

# Computer Vision Lecture 5

## Features – *Part2*

(Scale Invariant Features, Template Matching)

Dr. Ali Mahmoud Mayya

Computer Science Dept.

AL Manara University, Syria

2024

# What are features to study

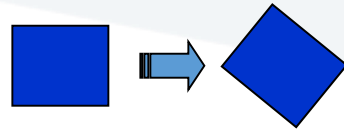
السمات التي سندرسها

- كشف السمات الثابتة Scale Invariant Features
- القوالب Templates

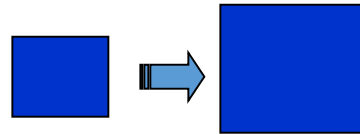
# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## العمليات الهندسية على الصورة

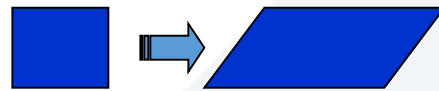
- تدوير Rotation



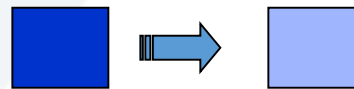
- تحجيم Scale



- تغيير الأبعاد Affine



- قياسات الضوء Photometric



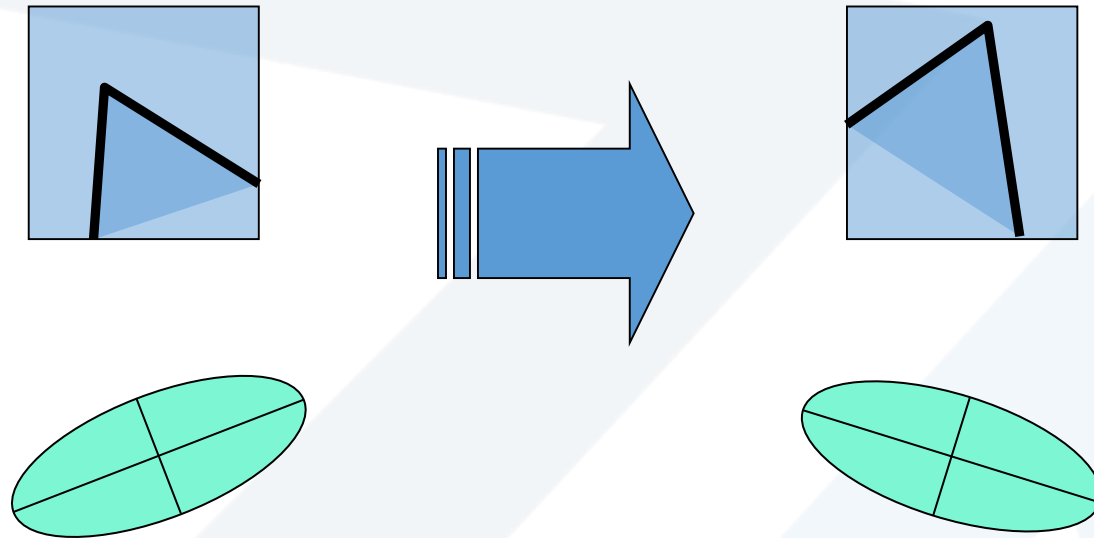
- Affine intensity change (التغير في السوية الرمادية  $I \rightarrow aI + b$ )

إذا كان لدينا صورتان لنفس  
المشهد لكن الثانية (مدورة أو  
محجمة) أي متغيرة Variance  
هل يمكن كشف السمات السابقة  
(الحواف والزوايا) بنفس المكان  
في كلتا الصورتين؟

# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## الخصائص الثابتة في كاشف الزوايا هاريس Harris Detector: Invariance Properties

### • التدوير Rotation



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

دوران الزاوية يؤدي لدوران القطع الناقص لمعاملتي  $\lambda_1$  و  $\lambda_2$  لكن شكله يبقى نفسه تماماً

*Corner response  $R$  is invariant to image rotation*

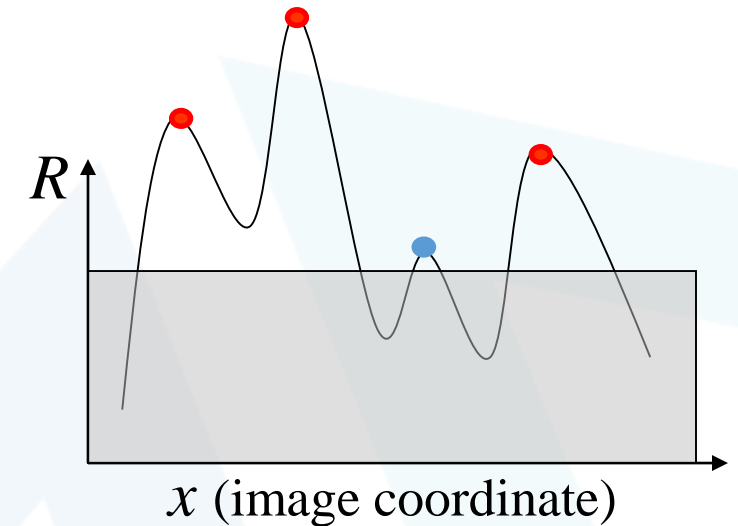
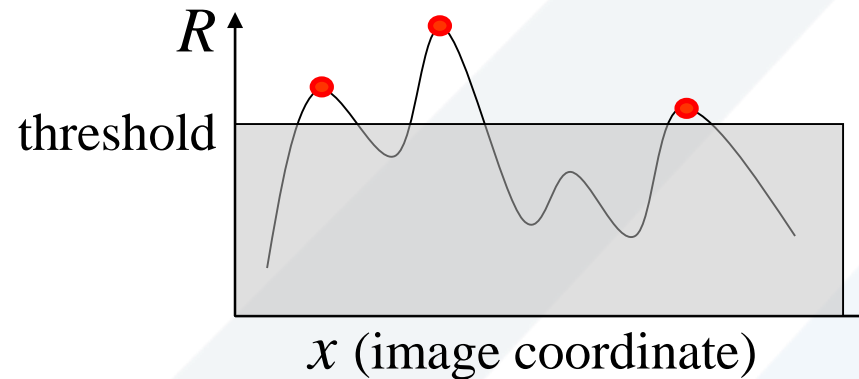
**نتيجة:** تابع استجابة الزاوية  $R$  في كاشف هاريس ثابت مع تغير زاوية التدوير

# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## الخصائص الثابتة في كاشف الزوايا هاريس Harris Detector: Invariance Properties

- السوية اللونية Affine

✓ Intensity scale:  $I \rightarrow a I$



✓ Only derivatives are used  $\Rightarrow$  invariance to intensity shift  $I \rightarrow I + b$

