

Computer Vision Lecture 4

Features- *Part1*

(Edges, Lines, Corners)

Harris - Hough

Dr. Ali Mahmoud Mayya

Computer Science Dept.

AL Manara University, Syria

2024

استخلاص سمات الصورة Feature Extraction

- Feature points are used for:
 - Motion tracking تعقب الحركة
 - Feature matching مطابقة السمات
 - 3D reconstruction بناء الصور ثلاثية الأبعاد
 - Object recognition التعرف على الأنماط
 - Indexing and database retrieval استرداد البيانات وفهرسة البيانات
 - Robot navigation التحكم بمسار روبوت

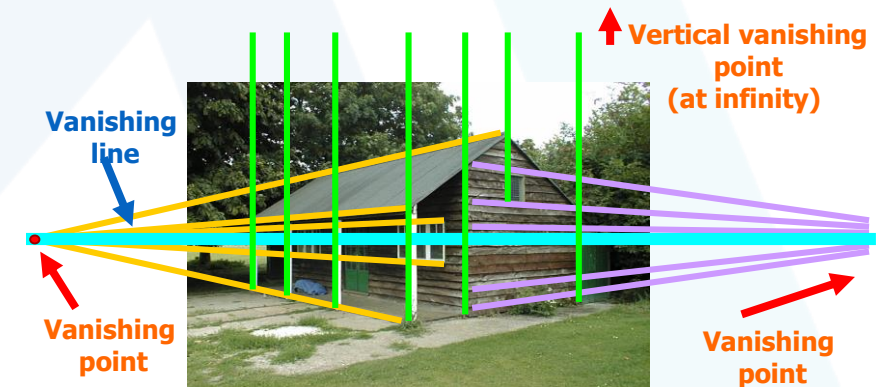
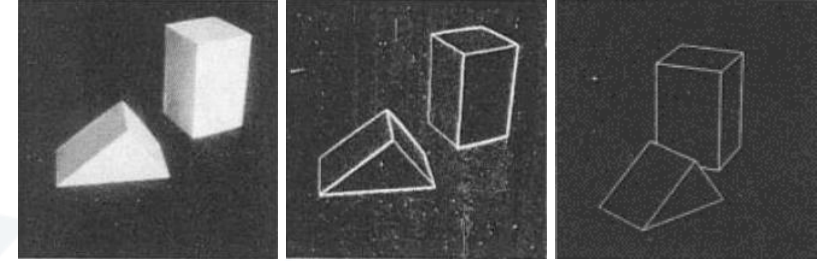
What are features to study

السمات التي سندرسها

- الحواف Edges
- الخطوط Lines والزوايا Corners

Why Edge detection

- تساعد في بناء الشكل ثلاثي الأبعاد للصورة Cues for 3D shape
- تجميع البكسلات ضمن أجزاء متشابهة Group pixels into objects or parts
- الهندسة العكسية والمناظير Recover geometry and viewpoint



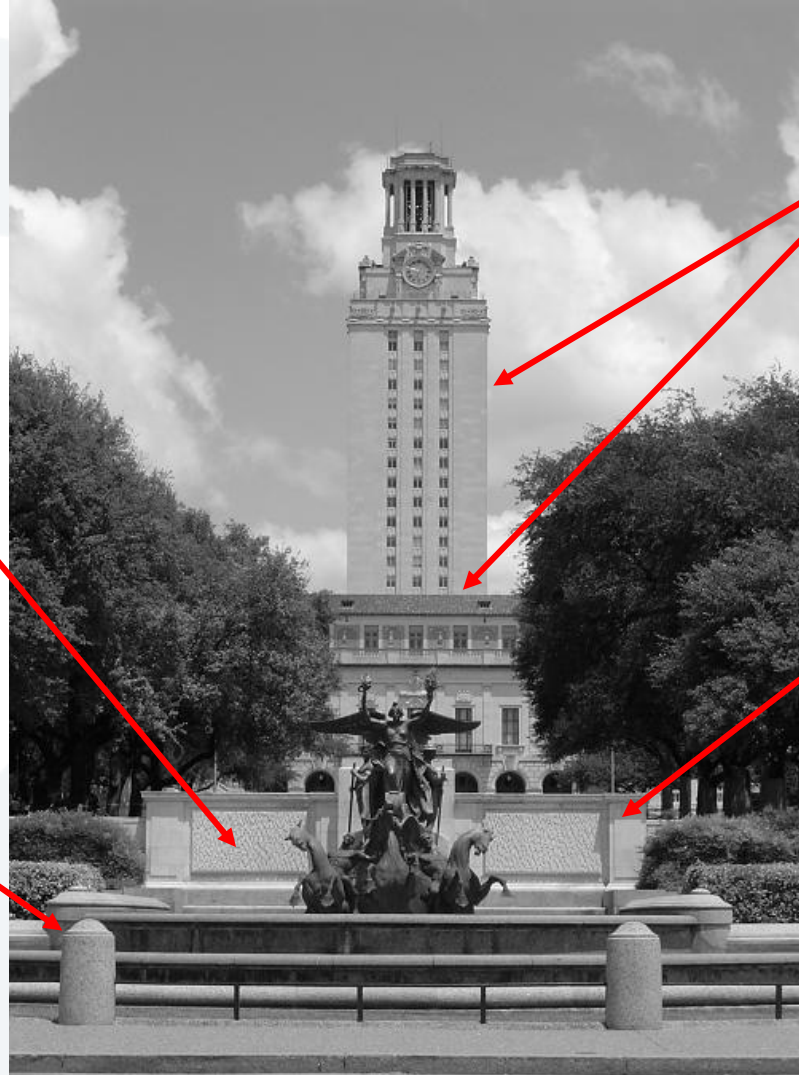
What causes an edge?

