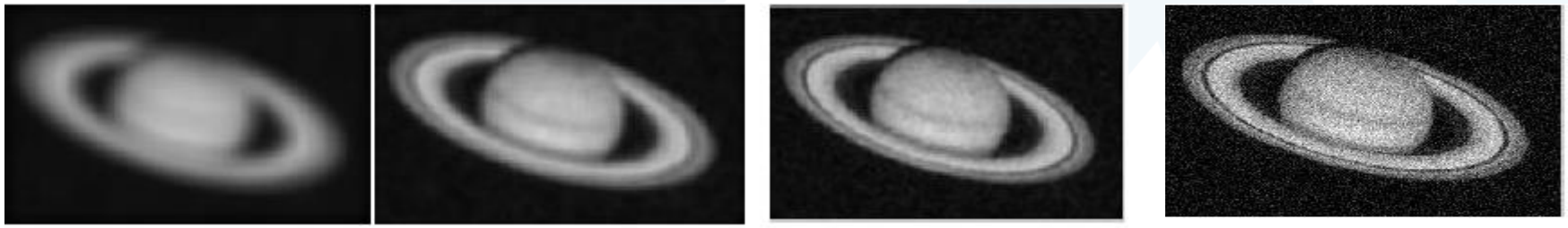
$$I'(u,v) = (P0 + P1 + P2 + P3 + P4 + P5 + P6 + P7 + P8) / 9$$

$$I'(u,v) = \frac{1}{9} \cdot \sum_{i=1}^{i=1} \sum_{j=-1}^{j=+1} I(u+i, v+j) \quad \Rightarrow \quad H(i,j) = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$H(i,j) = \frac{1}{MN}\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}$$

- ويؤدي زيادة حجم قناع المرشح إلى التقليل من تأثير تشويش غاوص ولكن على حساب جودة الصورة لأنه يقل تأثير قيمة البكسل الأصلي عن القيمة الجديدة له مقارنة بتأثير عدد أكبر من بكسلات الجوار في هذه القيمة .
- تضمن نسبة التقسيم $1/NM$ بقاء مجموع أوازن القناع مساو للواحد.

تأثير حجم المرشح في التخلص من التشويش



إضافة تشويش غاوص | قناع مرشح 5X5 | قناع مرشح 9X9 | قناع مرشح 21X21

مثال

$$\frac{1}{9}\times\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f(x,y)$

0	0	0	0	0	0	0	..
0	0	0	0	0	0	0	..
0	0	0	0	0	0	0	..
0	0	0	9	9	9	9	..
0	0	0	9	9	9	9	..
0	0	0	9	9	9	9	..
0	0	0	9	9	9	9	..
:	:	:	:	:	:	:	..

$g(x,y)=w(x,y)\otimes f(x,y)$

$f(x,y)$

0	0	0	0	0	0	0	..
0	0	0	0	0	0	0	..
0	0	0	0	0	0	0	..
0	0	0	9	9	9	9	..
0	0	0	9	9	9	9	..
0	0	0	9	9	9	9	..
0	0	0	9	9	9	9	..
:	:	:	:	:	:	:	..

$g(x,y)=w(x,y)\otimes f(x,y)$

	0	0	0	0	0	0	
	0	1	2	3	3	3	
	0	2	4	6	6	6	
	0	3	6	9	9	9	
	0	3	6	9	9	9	
	0	3	6	9	9	9	..
	0						

Average Filter :