✓ بما أنَّ الكاشف يعتمد على المشتقات فقط -- < كاشف الزوايا هاريس ثابت تجاه تغيرات الإزاحة (جمع ، طرح)

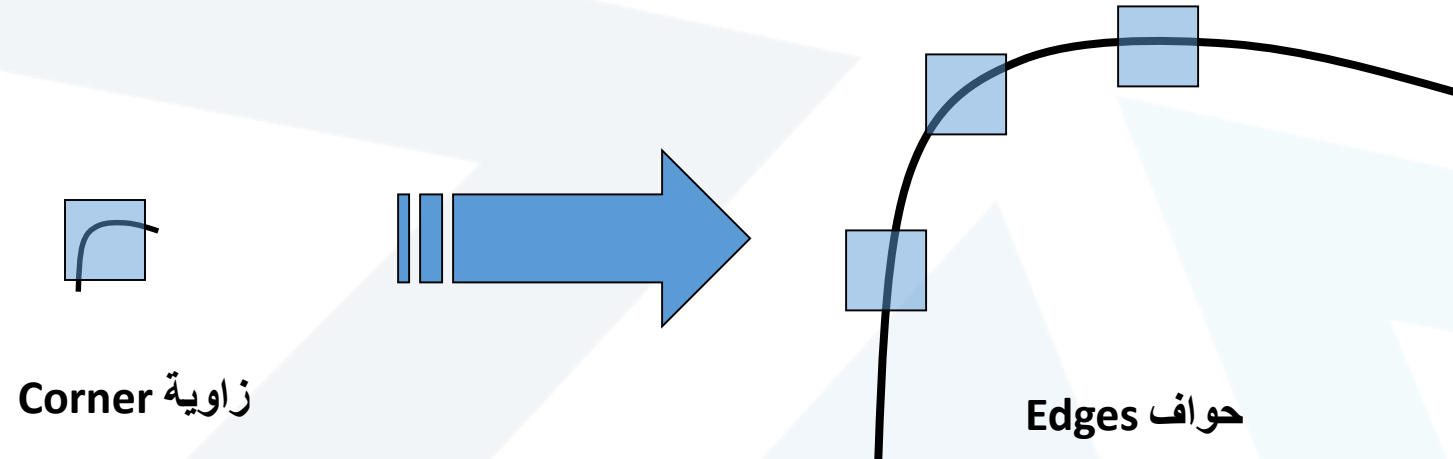
*Partially invariant to affine intensity change*

**نتيجة:** تابع استجابة الزاوية  $R$  في كاشف هاريس ثابت جزئياً (ليس بشكل كامل) مع تغير الإضاءة (السويات)

# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## الخصائص الثابتة في كاشف الزوايا هاريس Harris Detector: Invariance Properties

- التحجيم Scale



✓ عند زيادة حجم الزاوية تتحول عناصر الزاوية إلى حواف في استجابة كاشف هاريس

*Not invariant to scaling*

**نتيجة:** تابع استجابة الزاوية R في كاشف هاريس **متغير** مع تغير الحجم !

# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## كشف السمات الثابتة Scale-invariant feature detection

### Blobs Detection

1

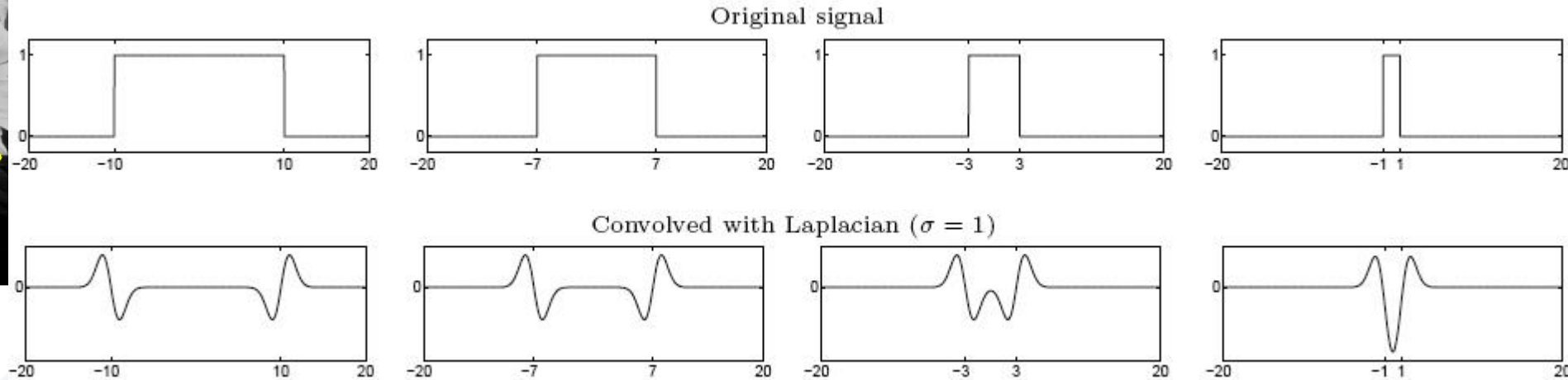
مرشح اللابلاسيان الغوسي Laplacian of Gaussian