Reflectance change:
appearance
information, texture

تغير في انعكاس الضوء بتغير
طبيعية الجسم

Change in surface
orientation: shape

تغير في الشكل



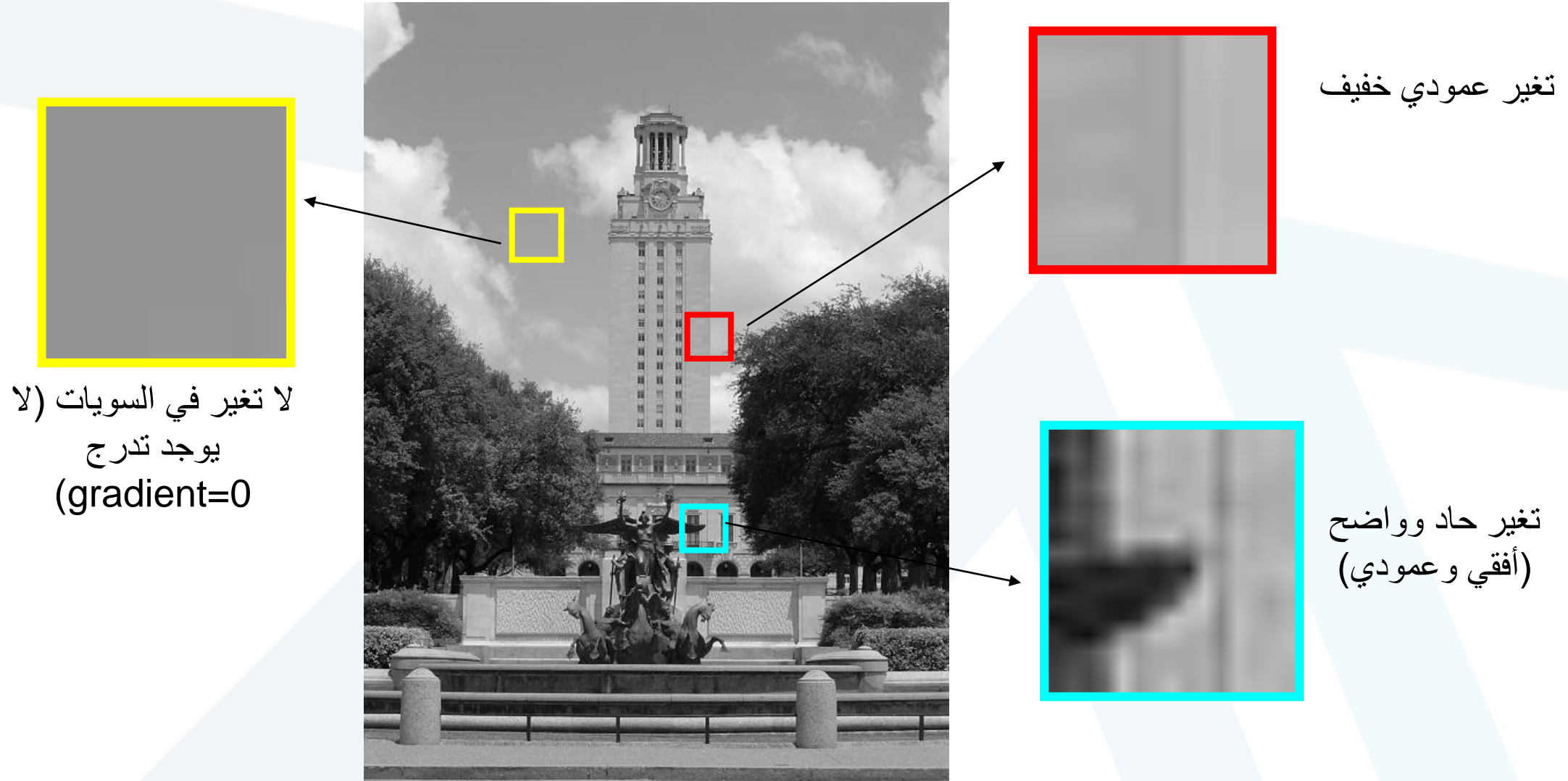
Depth discontinuity:
object boundary

تغير العمق

Cast shadows

Slide credit:
Kristen Grauman

Edges/gradients and invariance

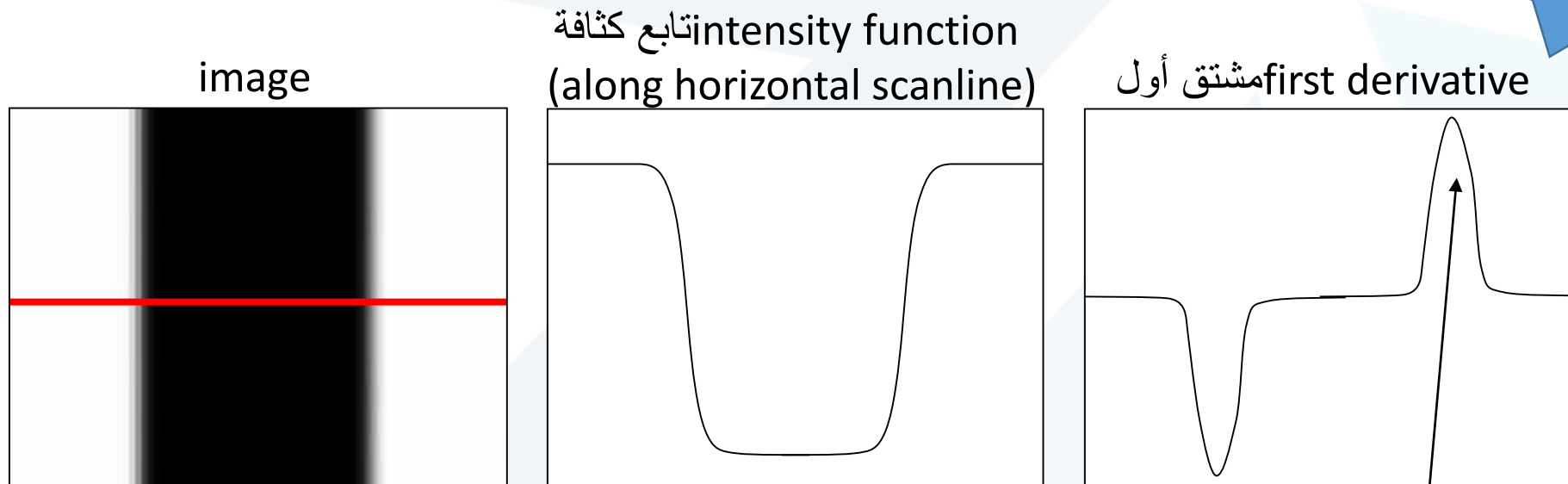


Slide credit:
Kristen Grauman

مبدأ كشف الحواف (الاشتقاق) Edge detection Principle

- An edge is a place of rapid change in the image intensity function

تكون الحافة في
مكان التغيرات
الكبيرة في السويات
الرمادية للصورة



edges correspond to extrema of derivative

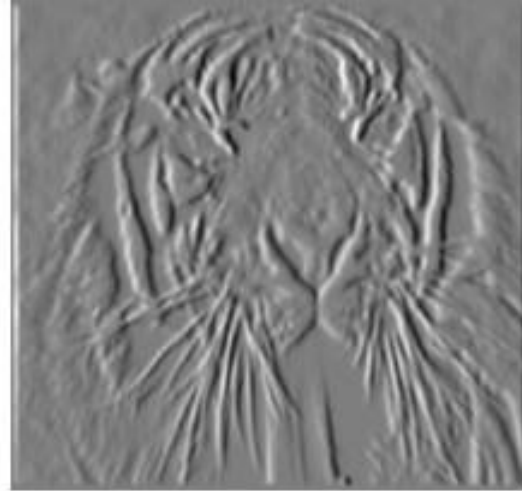
الحواف تتوافق مع القيم الحدية للمشتق

مثال على صورة



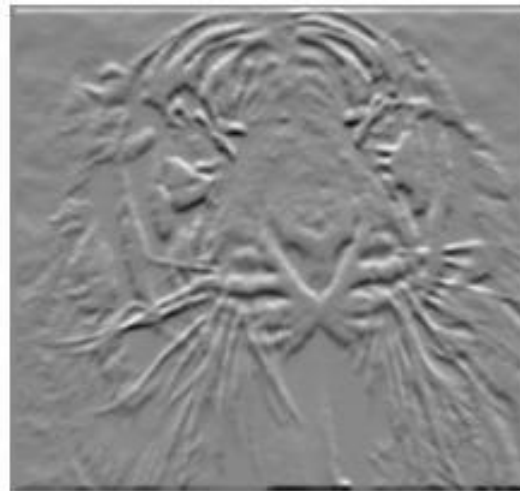
$$\frac{\partial f(x, y)}{\partial x}$$

-1	1
----	---



$$\frac{\partial f(x, y)}{\partial y}$$

-1	?	1
1	or	-1



التغير الأفقي ينتج
حافة عمودية
التغير العمودي ينتج
حافة أفقية

أي
المرشحين
هو
الصحيح؟

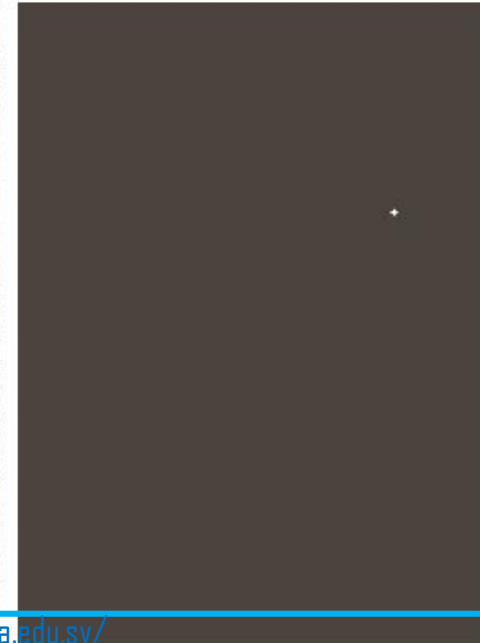
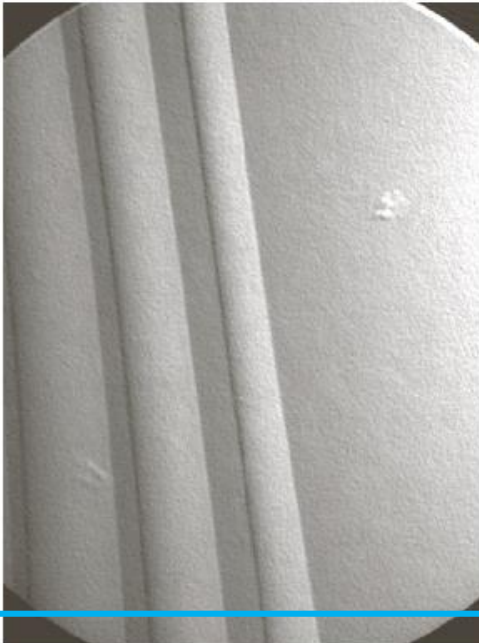
Edge detection Filters

- كشف النقطة Point
- كشف الخطوط Lines
- كشف الحواف Edges
- كشف زوايا Corner

Edge detection Filters

1	1	1
1	-8	1
1	1	1

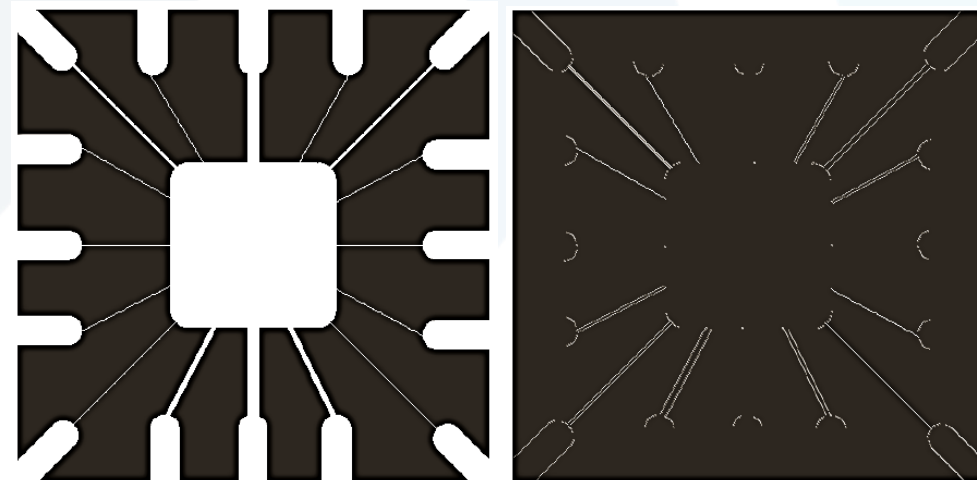
لكشف نقطة معزولة: نستخدم مرشح
اللابلاسيان



Edge detection Filters

FIGURE 10.3 Line masks.