Low pass filter

- Averaging filters

mask values

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

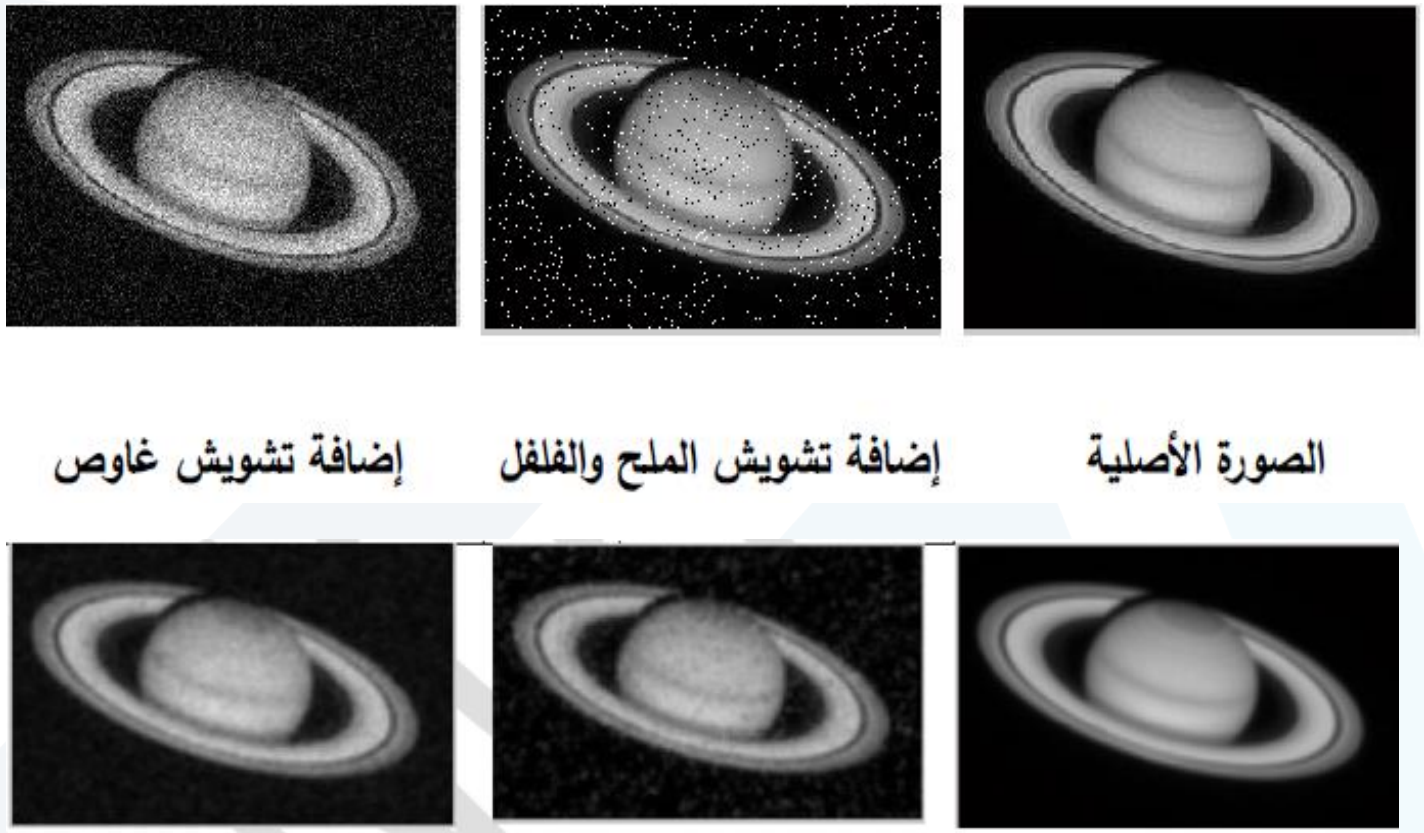
mask values

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

Input 3X3 5X5 7X7

Matlab code

```
a=imread('saturn.jpg');
b=imread('nspsaturn.jpg');
c=imread('ngsaturn.jpg');
k=ones(5)/25;
af=imfilter(a, k);
figure, imshow(af)
bf=imfilter(b, k);
figure, imshow(bf)
cf=imfilter(c, k);
figure, imshow(cf)
```



```
import cv2
import numpy as np
import os

# --- Function to apply a filter (imfilter in MATLAB) ---
def apply_filter(img, kernel):
    """Applies a 2D convolution (filtering) to an image."""
    filtered_img = cv2.filter2D(img, -1, kernel) # -1 for same depth/type as input
    return filtered_img


# Image paths (assuming they are in the current directory)
image_dir = '.'
original_image_name = 'saturn.jpg'
sp_image_name = 'nspsaturn.jpg'
gn_image_name = 'ngsaturn.jpg'

# Construct full image paths
original_image_path = os.path.join(image_dir, original_image_name)
sp_image_path = os.path.join(image_dir, sp_image_name)
gn_image_path = os.path.join(image_dir, gn_image_name)
```

```
# --- Load Images ---

# Original Grayscale Image
a = cv2.imread(original_image_path, cv2.IMREAD_GRAYSCALE)
if a is None:
    print(f"Error: Could not load original image: {original_image_path}")
    exit()

# Salt and Pepper Image
b = cv2.imread(sp_image_path, cv2.IMREAD_GRAYSCALE)
if b is None:
    print(f"Error: Could not load salt and pepper image: {sp_image_path}")
    exit()

# Gaussian Noise Image
c = cv2.imread(gn_image_path, cv2.IMREAD_GRAYSCALE)
if c is None:
    print(f"Error: Could not load Gaussian noise image: {gn_image_path}")
    exit()
```

```
# --- Define the Kernel ('ones(5)/25' in MATLAB) ---
k = np.ones((5, 5), np.float32) / 25 # Creates a 5x5 averaging kernel

# --- Apply Filters ---

# Filter the original image
af = apply_filter(a, k)
cv2.imshow('Filtered Original', af)
cv2.waitKey(0)

# Filter the salt and pepper image
bf = apply_filter(b, k)
cv2.imshow('Filtered Salt & Pepper', bf)
cv2.waitKey(0)

# Filter the Gaussian noise image
cf = apply_filter(c, k)
cv2.imshow('Filtered Gaussian', cf)
cv2.waitKey(0)

cv2.destroyAllWindows()
```

Application : noise reduction and image smoothing
Disadvantage: lose sharp details

المرشحات المستخدمة للتقليل من التشويش
مرشح القيمة المتوسطة المثقل
Weighted Mean Filter

- مرشح يأخذ بالحسبان مقدار قرب البكسل الجار من البكسل الأصلي ويعطي للبكسل الأصلي قيمة أكبر من جيرانه ليكون له وللجيران الأقرب نسب أكبر من المشاركة مقارنة مع البكسلات البعيدة

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

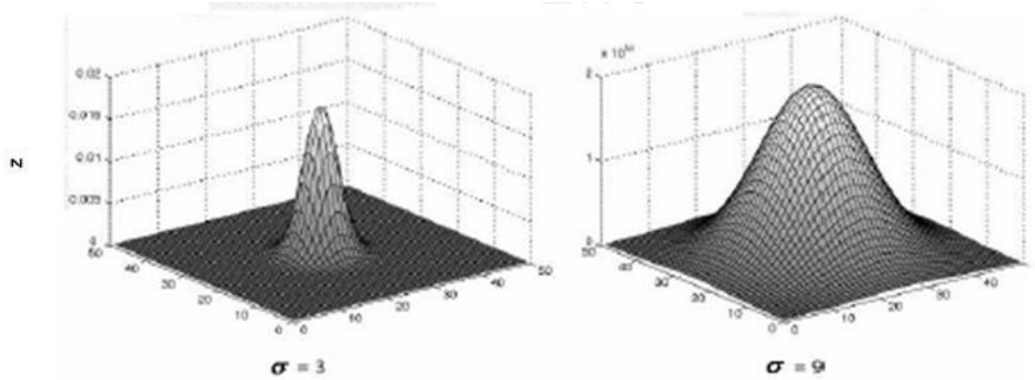
المرشحات المستخدمة للتقليل من التشويش
Gaussian Filter مرشح غاوص

- يمثل مرشح قيمة متوسطة مثقل يزداد وزن البكسل فيه كلما اقتربنا من البكسل المركزي
- يستخدم في إزالة التشويش وتنعيم الصور ويستخدم قناعاً يمثل شكل توزع غاوص

$$(1\text{-D}):\quad g(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{(x-\bar{x})^2}{2\sigma^2}}$$

يمثل σ الانحراف المعياري لتوزع غاوص

- توزع غاوص ثنائي البعد



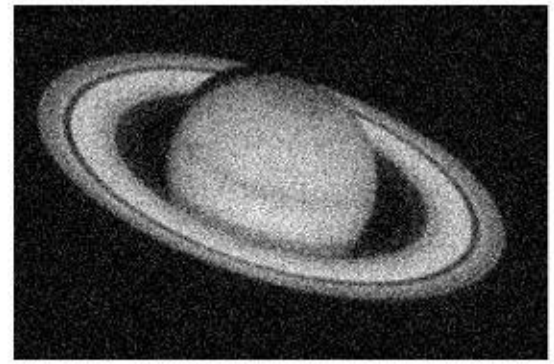
$$(2\text{-D}):\quad g(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