الحافة = موجة واحدة في المشتق Edge = ripple

Blob = superposition of two ripples

Blob هو تراكم لموجتين في استجابة المشتق

إذاً: لكشف Blob نستخدم مرشح LOG اللابلاسيان الغوسي

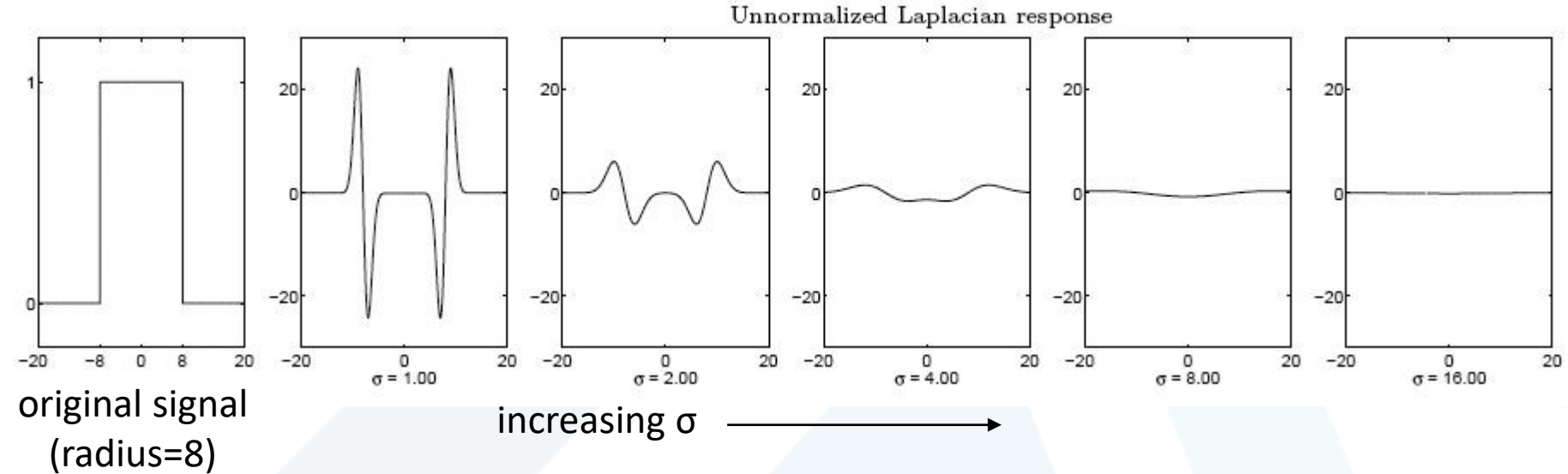


تزداد استجابة مرشح اللابلاسيان الغوسي بشكل أكبر مع  
الاقتراب من مركز Blob

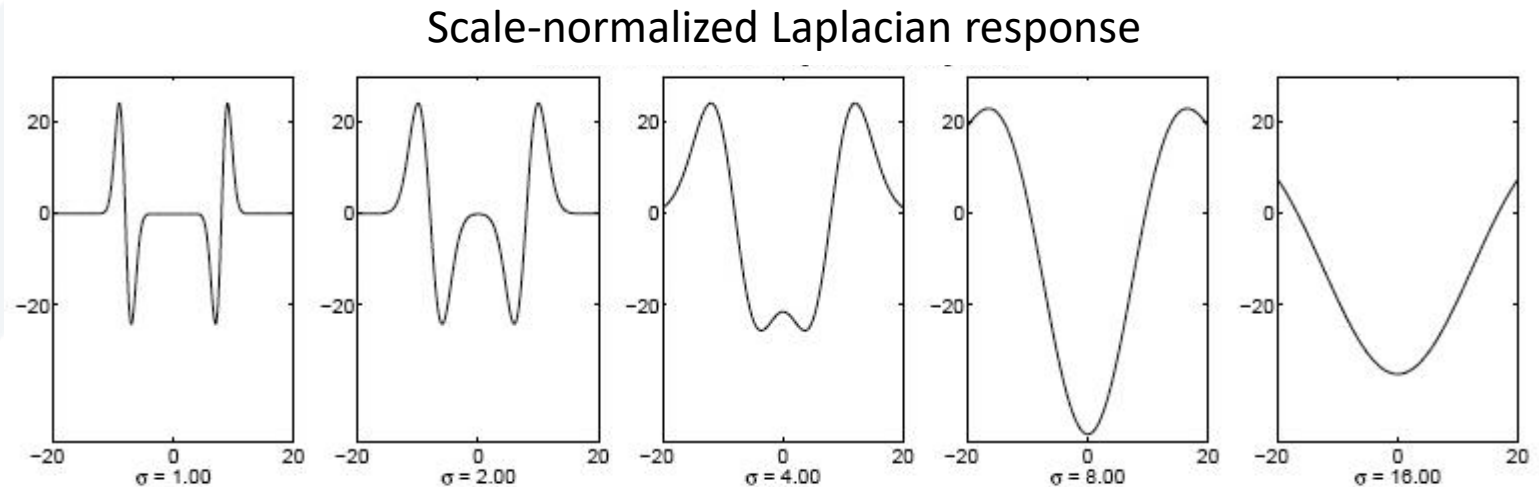
# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## كشف السمات الثابتة Scale-invariant feature detection

تقل استجابة مرشح  
اللابلاسيان الغوصي بشكل  
أكبر مع زيادة قيمة التنعيم  
(زيادة  $\sigma$ )



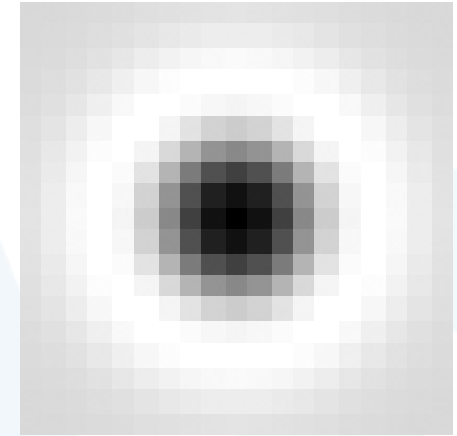
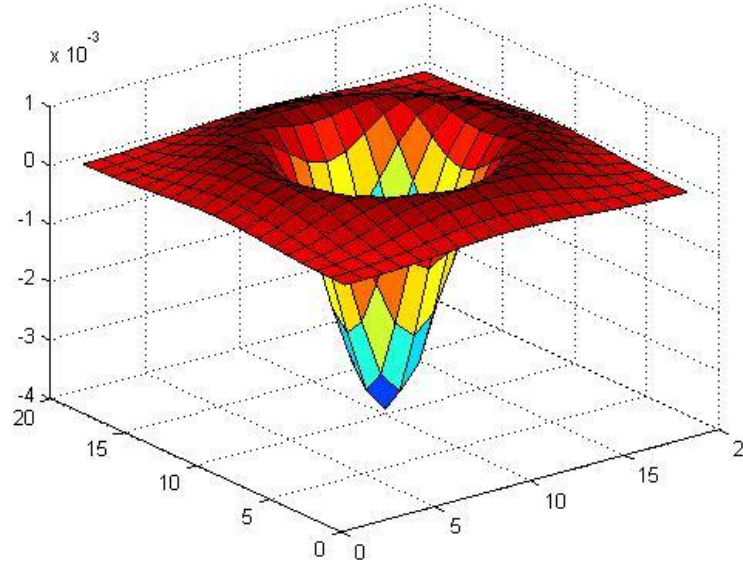
لجعل استجابة مرشح  
اللابلاسيان الغوصي ثابتة  
نضرب المشتق الغوصي  
بقيمة  $\sigma$





# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## كشف السمات الثابتة Scale-invariant feature detection



$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \quad \xrightarrow{\text{بعد الجداء بقيمة } \sigma} \quad \nabla_{\text{norm}}^2 g = \sigma^2 \left( \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$