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
Horizontal			+45°			Vertical			-45°		



45

Edge detection Filters

• كواشف الحواف

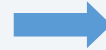
Roberts cross-gradient operators



-1	0	0	-1
0	1	1	0

Roberts

Prewitt operators



-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Prewitt

Sobel operators



-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Sobel

Edge detection Filters

• كواشف الحواف

Prewitt masks for
detecting diagonal edges →

0	1	1	-1	-1	0
-1	0	1	-1	0	1
-1	-1	0	0	1	1

Prewitt

Sobel masks for
detecting diagonal edges →

0	1	2	-2	-1	0
-1	0	1	-1	0	1
-2	-1	0	0	1	2

Sobel

a	b
c	d

FIGURE 10.9 Prewitt and Sobel masks for detecting diagonal edges.

Edge detection Filters

- مثال عن كشف الحواف الأفقية والعمودية وطويلة التدرج باستخدام سوبل

a b
c d

FIGURE 10.10

(a) Original image. (b) $|G_x|$, component of the gradient in the x -direction.

(c) $|G_y|$, component in the y -direction.

(d) Gradient image, $|G_x| + |G_y|$.



Edge detection Filters

- مثال عن كشف الحواف المائلة باستخدام سوبل



a b

FIGURE 10.12
Diagonal edge detection.

(a) Result of using the mask in Fig. 10.9(c).

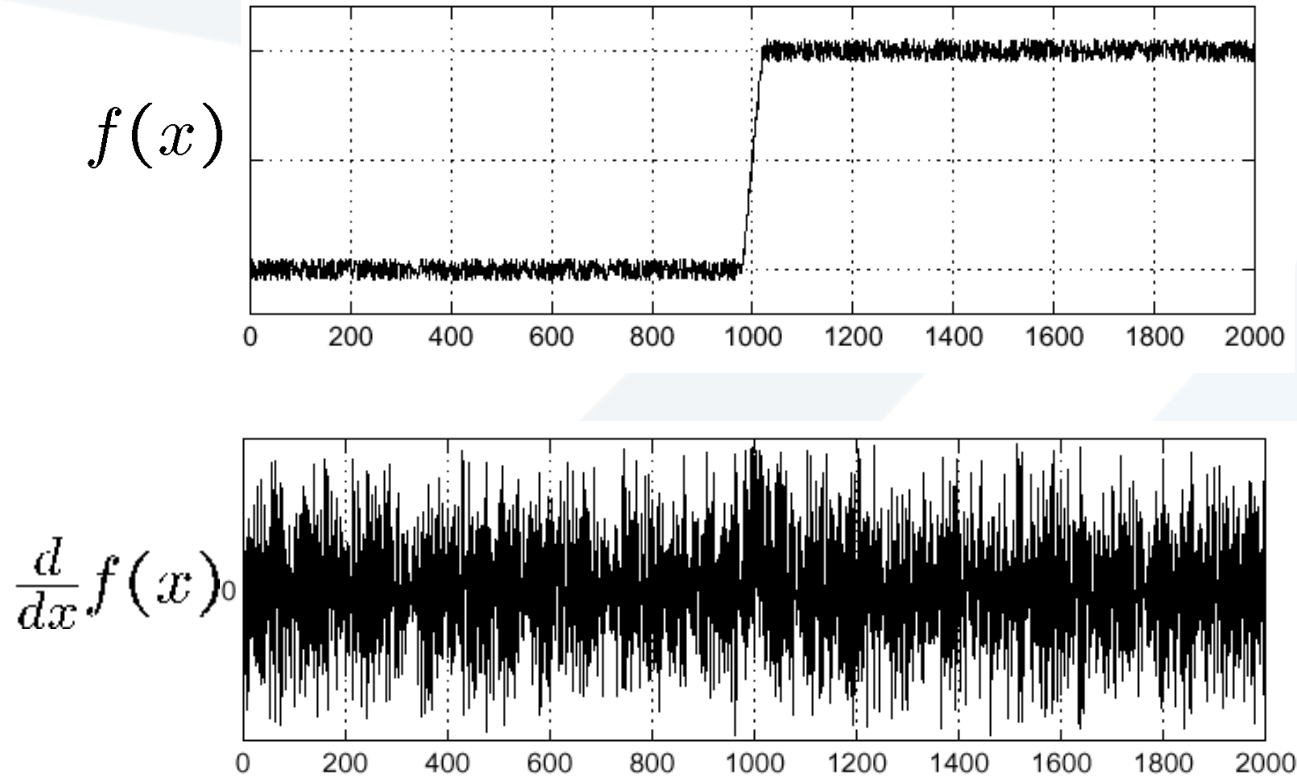
(b) Result of using the mask in Fig. 10.9(d). The input in both cases was Fig. 10.11(a).

0	1	2
-1	0	1
-2	-1	0

-2	-1	0
-1	0	1
0	1	2

تأثير الضجيج على الحواف ! Edge are affected by noise !

• يظهر الضجيج في نتيجة كشف الحواف، لماذا؟

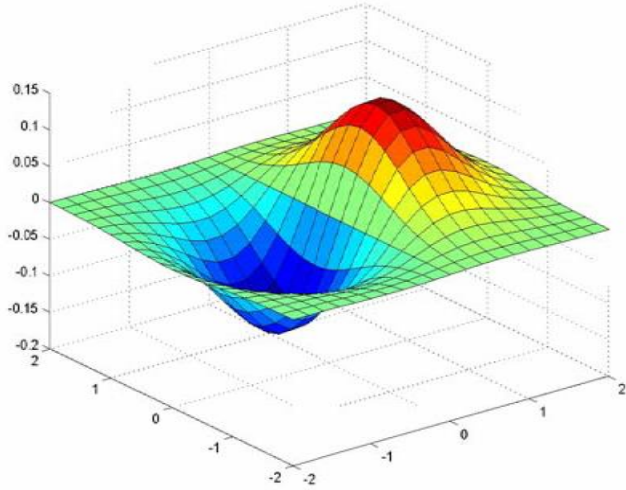


Where is the edge?

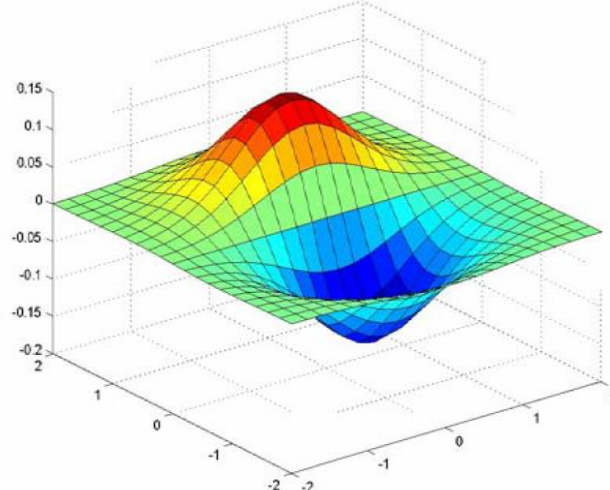
Source: S. Seitz

مرشحات كشف الحواف بدون تأثير الضجيج

$g^* [1 \ 0 \ -1]^T$



$g^* [1 \ 0 \ -1]^T$



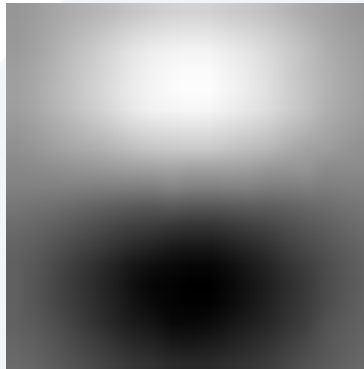
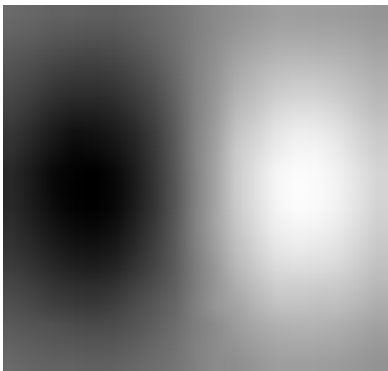
• أولاً Derivative of Gaussian filter

• مرشح المشتق الغوسي:

• تنعيم أولاً ثم اشتقاق

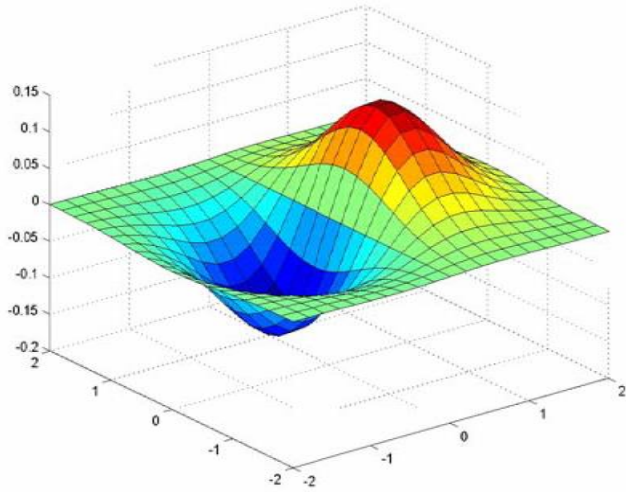
• التنعيم يخفف الضجيج

• الاشتقاق يكشف الحواف

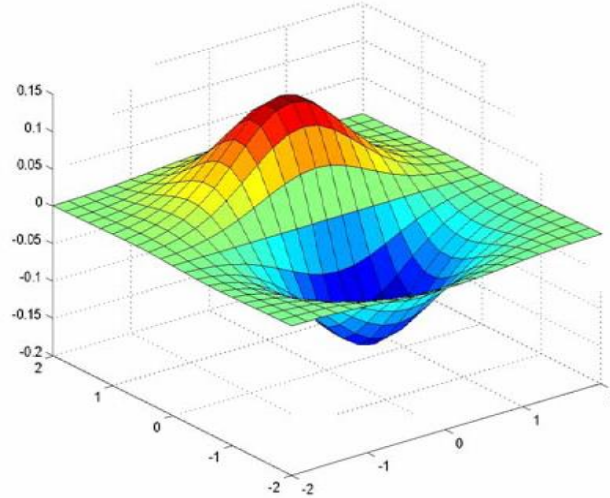


مرشحات كشف الحواف بدون تأثير الضجيج

$$g^* [1 \ 0 \ -1]^T$$



$$g^* [1 \ 0 \ -1]^T$$



Derivative of Gaussian filter •

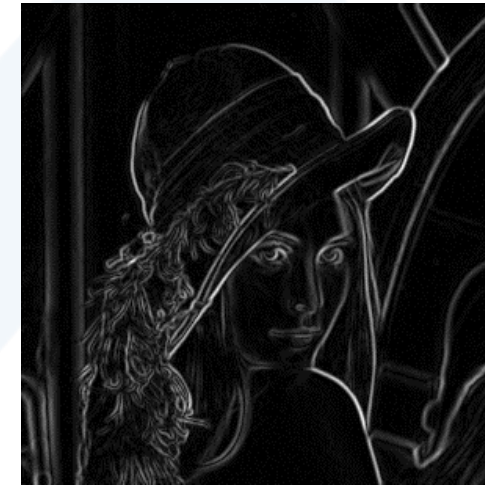
• مرشح المشتق الغوسي



X-Derivative of Gaussian



Y-Derivative of Gaussian

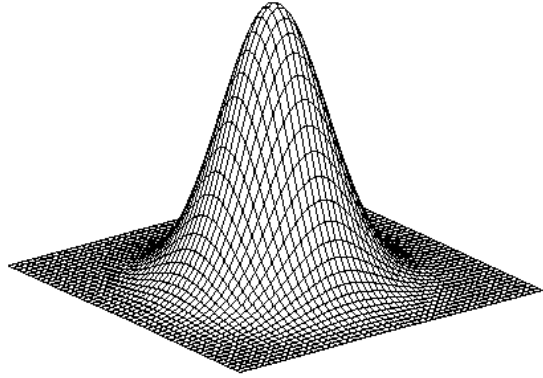


Gradient Magnitude

مرشحات كشف الحواف بدون تأثير الضجيج

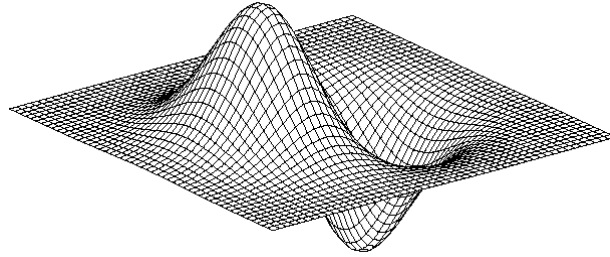
• **ثانياً** Laplacian of Gaussian filter

• مرشح اللابلاسيان الغوسي:



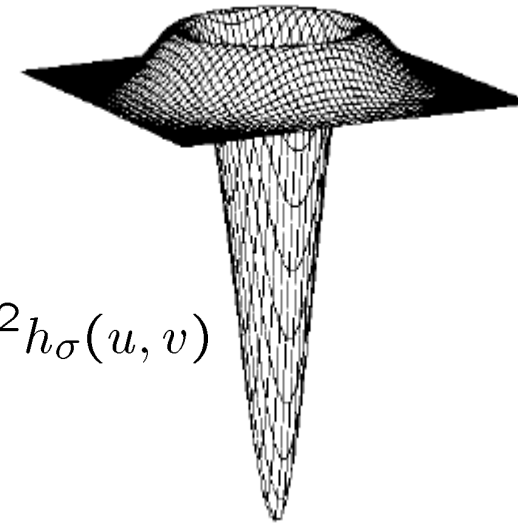
Gaussian

$$h_{\sigma}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

$$\frac{\partial}{\partial x} h_{\sigma}(u, v)$$



Laplacian of Gaussian

$$\nabla^2 h_{\sigma}(u, v)$$

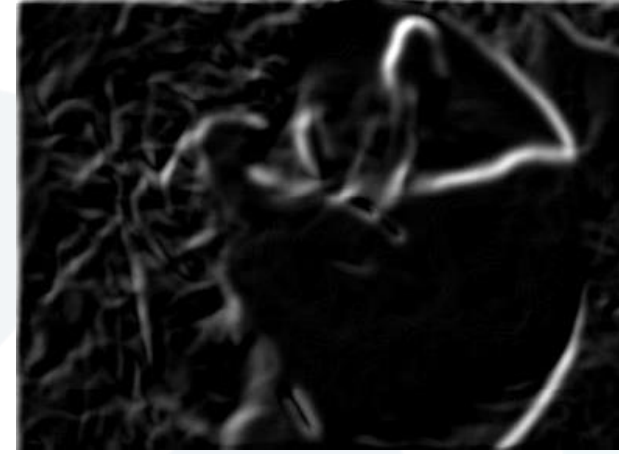
- ∇^2 is the Laplacian operator:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Edge detection Filters



$\sigma = 1$ pixel



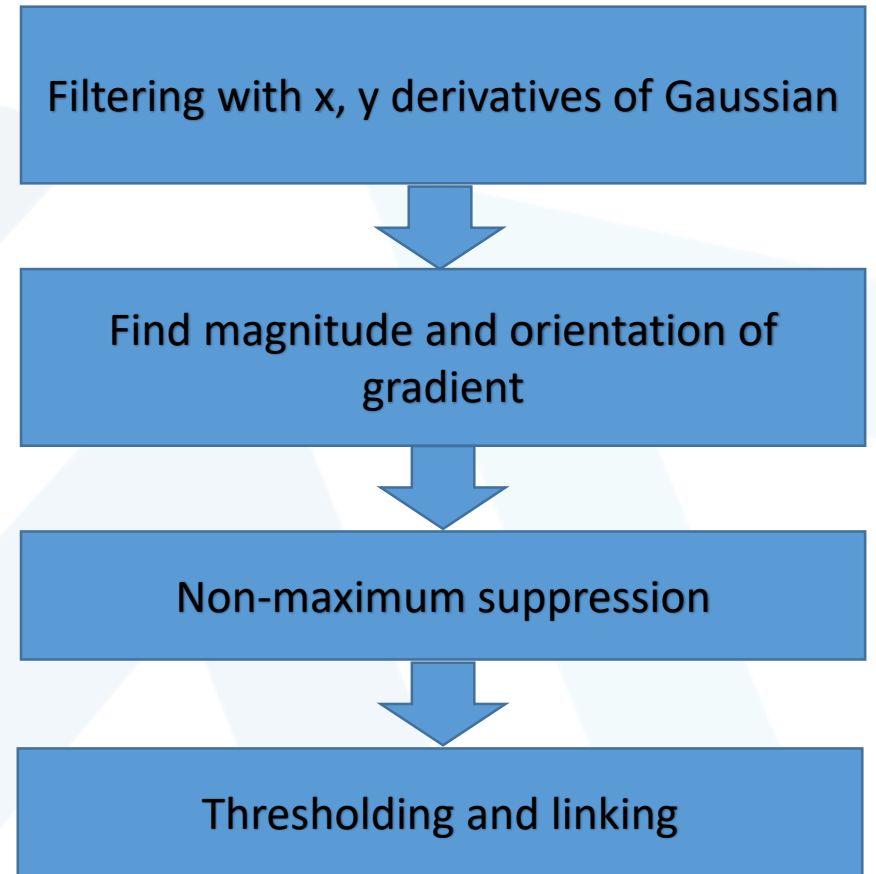
$\sigma = 3$ pixels

تختلف الحواف باختلاف معامل الانحراف المعياري لمرشح غوص
زيادة σ تعني تنعيم التغيرات الرمادية أكثر --> الحواف الناتجة أقل
تقليل σ تعني تقليل تنعيم التغيرات الرمادية أكثر --> الحواف الناتجة أكثر
ما الأفضل؟ حسب الحالة