المرشحات المستخدمة للتقليل من التشويش
Gaussian Filter مرشح غاوص

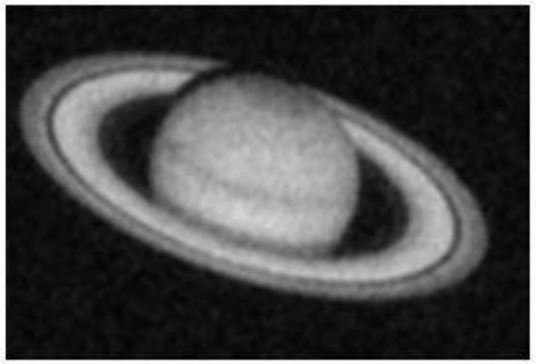
- يقوم مرشح غاوص بتنعيم الصورة وإزالة التشويش مثله مثل مرشح القيمة المتوسطة ولكن يختلف عنه بأنه يستخدم مرشح غاوصي أو قناعا له شكل توزع غاوص
- تحدد درجة التنعيم للصورة عن طريق تغيير قيمة الانحراف المعياري

Matlab code

```
a=imread('ngsaturn.jpg');
figure, imshow(a)
k=fspecial('gaussian',[5 5],2);
af=imfilter(a, k, 'conv');
```



صورة تحتوي على تشويش غاوص



الصورة بعد تطبيق مرشح غاوص

- يقوم برنامج الماتلاب بتوليد المصفوفة الخاصة بقناع مرشح غاوص باستخدام التعليمة fspecial

Mask = fspecial (‘ type ‘) ;

Image = imfilter (gray image , mask) ;

```
'average'   averaging filter
'disk'      circular averaging filter
'gaussian'  Gaussian lowpass filter
'laplacian' filter approximating the 2-D Laplacian operator
'log'       Laplacian of Gaussian filter
'motion'    motion filter
'prewitt'   Prewitt horizontal edge-emphasizing filter
'sobel'     Sobel horizontal edge-emphasizing filter
'unsharp'   unsharp contrast enhancement filter
```

- يقوم مرشح غاوص بتنعيم الصورة على نحو مشابه لمرشح القيمة المتوسطة. ويتمّ تحديد درجة التنعيم للصورة عن طريق تغيير قيمة الانح ا رف المعياري؛
- إذ تؤدي زيادة الانحراف المعياري لتوزع غاوص إلى زيادة درجة التنعيم في الصورة.
- ويعدّ مرشح غاوص مرشح قيمة متوسطة مثقل يزداد وزن البكسل فيه كلما اقترب من البكسل المركزي.
- لذلك ينتج عنه تنعيم سلس وحفاظ على تفاصيل الصورة على نحو أفضل من مرشح القيمة المتوسطة ويستخدم لهذا كخطوة أولى في خوارزميات الكشف عن الحواف) مثل كاشف الحواف كاني
- يستخدم مرشح غاوص عادةً كمرشح ترددات منخفضة إلا أنه يمكن تطبيقه بشكل مباشر على الصورة وذلك عن طريق توليد قناع تمثل أوزانه القيم التقريبية لتوزع غاوص.

Gaussian filters

$5 \times 5, \sigma = 0.5$

$11 \times 11, \sigma = 1$

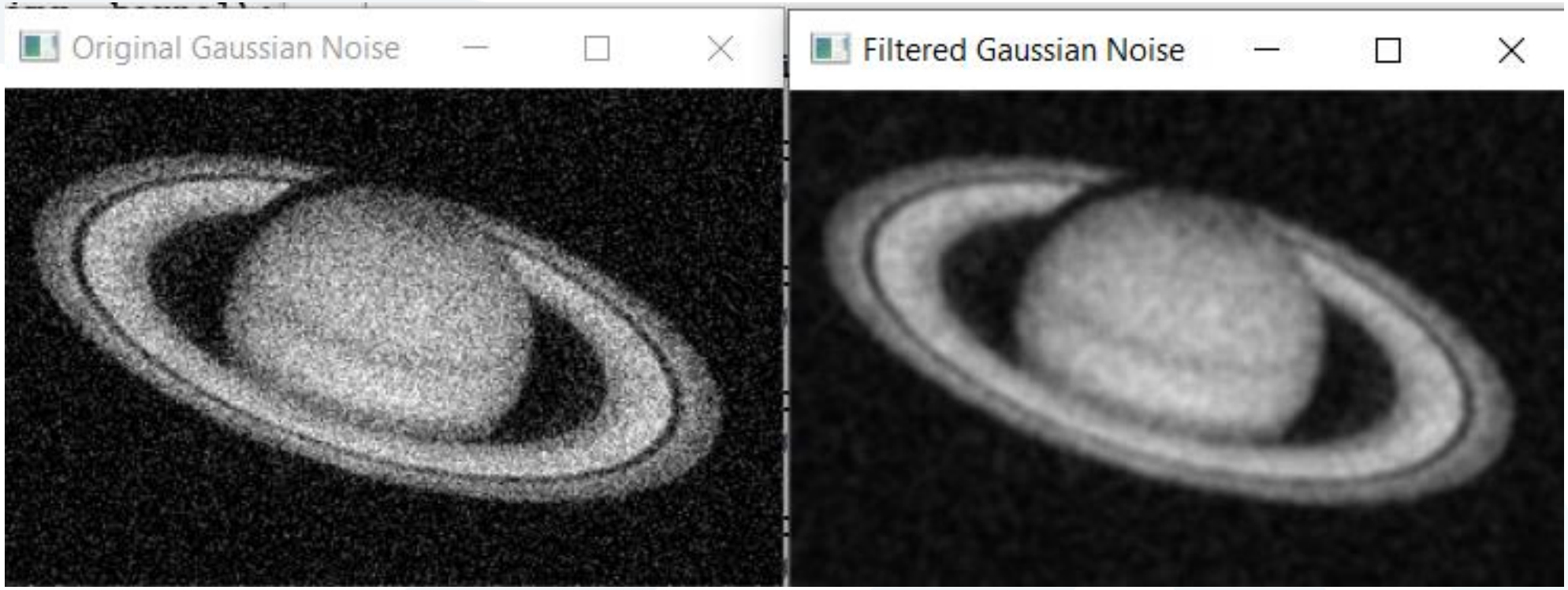
$5 \times 5, \sigma = 2$

$11 \times 11, \sigma = 5$

```
import cv2
import numpy as np
import os

# --- Function to apply a filter (imfilter in MATLAB) ---
def apply_filter(img, kernel):
    """Applies a 2D convolution (filtering) to an image."""
    filtered_img = cv2.filter2D(img, -1, kernel) # -1 for same
depth/type as input
    return filtered_img

# --- Script starts here ---
# Image path
image_dir = '.' # Current directory
gn_image_name = 'ngsaturn.jpg'
gn_image_path = os.path.join(image_dir, gn_image_name)
```