Scale-normalized Laplacian

# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## كشف السمات الثابتة Scale-invariant feature detection

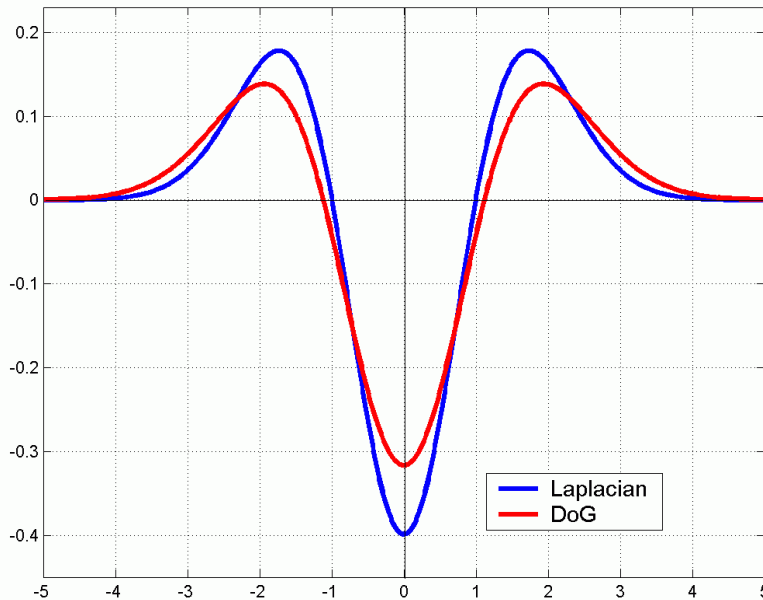
2

- مرشح الفرق الغوسي Derivative of Gaussian
- نرشح الصورة بمرشحي غوص بقيمتي  $\sigma$  مختلفتين ونطرح نواتج التنعيم فنحصل على الفرق الغوسي:

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G.$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G.$$

- $k$  is not necessarily very small in practice



# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## خطوات خوارزمية كشف السمات الثابتة Scale-invariant feature detection

الخطوة الأولى: كشف النقاط العظمى (السمات الأولية)

### -Scale space extrema detection

1. ترشيح الصورة باستخدام مرشح اللابلاسيان الغوسي الثابت scale-normalized Laplacian of Gaussian وذلك في عدة أحجام مختلفة، كما يمكن استخدام مرشح الفرق الغوسي Derivative of Gaussian بدلاً من اللابلاسيان.

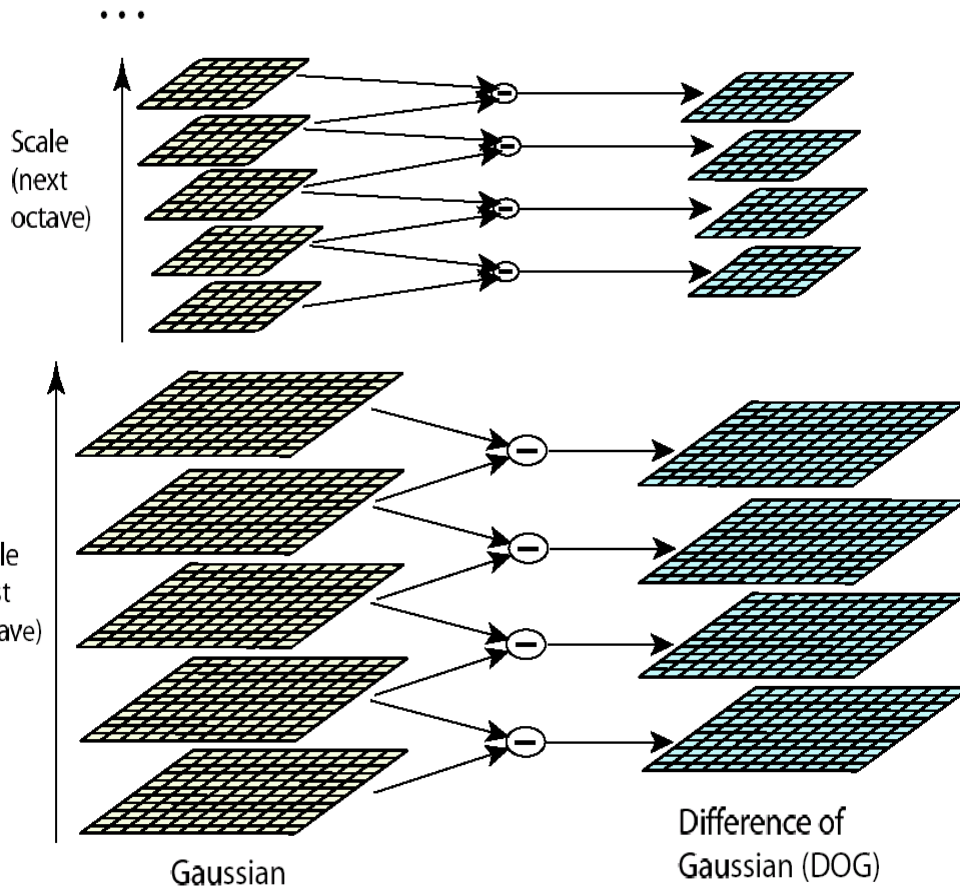
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