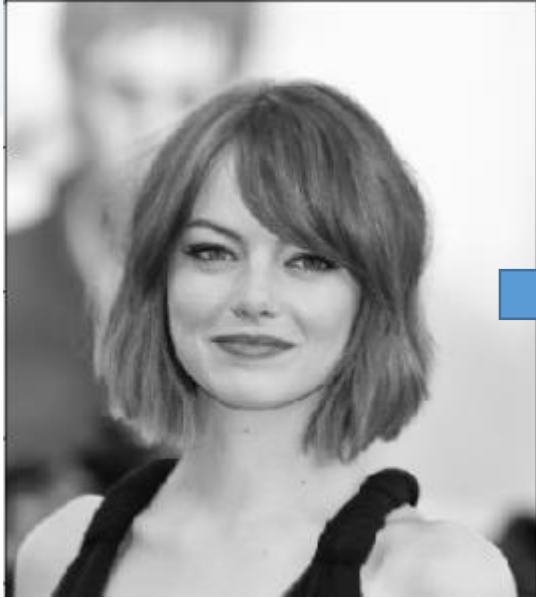
(هل تتضمن الصورة ضجيج ؟ هل نريد حواف قوية فقط؟)

Canny Edge detector

1. **Filter** image with x, y derivatives of Gaussian
ترشيح الصورة باستخدام المشتق الغوسي
2. Find **magnitude** and **orientation** of gradient
إيجاد طولية التدرج وزاوية التدرج
3. **Non-maximum suppression**: إيجاد القيم العظمى
Thin multi-pixel wide “ridges” down to single pixel width
ترقيق البكسلات لحجم بكسل واحد فقط
4. **Thresholding and linking** (hysteresis): التعتيب والربط
Define two thresholds: low and high نحدد عتبتين منخفضة وعالية ومنخفضة
Use the high threshold to start edge curves and the low threshold to continue them
تستخدم العتبة العالية للبدء بمنحنيات الحواف والمنخفضة لجعلها مستمرة



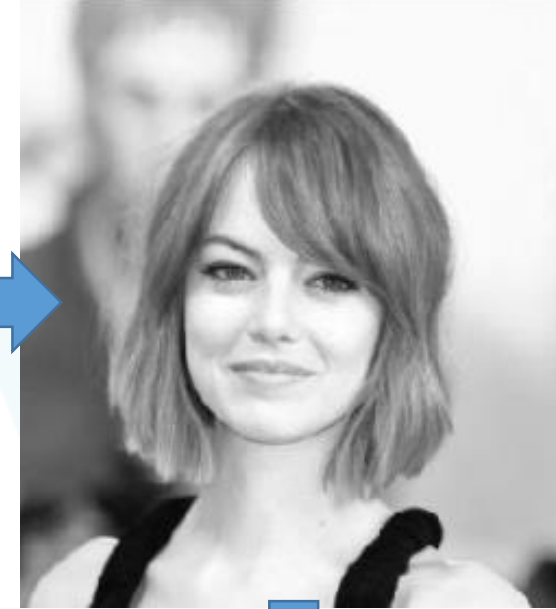
Canny Edge detector Step^{1&2}: Derivative of Gaussian



تنعيم غوسي

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i - (k+1))^2 + (j - (k+1))^2}{2\sigma^2}\right); 1 \leq i, j \leq (2k+1)$$

Gaussian filter kernel equation



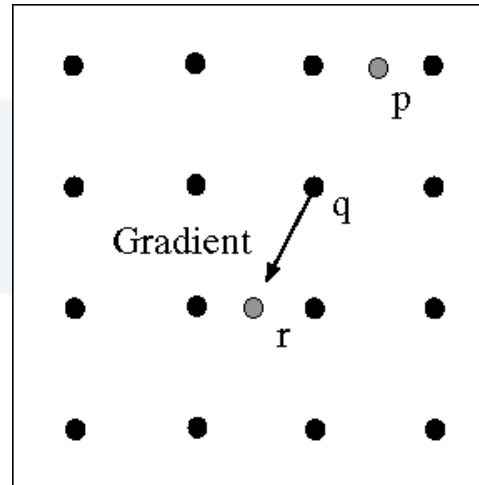
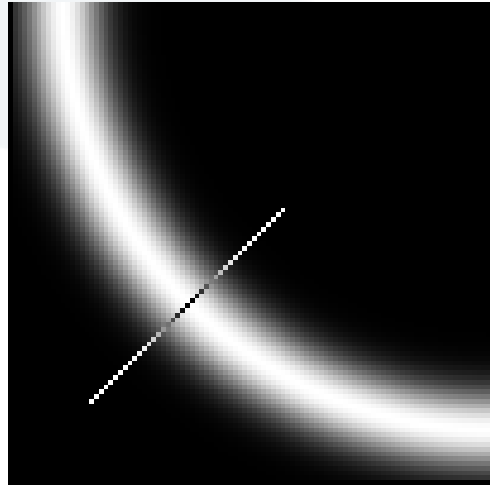
$$|G| = \sqrt{I_x^2 + I_y^2},$$
$$\theta(x, y) = \arctan\left(\frac{I_y}{I_x}\right)$$



طويلة التدرج (كشف حواف باستخدام سوبل)

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

Canny Edge detector Step³: Non Max Suppression

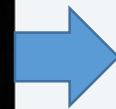


خطوات Non Max Suppression

نأخذ قيم زوايا التدرج المحسوبة في الخطوة 2 وهي (0-45-90)

يتم التحقق من أن البكسل يمثل قيمة عظمى وباتجاه الحافة.

في حال عدم تحقق الشرط لا يعتبر البكسل حافة ويحذف.



Non-max
suppression



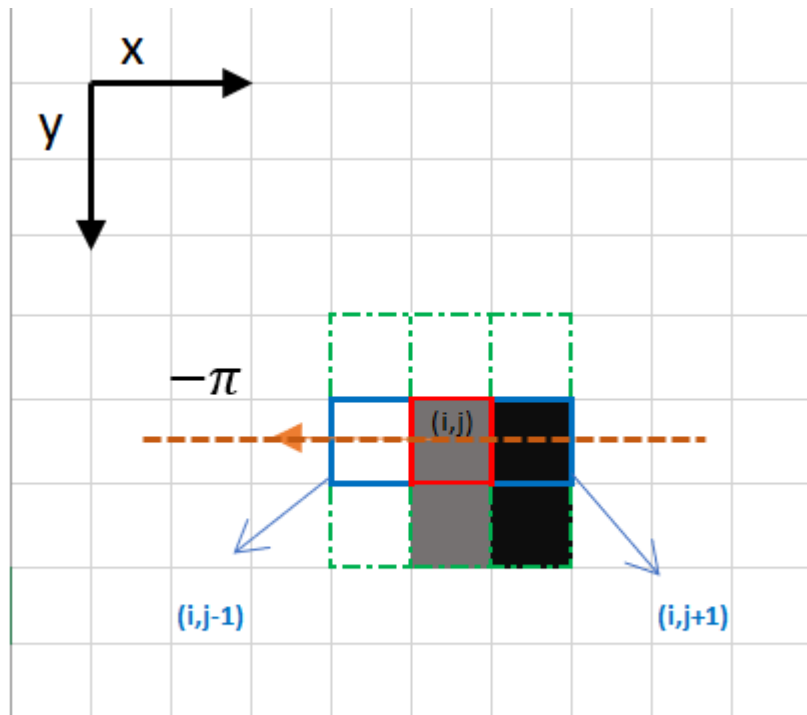
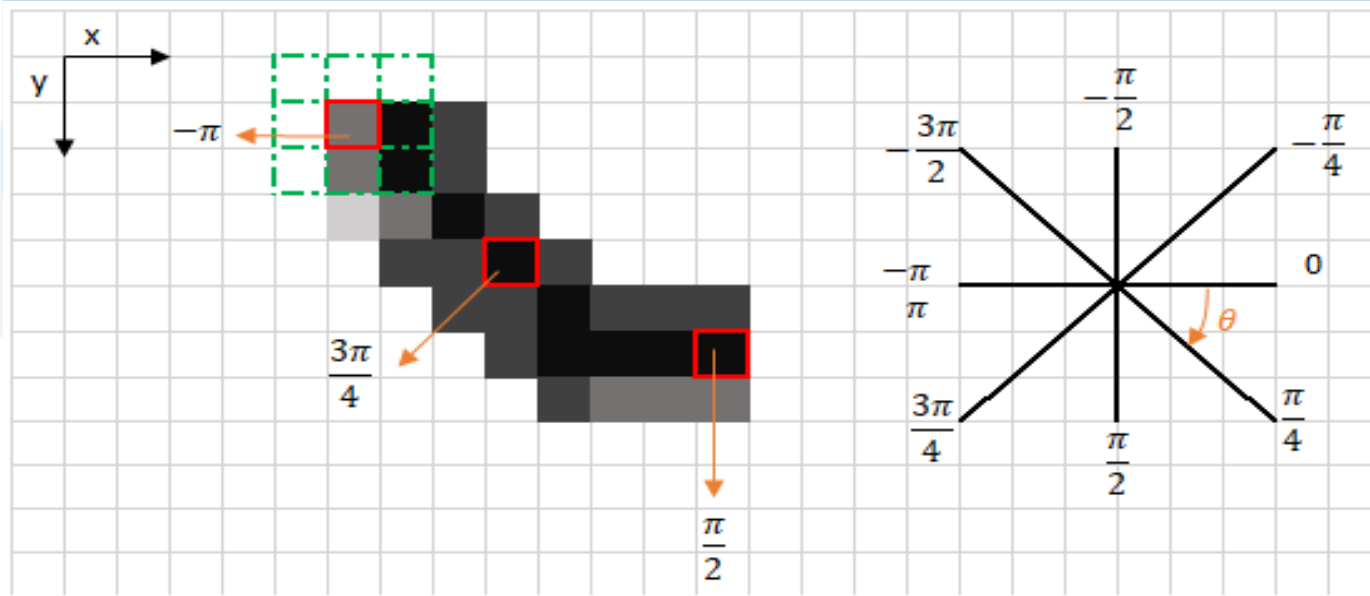
Result of the
non-max
suppression.