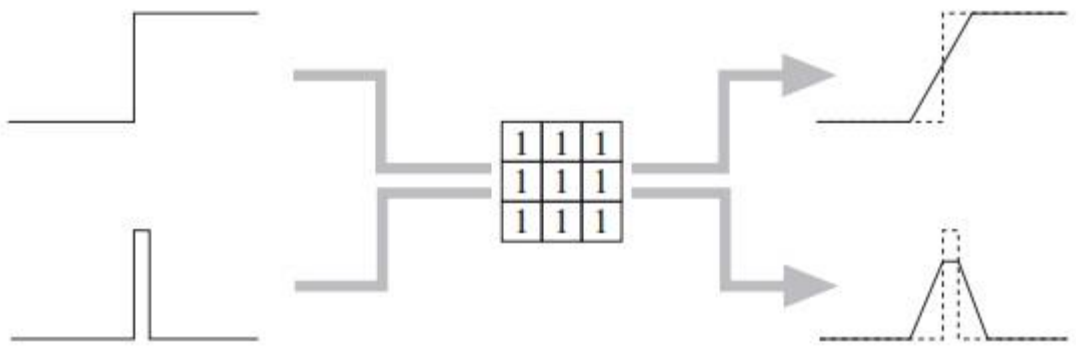
```
# --- Load Image ---
a = cv2.imread(gn_image_path, cv2.IMREAD_GRAYSCALE)
if a is None:
    print(f"Error: Could not load image: {gn_image_path}")
    exit()

# Display original image
cv2.imshow('Original Gaussian Noise', a)
cv2.waitKey(0)
```

```
# --- Create Gaussian Kernel (fspecial('gaussian', [5 5], 2) in MATLAB) ---
# Sigma = 2, kernel size [5,5]
k = cv2.getGaussianKernel(5, 2) # Creates a 1D Gaussian kernel, then combines it.
k = k @ k.T # Creates a 2D Gaussian kernel by outer product.


# --- Apply Filter ---
af = apply_filter(a, k) # Apply the Gaussian filter.

# Display the filtered image
cv2.imshow('Filtered Gaussian Noise', af)
cv2.waitKey(0)

cv2.destroyAllWindows()
```


Original Gaussian Noise
Filtered Gaussian Noise

المرشحات غير الخطية

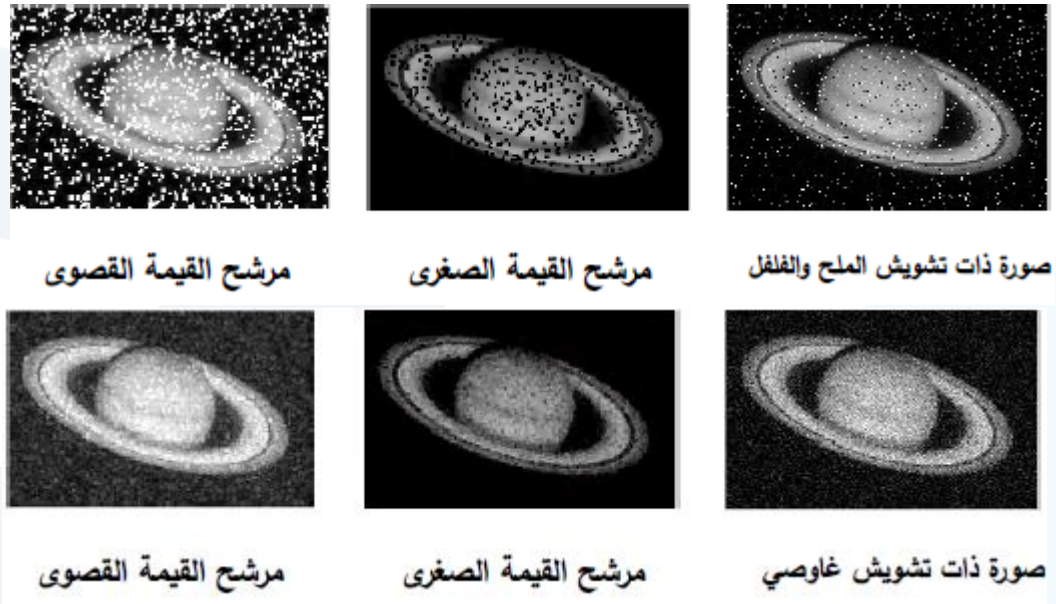
- من سيئات المرشحات الخطية أنها تقوم بتنعيم كل محتويات وتفاصيل الصورة بما فيها النقاط والخطوط والحواف



- بعض المرشحات غير الخطية التي تستخدم في التخلص من التشويش
  - ✓ مرشحات القيمة العظمى والصغرى Max and Min Filters
  - ✓ مرشح الوسيط الهندسي Median filter
  - ✓ مرشح الوسيط الهندسي المثقل Weighted median filter