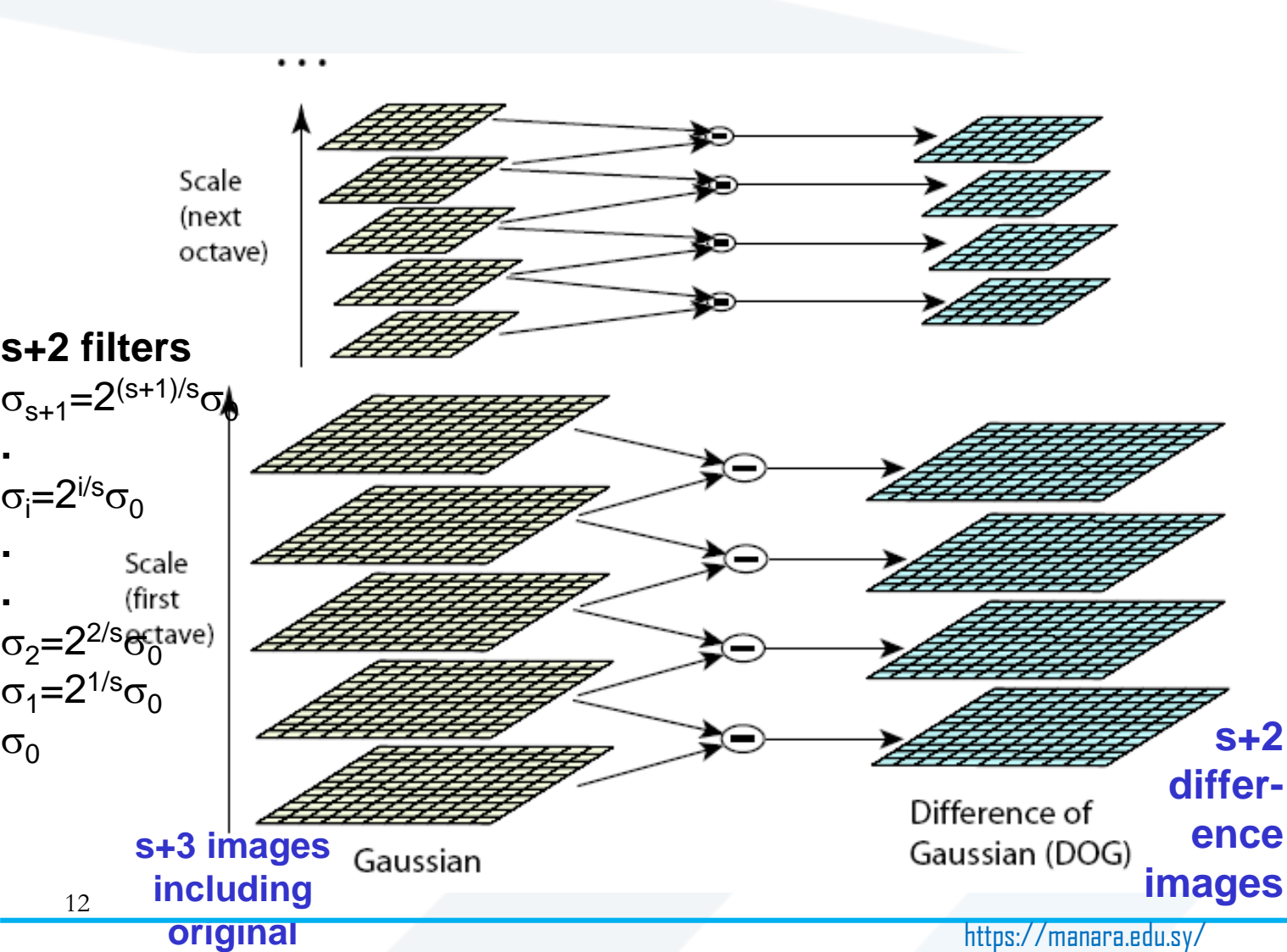
الهدف من هذه الخطوة جعل السمات مستقلة عن الحجم

Scale Invariant



# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## خطوات خوارزمية كشف السمات الثابتة Scale-invariant feature detection



- فضاء التحجيم مقسم إلى أوكتاف الأول يستخدم قيمة  $\sigma$  أما الثاني يستخدم  $2\sigma$  وهكذا.
- في كل أوكتاف يطبق المرشح الغوسي على الصورة لإنتاج  $s+3$  صورة تختلف بقيمة  $\sigma$
- كما يتم في كل أوكتاف طرح كل صورتين منعمتين من بعضهما لإنتاج الفرق الغوسي سنحصل على  $s+2$  فرق غوسي.
- بعد الانتهاء من الأوكتاف الأول يتم تقليل حجم الصورة بمقدار الضعف وتكرر العمليات نفسها للأوكتاف التالي.

# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## خطوات خوارزمية كشف السمات الثابتة Scale-invariant feature detection

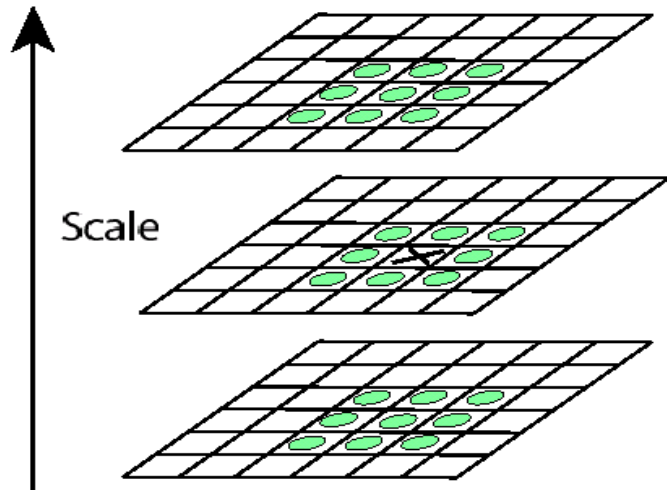
### الخطوة الثانية: تحديد النقاط المهمة Keypoint localization

1- تحديد القيم العظمى والصغرى Local Maxima, Local Minima لناتج الفرق الغوسي في فضاء الحجم Scale Space.

2- ندرس المجاورات الثمانية لكل نقطة سمات ناتجة من المرحلة الأولى لتحديد إن كان البكسل هو أعظم أو أقل قيمة مقارنة مع مجاوراته الثمانية ثم يقارن مع المجاورات التسعة التي في الصورة التي فوقه، ثم مع الصورة التي تحته، وفي حال كان أعظم قيمة في كل هذه الجوارات نختاره كـ Candidate keypoint وإلا يحذف.

لازالت بعض النقاط غير مهمة ويجب حذفها مثل (النقاط ضعيفة التباين ونقاط الحواف الضعيفة) low-contrast/edge points

**الهدف** من هذه الخطوة جعل السمات ثابتة مع الموقع Location Invariant.



الآن لكل نقطة عظمى أو صغرى  
لدينا موقع وحجم محدد

# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## خطوات خوارزمية كشف السمات الثابتة Scale-invariant feature detection

### الخطوة الثالثة: إعطاء اتجاه لكل نقطة مهمة Assign orientation to each keypoint

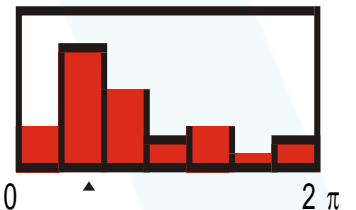
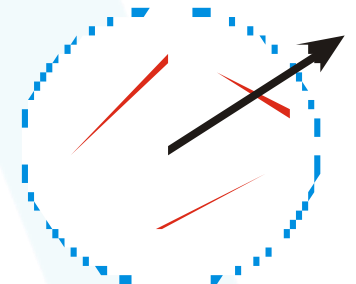
كل نقطة keypoint ناتجة نحدد اتجاهها Orientation وبالتالي تصبح مستقلة عن الدوران Invariant to rotation. يحسب الاتجاه من خلال حساب زاوية التدرج لمرشح اللابلاسيان الغوصي (أو مرشح الفرق الغوصي) لكل نقطة وذلك ضمن Scale محدد وفق الآتي:

1- نحسب طويلة التدرج  $m$  وزوايا التدرج  $\theta$  لكل نقطة Keypoint باستخدام نافذة (جوار محدد) ونبني Orientation Histogram مؤلف من 36 قيمة (كل قيمة منه تغطي مجال زاوية بمقدار عشر درجات). تضرب كل قيمة بكسل في جوار Keypoint بقيمة التدرج الموافق لها في  $m$  ونعد الزوايا الموجودة في الجوار لبناء الهيستوغرام.