Non Max Suppression

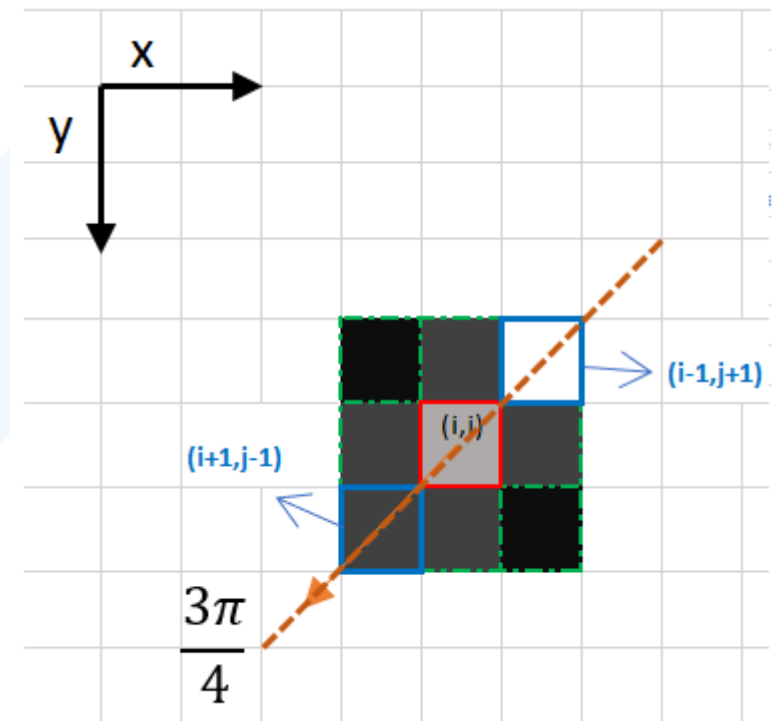
البكسل المدروس (i,j)

زاوية الحافة 45

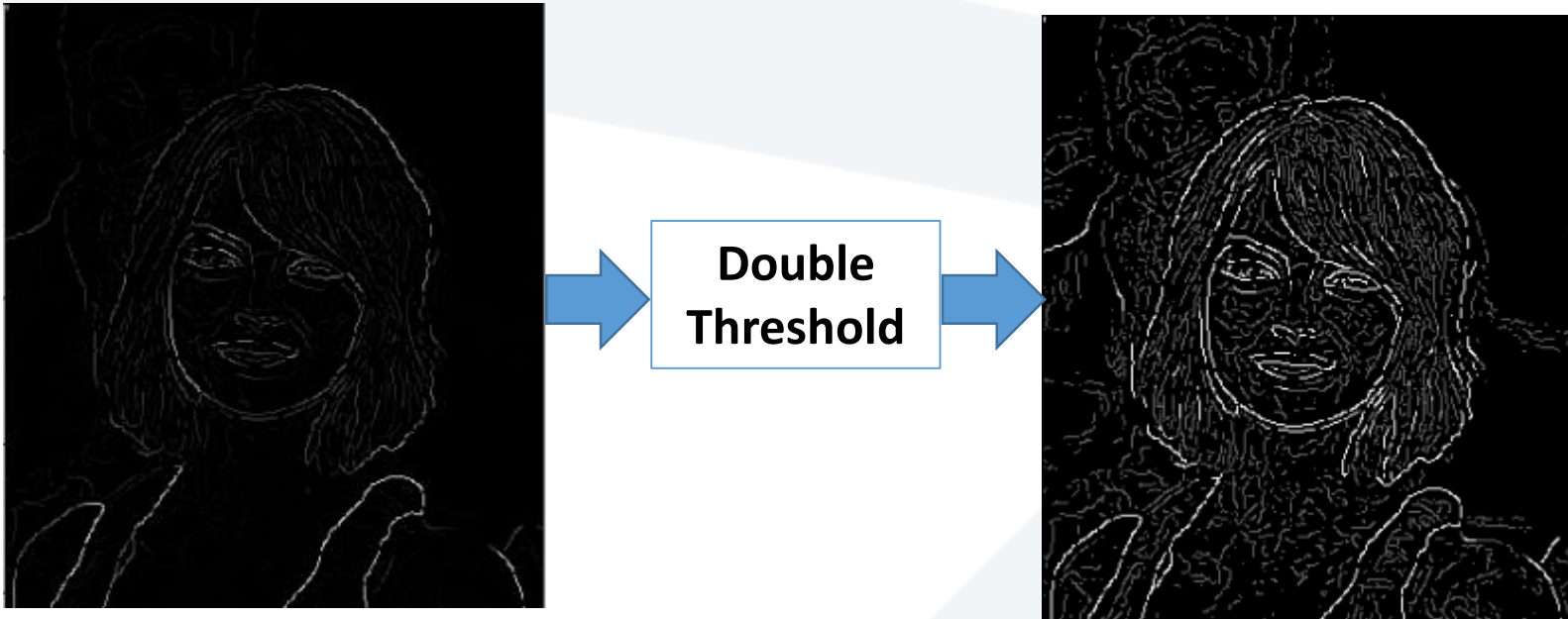
ندرس البكسلين $(i-1, j+1)$ و $(i+1, j-1)$ المجاورين للبكسل (i, j) ، هل يوجد بكسل منهما سويته أعلى من البكسل (i, j) ؟ نعم البكسل $(i-1, j+1)$ ؟
إذاً، نحتفظ بالبكسل $(i-1, j+1)$ ونلغي البكسل (i, j) .



Focus on the upper left corner red box pixel



Canny Edge detector Step⁴: Thresholding and linking (hysteresis)



- **أولاً Double Threshold:** تهدف هذه العملية إلى تحديد 3 أنواع من الحواف: قوية- ضعيفة- غير مرتبطة

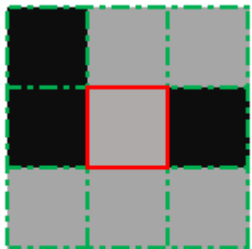
- **البكسلات القوية Strong Pixels** تمتلك سوية رمادية أعلى من العتبة.

- **البكسلات الضعيفة Weak Pixels** هي بكسلات تمتلك سويات أقل من العتبة القوية لكنها أعلى من العتبة الضعيفة.

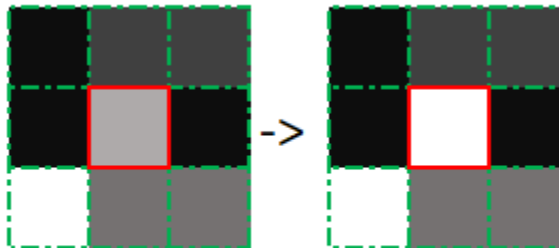
- **البكسلات غير المرتبطة Non-relevant Pixels** هي البكسلات التي تمتلك سويات أقل من العتبة الضعيفة.

- **ثانياً تعقب الحواف Edge Tracking by Hysteresis**

- بالاعتماد على نتيجة المرحلة الأولى، يتم ربط الحواف الضعيفة والقوية معاً.