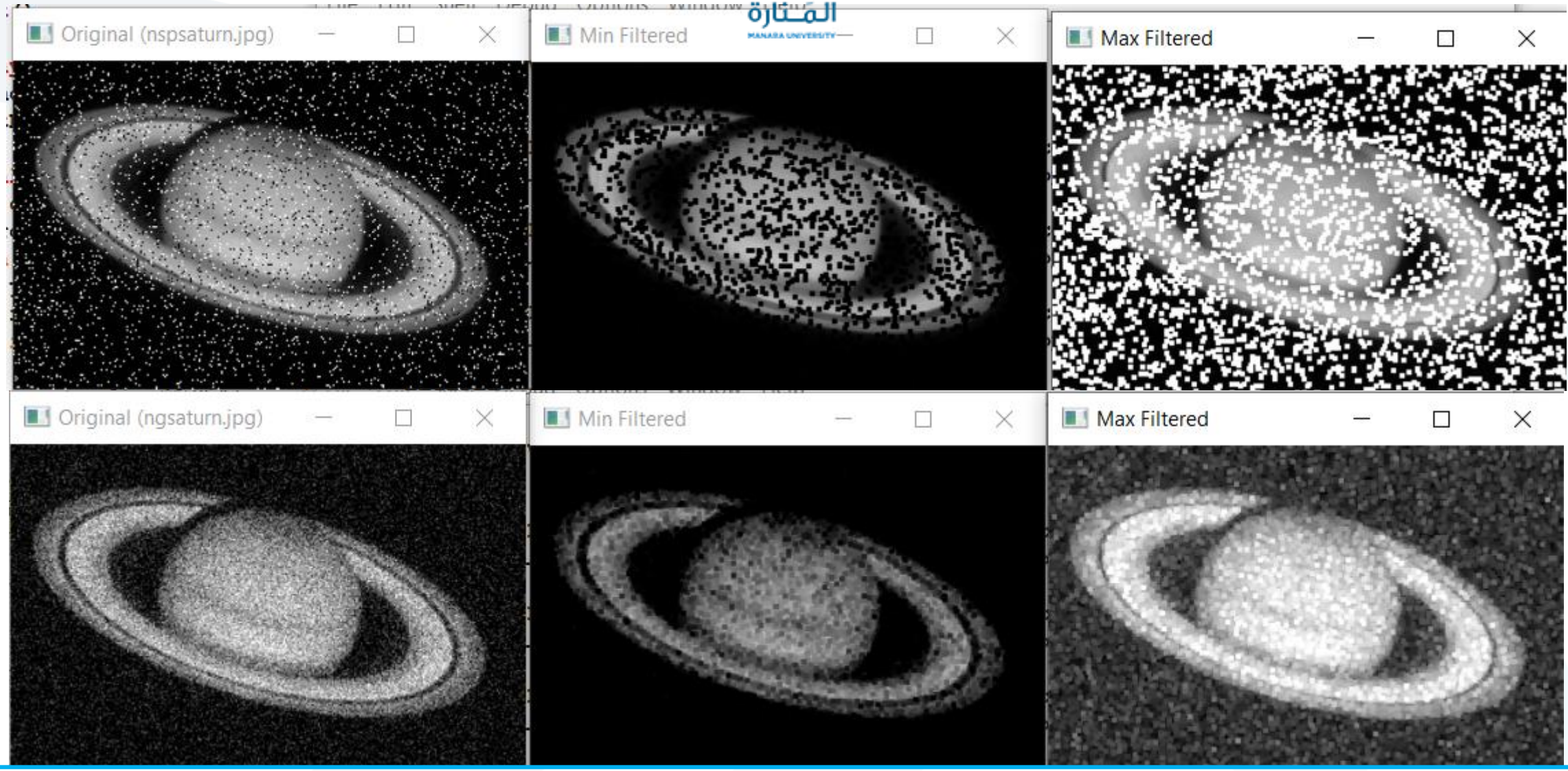
Matlab code

```
a=imread('nspsaturn.jpg');
figure, imshow(a)
amin=uint8(colfilt(a,[3 3],'sliding', @min));
figure, imshow(amin)
amax=uint8(colfilt(a,[3 3],'sliding', @max));
figure, imshow(amax)
```



- مرشح القيمة الصغرى قد استطاع التخلص من النقاط البيضاء – في الصورة "الملح" لأنه سيتمّ استبدال أيّ بكسل أبيض وحيد في الصورة بقيمة بكسل آخر من جواره تكون قيمته أصغر ولكن هذا المرشح قد زاد في الوقت نفسه من النقاط السوداء في الصورة "الفلفل".
- ويظهر مرشح القيمة القصوى التأثير المعاكس لسابقه فوجود أيّ بكسل أبيض وحيد في نافذة المرشح سيعطي قيمته للبكسل وبالآتي سيزيد من تشويش الملح أما النقط السوداء في الصورة فستختفي تماماً.