2- يتم حساب Peaks أو قمم الهيستوغرام الأعلى بنسبة 80% وما فوق وتمثل الزوايا الأكثر تكراراً للبكسل المدروس (Dominant Orientations) وبالتالي نحصل على عدة Orientations لنفس البكسل ما يجعله مستقل عن الدوران.

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y)))$$



**الهدف** من هذه الخطوة جعل السمات ثابتة مع الدوران Orientation Invariant.



# كشف السمات المستقل عن التغيرات Invariance Feature Detection

## خطوات خوارزمية كشف السمات الثابتة Scale-invariant feature detection

### الخطوة الرابعة: تشكيل السمات Feature Descriptor

دخل هذه المرحلة هو Keypoints مستقلة الحجم والموقع والدوران، والآن بقي علينا حساب شعاع السمات النهائي.

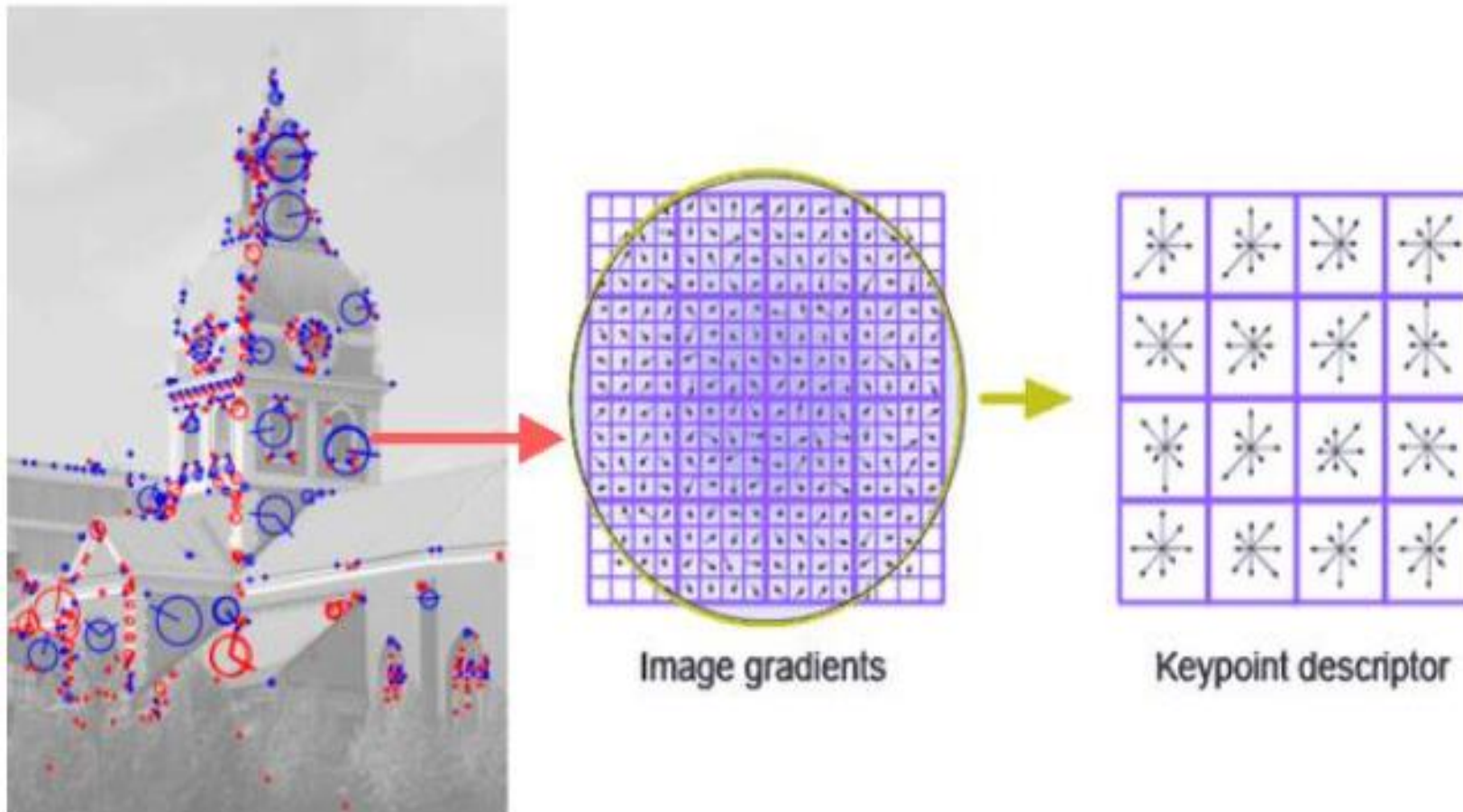
نقوم بتشكيل مصفوفة من ناتج orientation histogram كل مصفوفة تمثل هيستوغرام فرعي بحجم 4x4 كل خلية تتضمن 8 اتجاهات بمعدل 128 اتجاه.

8 orientations x 4x4 histogram array = 128 dimensions

يتم بعدها تطبيق عملية Normalization على الشعاع الناتج (128 قيمة) للتخلص من تغيرات الإضاءة.

كذلك يتم تطبيق عتبة 0.2 على السمات الناتجة ثم يتم تطبيق Normalization مرة أخرى.