No strong pixels around



One strong pixel around

Canny Edge detection Example

Effect of σ (Gaussian kernel spread/size)
تأثير تغير σ على الحواف الناتجة



original



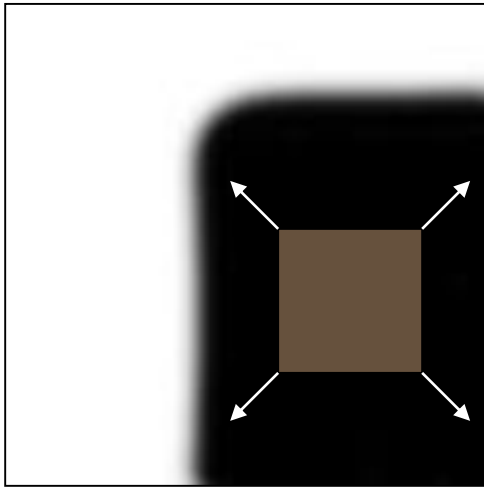
Canny with $\sigma = 1$



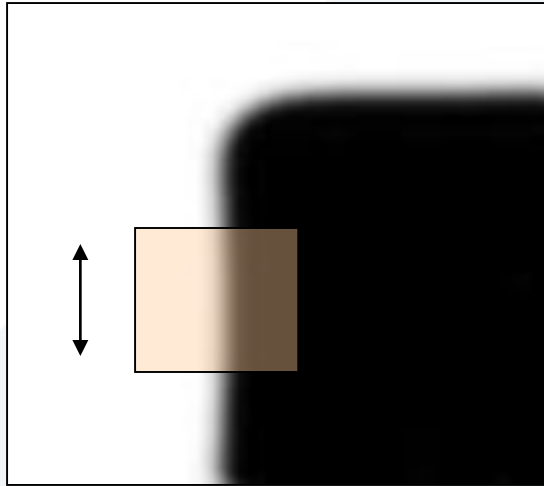
Canny with $\sigma = 2$

كشف الزوايا Corner Detection

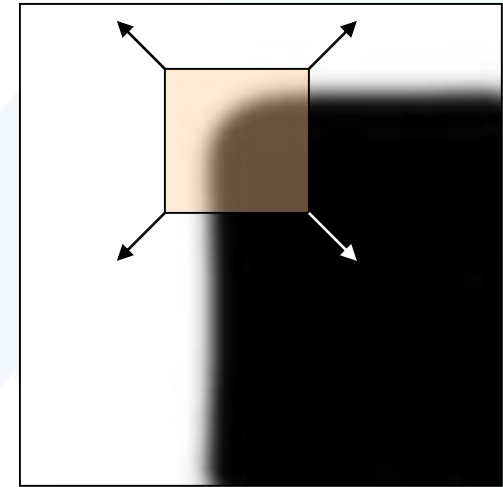
- We should easily recognize the point by looking through a small window
- تستخدم نافذة بأبعاد صغيرة تمرر عبر الصورة لتمييز نقاط الزوايا.
- Shifting a window in *any direction* should give a *large change* in intensity
- تتم إزاحة النافذة في أي اتجاه (كل الاتجاهات الممكنة) بحثاً عن تغيرات كبيرة في السويات الرمادية



“flat” region:
no change in all
directions
منطقة مسطحة: لا يوجد
أي تغير في أي اتجاه



“edge”:
no change along the edge
direction
حافة: لا يوجد تغير باتجاه الحافة (التغير
بالاتجاه المعاكس) وباتجاه واحد فقط



“corner”:
significant change in
all directions
الزاوية: يوجد تغير في
السويات عدة اتجاهات

Corner Detection: Harris Detector كاشف الزوايا

Change of intensity for the shift $[u, v]$:

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

Window
function

تابع النافذة

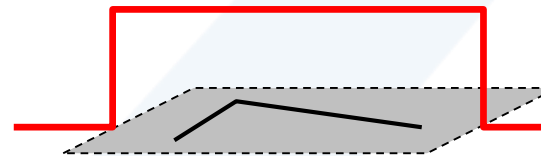
Shifted
intensity

السويات المزاحة
(بكسلات النافذة)

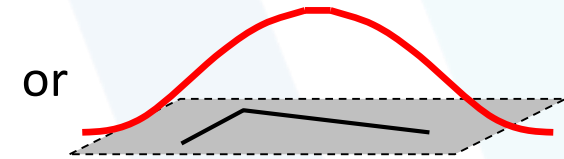
Intensity

السوية
الرمادية
الأصلية

Window function $w(x, y) =$
تابع النافذة (إما مستطيل أو غوسي)



1 in window, 0 outside



Gaussian

كشف الزوايا Harris Detector

$$E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$

يمكن تقريب المعادلة السابقة إلى الشكل The bilinear approximation simplifies to

$$E(u, v) \approx [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

where M is a 2×2 matrix computed from image derivatives

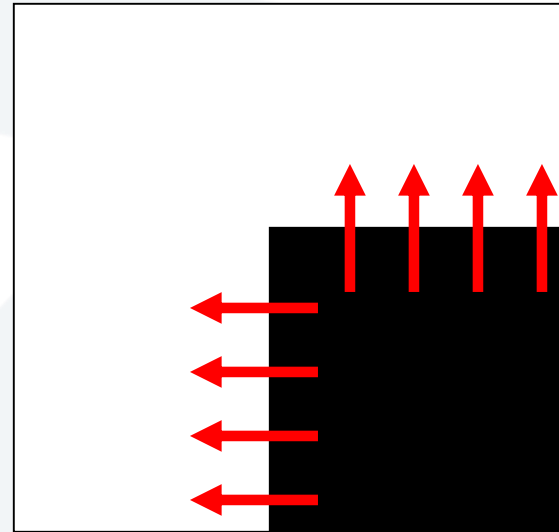
$$M = \sum_{x, y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

M مصفوفة بحجم
2x2 تحسب من
مشتقات الصورة

كشف الزوايا Harris Detector

Interpreting the second moment matrix

First, consider an axis-aligned corner:



في الزوايا، تكون
اتجاهات المشتق
المهمة عادةً
محاذية للمحور X
أو المحور Y،

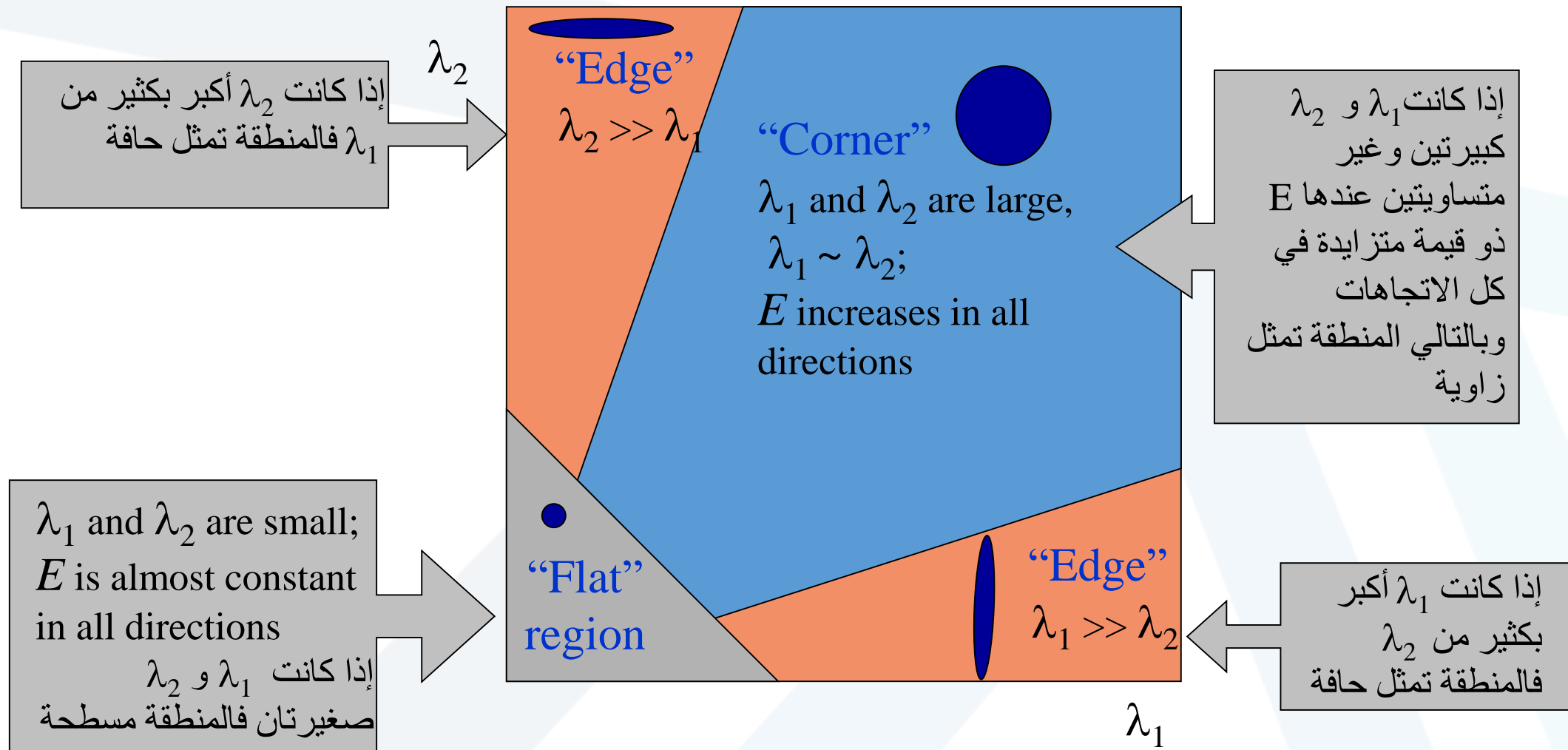
لذا يكفي دراسة
التغير أفقياً أو
عمودياً ويعتبر
التغير المائل غير
مهماً