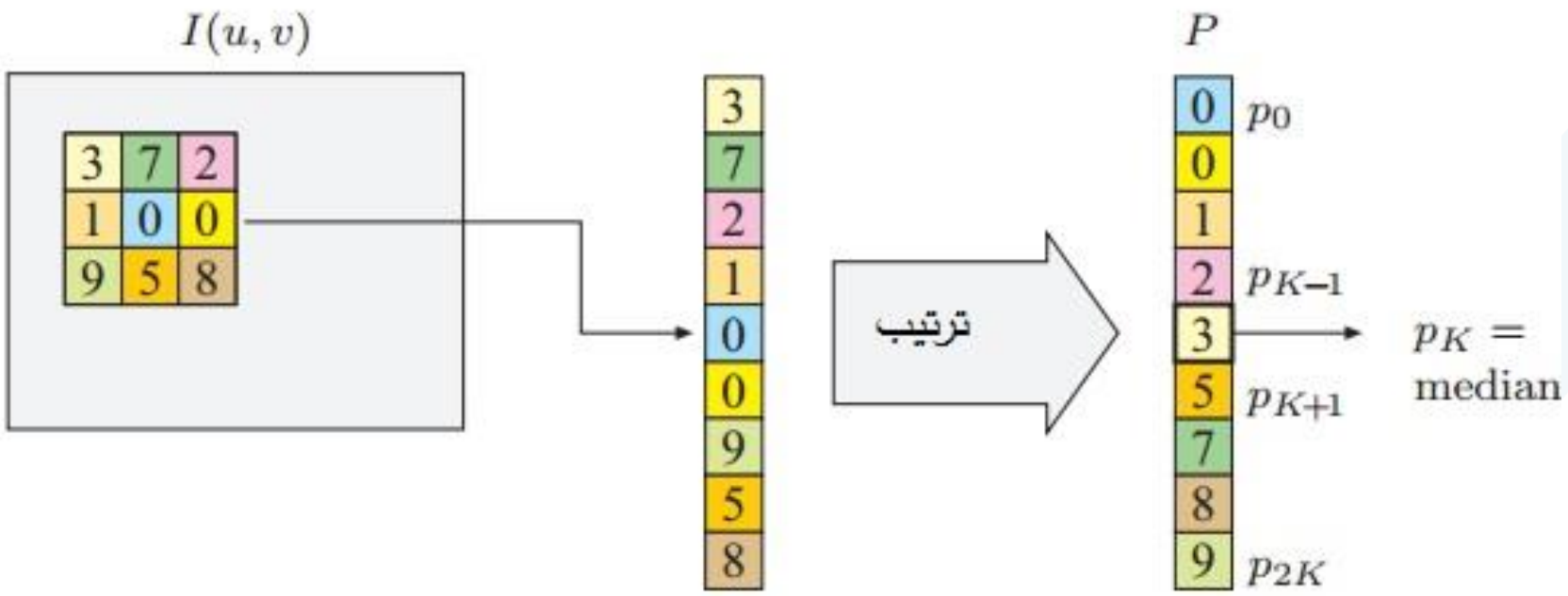
```
import cv2
import numpy as np
import os
# --- Function to apply filter (using sliding window and min/max) ---
def apply_min_max_filter(img, kernel_size, operation):
    """Applies a min or max filter using a sliding window."""
    rows, cols = img.shape
    filtered_img = np.zeros_like(img)

    half_kernel = kernel_size // 2 # Integer division for kernel radius

    for r in range(rows):
        for c in range(cols):
            # Define the window boundaries
            row_start = max(0, r - half_kernel)
            row_end = min(rows, r + half_kernel + 1)
            col_start = max(0, c - half_kernel)
            col_end = min(cols, c + half_kernel + 1)

            # Extract the window (the kernel area)
            window = img[row_start:row_end, col_start:col_end]

            # Apply min or max operation
            if operation == 'min':
                filtered_img[r, c] = np.min(window)
            elif operation == 'max':
                filtered_img[r, c] = np.max(window)
            else:
                raise ValueError("Invalid operation. Choose 'min' or 'max'.")

    return filtered_img.astype(np.uint8) # Ensure uint8 output

# Image path
image_dir = '.'
#sp_image_name = ‘nspsaturn.jpg'
sp_image_name = 'ngsaturn.jpg’
sp_image_path = os.path.join(image_dir, sp_image_name)

# --- Load Image ---
a = cv2.imread(sp_image_path, cv2.IMREAD_GRAYSCALE)
if a is None:
    print(f"Error: Could not load image: {sp_image_path}")
    exit()

# Display Original Image
# cv2.imshow('Original (nspsaturn.jpg)', a)
cv2.imshow('Original (ngsaturn.jpg)', a)
cv2.waitKey(0)

# --- Apply Min Filter (3x3) ---
amin = apply_min_max_filter(a, 3, 'min')
cv2.imshow('Min Filtered', amin)
cv2.waitKey(0)

# --- Apply Max Filter (3x3) ---
amax = apply_min_max_filter(a, 3, 'max')
cv2.imshow('Max Filtered', amax)
cv2.waitKey(0)

cv2.destroyAllWindows()
```


Original (nspsaturn.jpg)
Min Filtered
Max Filtered
Original (ngsaturn.jpg)
Min Filtered
Max Filtered

المرشحات غير الخطية
مرشح الوسيط الهندسي
Median Filter

- إن الهدف الرئيس من المرشحات هو التخلص من التشويش مع إبقاء كلّ تفاصيل الصورة ولكن هذا السبب صعب التحقق لأن المرشحات لا تستطيع تمييز بين محتويات الصورة المهمة وغير المهمة.
- ويعدّ مرشح الوسيط الهندسي من أكثر المرشحات شعبية في مجال التخلص من التشويش.
- عند تطبيق مرشح الوسيط الحسابي يتمّ استبدال كلّ بكسل في الصورة بقيمة الوسيط الهندسي لنافذة المرشح R وفق المعادلة الآتية:

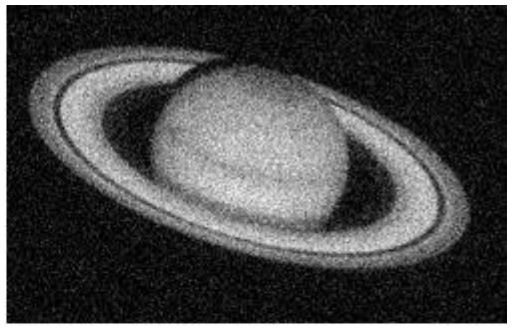
$$I'(u,v) = median\{I(u+i, v+j) \mid (i,j) \in R\}$$



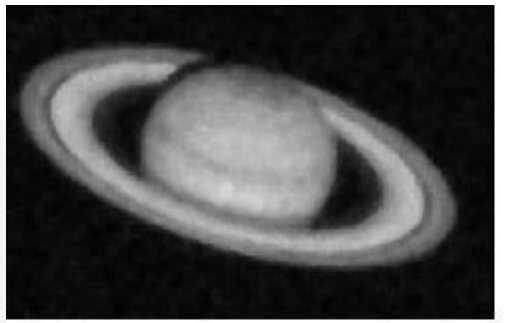
مرشح الوسيط الهندسي

مثال

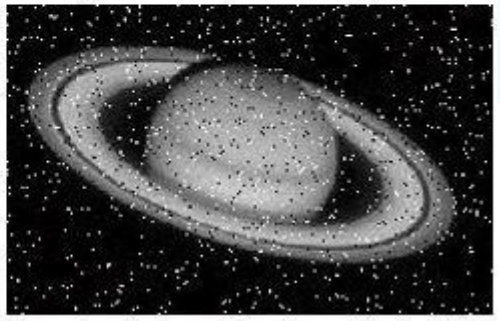
```
a=imread('ngsaturn.jpg');
figure, imshow(a)
af=medfilt2(a,[7 7]);
figure, imshow(af)
```