تجعل هذه العمليات السمات مستقلة عن تغيرات الإضاءة Illumination-variation



# كشف السمات المستقل عن التغيرات Invariance Feature Detection

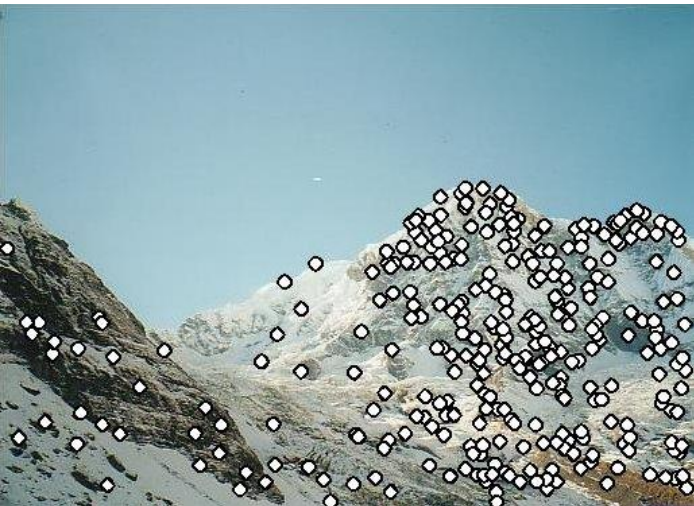
## خطوات خوارزمية كشف السمات الثابتة Scale-invariant feature detection

### الخطوة الخامسة: مطابقة السمات Feature Matching (التطبيق العملي للخوارزمية)

نطبق الخطوات من 1 إلى 4 على الصورة الأولى للحصول على سماتها Feature Descriptors  
نطبق الخطوات من 1 إلى 4 على الصورة الثانية للحصول على سماتها Feature Descriptors  
نحسب المسافة الإقليدية بين سمات A و B لنطابق السمات من A إلى أقرب سمات مقابلة لها في B.

A

B





# كشف السمات المستقل عن التغيرات Invariance Feature Detection

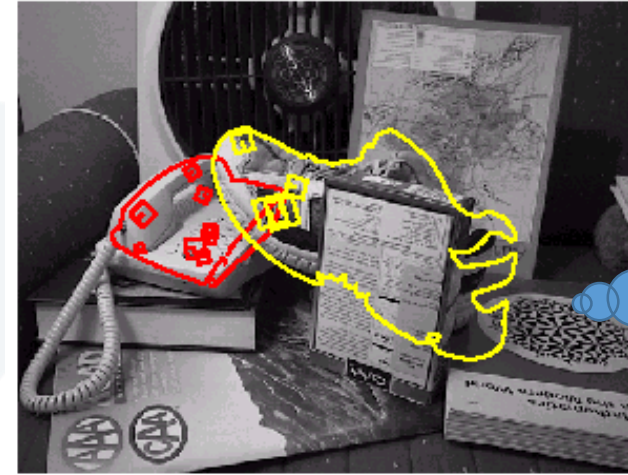
## خطوات خوارزمية كشف السمات الثابتة Scale-invariant feature detection



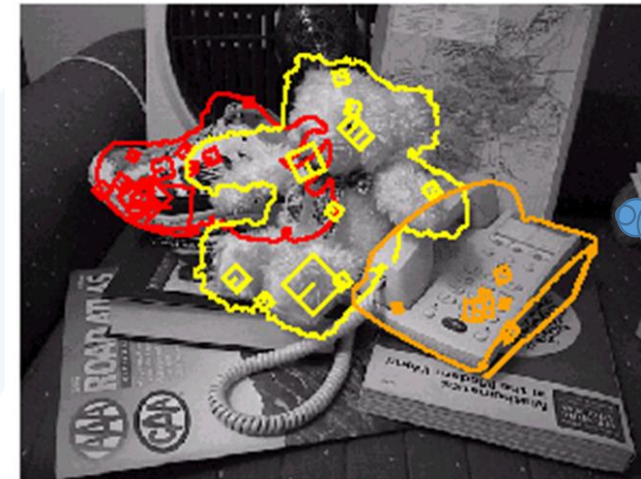
البحث عن  
سمات أشكال  
ذات سمات  
محددة في  
الصور



سمات الشكل  
واضحة



سمات الشكل  
مغطاة بأشكال  
أخرى



كشف عدة  
أشكال

تطبيق

# كشف السمات المستقل عن التغيرات Invariance Feature Detection

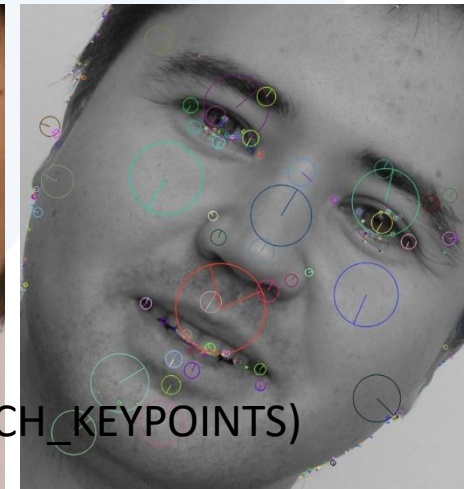
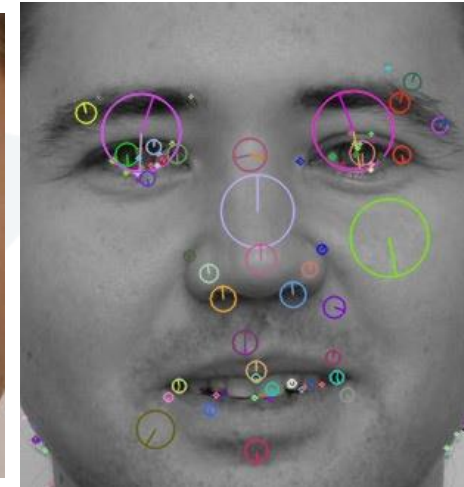
## خطوات خوارزمية كشف السمات الثابتة Scale invariant feature detection

استخلاص  
سمات الوجوه

تطبيق

نفس السمات  
بعد تطبيق  
Zoom in  
على الوجه

نفس السمات  
بعد تطبيق  
rotation  
على الوجه



```
import numpy as np
import cv2 as cv
img = cv.imread('1.jpg')
gray= cv.cvtColor(img,cv.COLOR_BGR2GRAY)
sift = cv.SIFT_create()
kp = sift.detect(gray,None)
img=cv.drawKeypoints(gray,kp,img,flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
cv.imwrite('sift_keypoints.jpg',img)
```



# Template Matching

## مطابقة القوالب

تستخدم لكشف مناطق مخفية ضمن الصور من الصعب  
كشفها بالطرق التقليدية أو العين المجردة.

# Where's Waldo?



Template



Scene

# Template Matching

## مطابقة القوالب

الدخل: الصورة الأصلية – القالب.

الخرج: مكان القالب في الصورة.

### خطوات مطابقة القوالب:

إجراء عملية ارتباط خطي Correlation بين الصورة والقالب (استخدام القالب كمرشح).

عملية المطابقة Match تتم باستخدام طريقة القيم العظمى Local Maximum:

البحث ضمن الصورة التي نتجت عن Correlation عن البكسلات التي تتجاوز قيمة عتبة وتكون هي المنطقة المطلوبة.

**مشكلة الطريقة:** في حال وجود منطقة ذات إضاءة عالية في الصورة نحصل دوماً على نتيجة خاطئة لأن هذه المناطق

تمثل Local Maximum Region.

**الحل:** إجراء عملية Normalization وفق العلاقة التالية

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

تضمن هذه العملية أنّ القيم العظمى والصغرى في الصورة ستعامل بذات الطريقة.

# Template Matching

## مطابقة القوالب

طرق أخرى لمطابقة القوالب:

Square Difference TM

تعتمد على حساب مربع الفرق بين عناصر المرشح (القالب) وعناصر الصورة (مكان الترشيح)

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$$

Square Difference Normalized TM

هي ذات الطريقة السابقة مع إضافة فكرة التطبيع

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

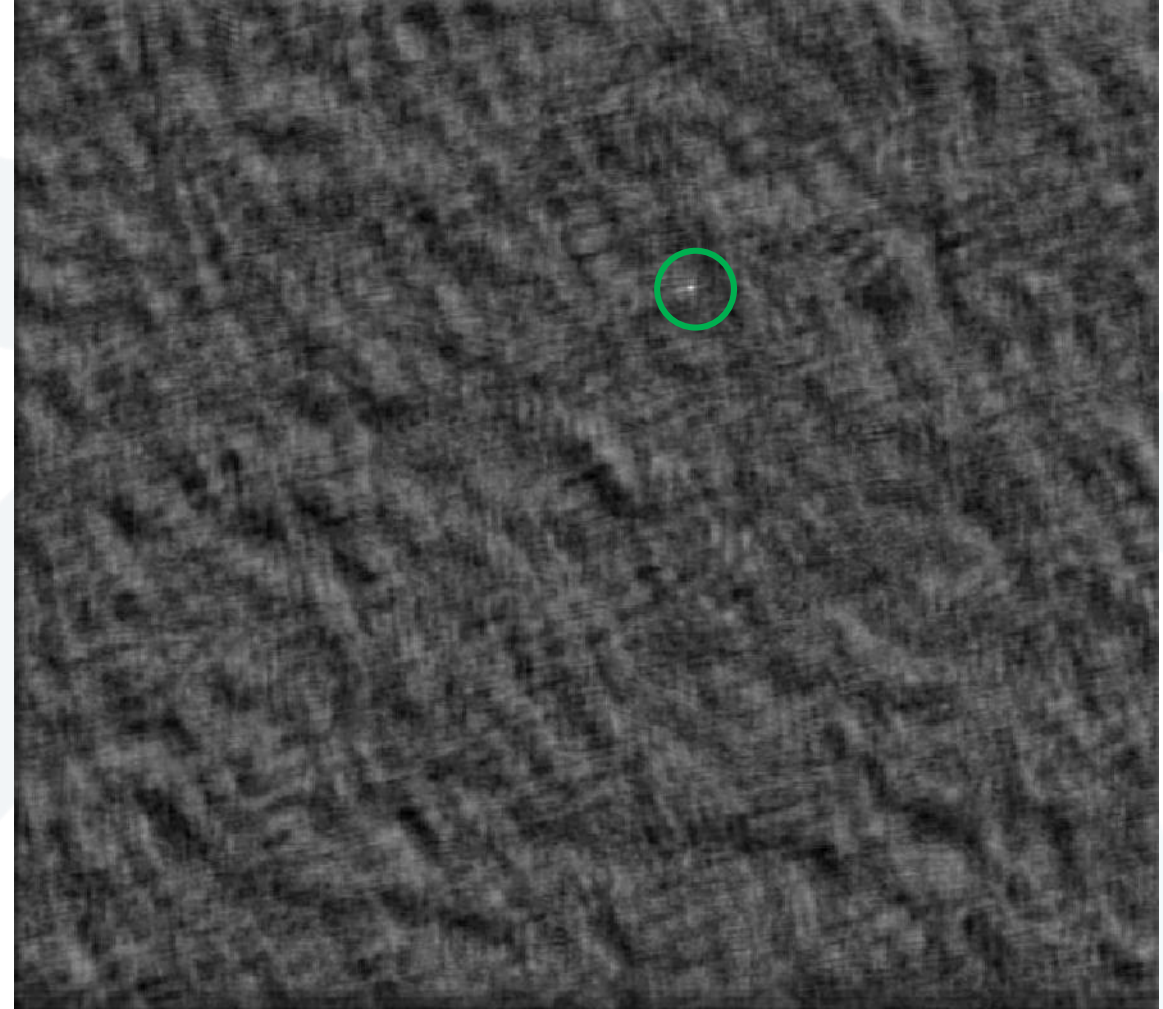


# Template Matching

## مطابقة القوالب



**Detected template**



**Correlation map**



# Template Matching

## مطابقة القوالب

Matching Result



Detected Point



Correlation

تفشل طريقة Correlation أحياناً

Matching Result



Detected Point



Normalized  
Correlation

الحل بتطبيق عملية  
normalization

# Template Matching

## مطابقة القوالب

Matching Result



Detected Point



Square of  
Difference

Matching Result



Detected Point



Square of  
Difference  
Normalized



# Template Matching – Opencv Python

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('img1.jpg',0)
img2 = img.copy()
template = cv2.imread('temp.jpg',0)
w, h = template.shape[::-1]
# All the 6 methods for comparison in a list
methods = ['cv2.TM_CCOEFF', 'cv2.TM_CCOEFF_NORMED',
'cv2.TM_CCORR',
        'cv2.TM_CCORR_NORMED', 'cv2.TM_SQDIFF',
'cv2.TM_SQDIFF_NORMED']
for meth in methods:
    img = img2.copy()
    method = eval(meth)
    # Apply template Matching
    res = cv2.matchTemplate(img,template,method)
    min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(res)
```

```
# If the method is TM_SQDIFF or TM_SQDIFF_NORMED,
take minimum
    if method in [cv2.TM_SQDIFF, cv2.TM_SQDIFF_NORMED]:
        top_left = min_loc
    else:
        top_left = max_loc
    bottom_right = (top_left[0] + w, top_left[1] + h)
    cv2.rectangle(img,top_left, bottom_right, 255, 2)
    plt.subplot(121),plt.imshow(res,cmap = 'gray')
    plt.title('Matching Result'), plt.xticks([], plt.yticks([]))
    plt.subplot(122),plt.imshow(img,cmap = 'gray')
    plt.title('Detected Point'), plt.xticks([], plt.yticks([]))
    plt.suptitle(meth)
    plt.show()
```