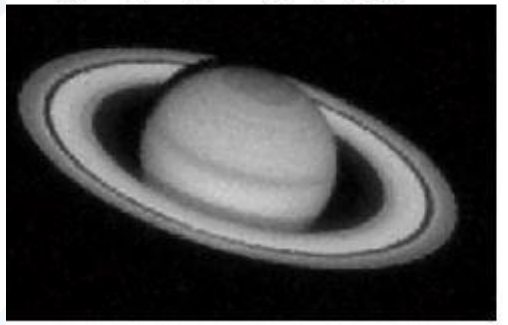
صورة ذات تشويش غاوصي



تطبيق مرشح الوسيط الهندسي



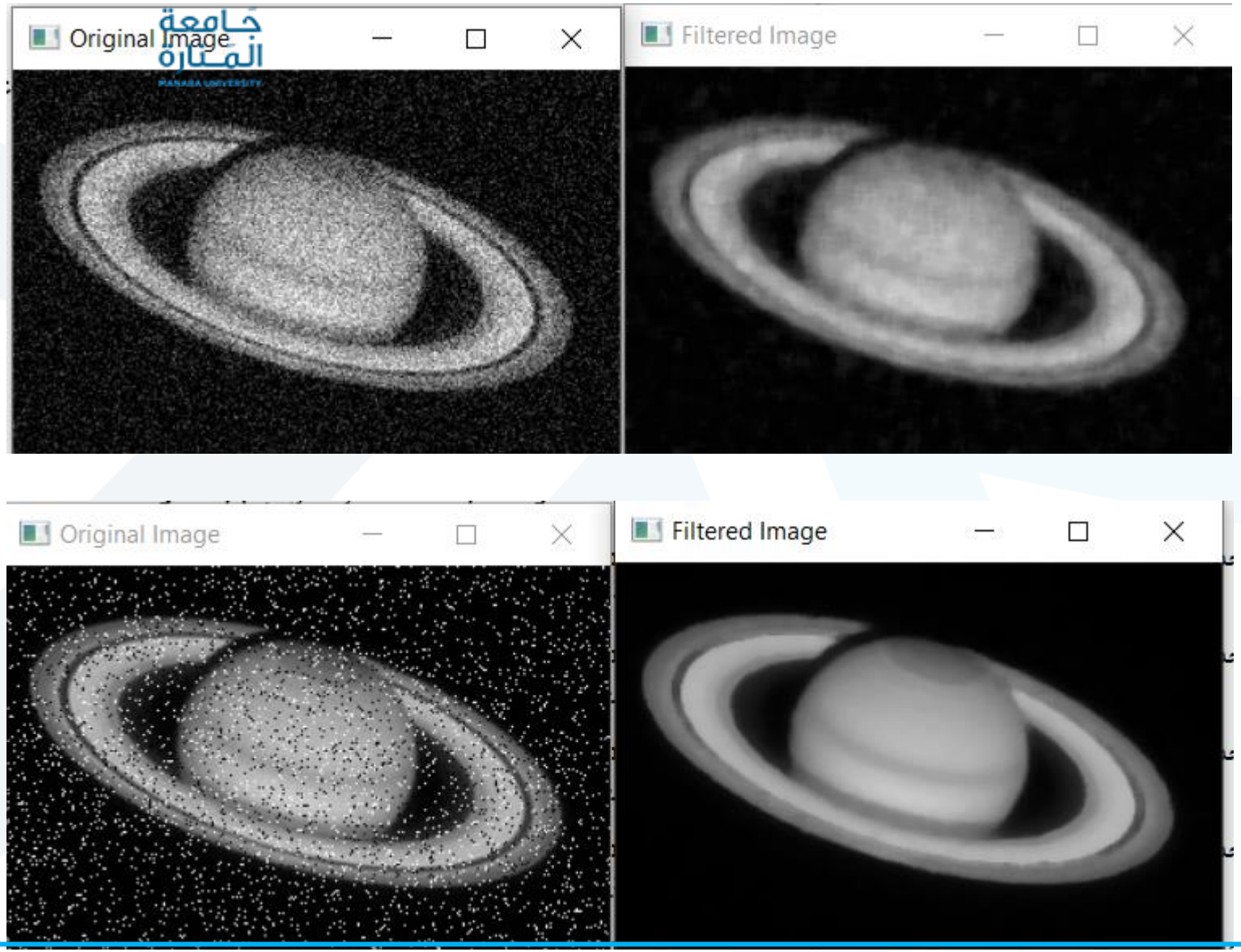
صورة ذات تشويش الملح والفلفل

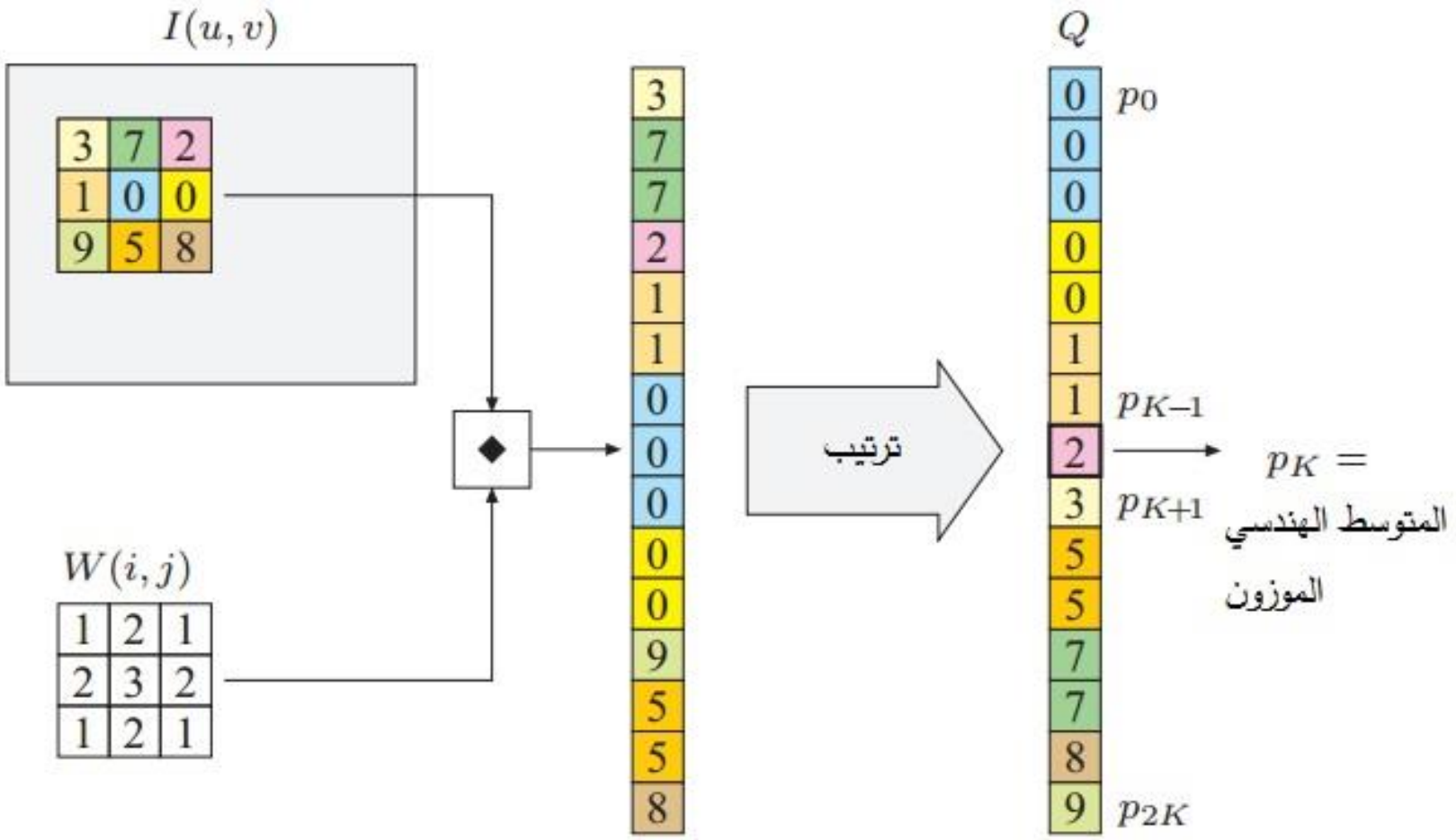


تطبيق مرشح الوسيط الهندسي

- أن مرشح الوسيط الهندسي قد استطاع التغلب على - بقية المرشحات السابقة في الحفاظ على تفاصيل الصورة مع القضاء على التشويش
- وخصوصاً عندما تكون بكسلات التشويش معزولة وذات قيمة مرتفعة جداً أو منخفضة جداً)كتشويش الملح والفلفل(أما في حالة تشويش غاوص فقد استطاع هذا المرشح
- إ زالة التشويش ولكن على حساب التقليل من جودة الصورة.

```
import cv2
import numpy as np

# Read the image
image = cv2.imread('ngsaturn.jpg’)
#image = cv2.imread(‘nspsaturn.jpg')

# Display the original image
cv2.imshow('Original Image', image)

# Apply median filtering with a 7x7 kernel
filtered_image = cv2.medianBlur(image, 7)

# Display the filtered image
cv2.imshow('Filtered Image', filtered_image)

# Wait for a key press and close the image windows
cv2.waitKey(0)
cv2.destroyAllWindows()
```



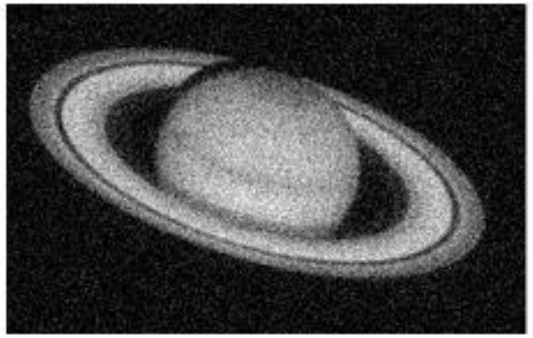


- تستدعي هذه المصفوفة توسعة شعاع البكسل ذي الطول 9 عناصر ليصبح طوله 15 عنصر والذي يساوي مجموع عناصر مصفوفة الأوزان W .
- يحدد مرشح الوسيط الهندسي المثقل أوزاناً منفصلة لكلّ موقع في نافذة المرشح والتي تفسر على أنها عدد الأصوات المعطاة لهذا الموقع.

مرشح الوسيط الهندسي المثقل
مثال

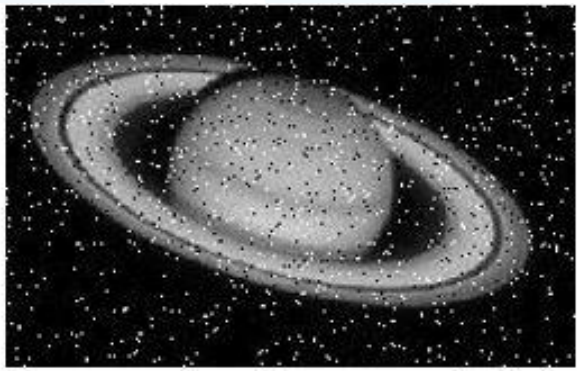
```
a=imread('ngsaturn.jpg');
figure, imshow(a)
af=ordfilt2(a,5,ones(5));
```

B = ordfilt2(A,order,domain) replaces each element in A by the orderth element in the sorted set of neighbors specified by the nonzero elements in domain.

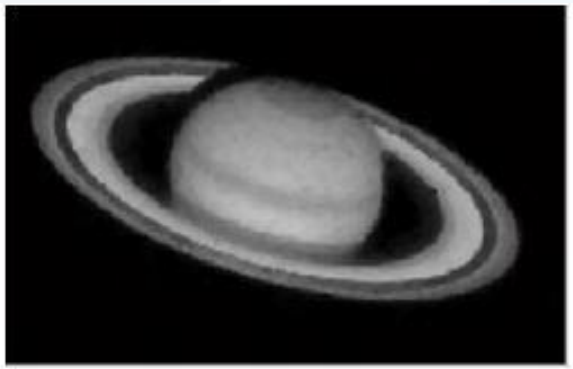


صورة تحتوي على تشويش غاوص

تطبيق مرشح الوسيط الهندسي



صورة تحتوي على تشويش الملح والفلفل



تطبيق مرشح الوسيط الهندسي

Type of Filtering Operation	MATLAB code	Neighborhood	Sample Image Data, Indica
Median filter	`B = ordfilt2(A,5,ones(3,3))`	1 1 1 1 1 1 1 1 1	88 16 56 5 3 **30** 21 63 42
Minimum filter	`B = ordfilt2(A,1,ones(3,3))`	1 1 1 1 1 1 1 1 1	88 16 56 5 **3** 30 21 63 42
Maximum filter	`B = ordfilt2(A,9,ones(3,3))`	1 1 1 1 1 1 1 1 1	**88** 16 56 5 3 30 21 63 42
Minimum of north, east, south, and west neighbors	`B = ordfilt2(A,1,[0 1 0; 1 0 1; 0 1 0])`	0 1 0 1 0 1 0 1 0	88 16 56 **5** 3 30 21 63 42

جَامعة
المَنارة
MANARA UNIVERSITY

نهاية المحاضرة