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

المهمة الآن تكمن
في البحث عن
التدرجات التي
تكون عندها كلا
قيمتي λ كبيرة

كشف الزوايا Harris Detector

Classification of image points using eigenvalues of M :



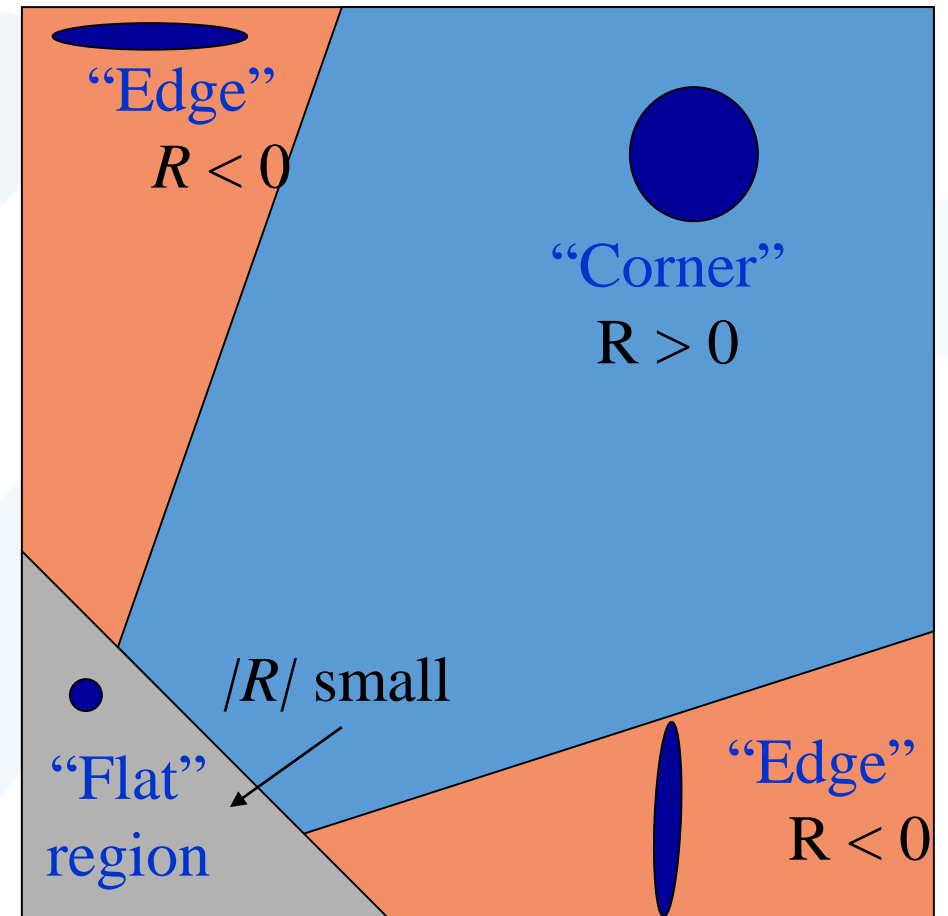
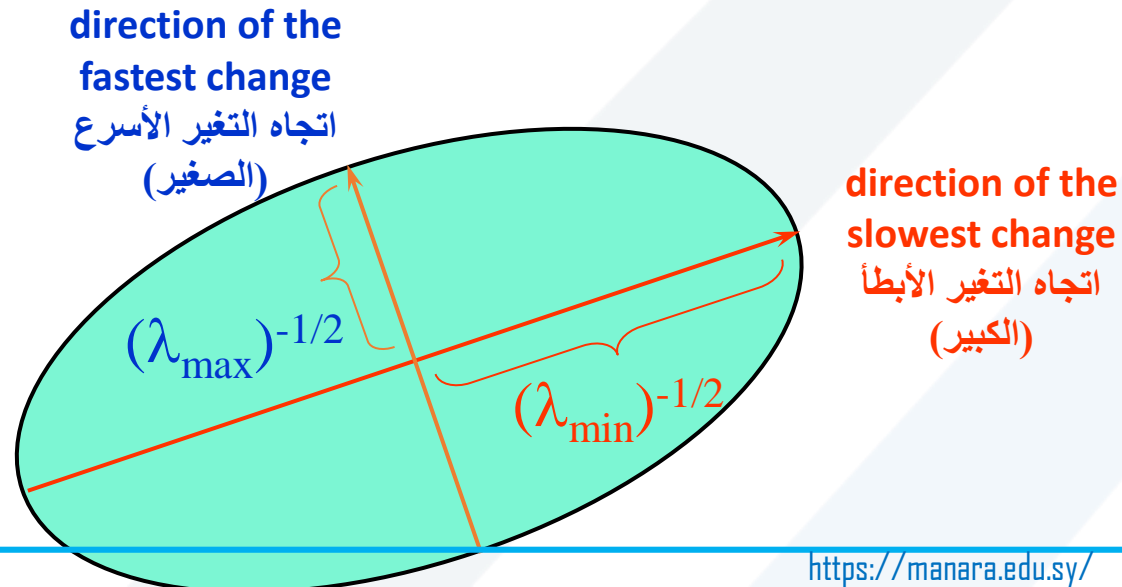
كشف الزوايا Harris Detector

Corner response function

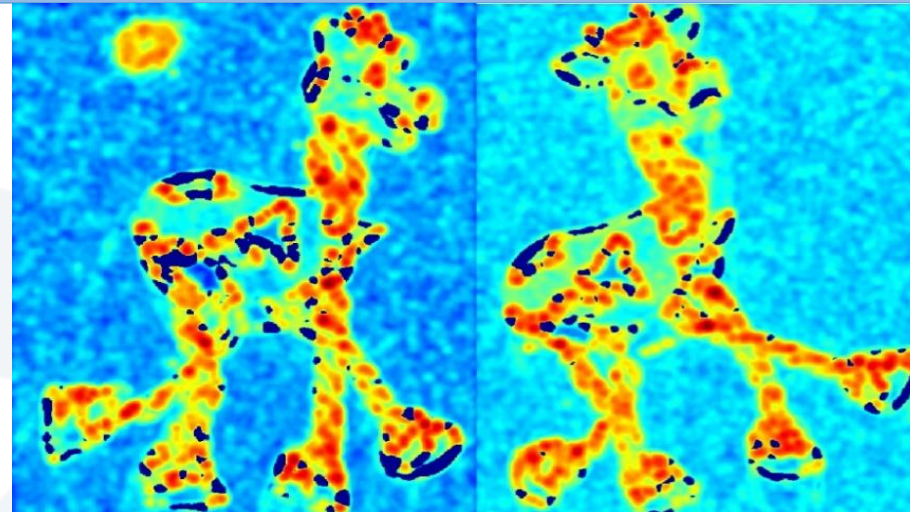
تابع استجابة الزاوية R

$$R = \det(M) - \alpha \text{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

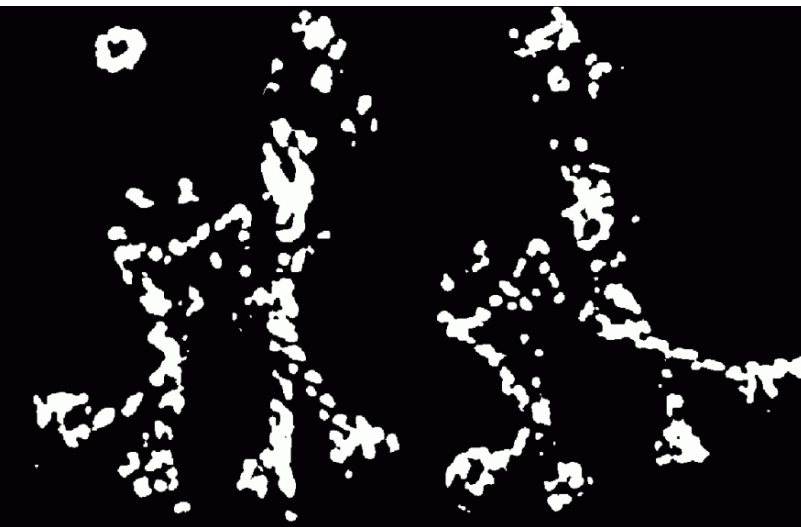
α : constant (0.04 to 0.06)



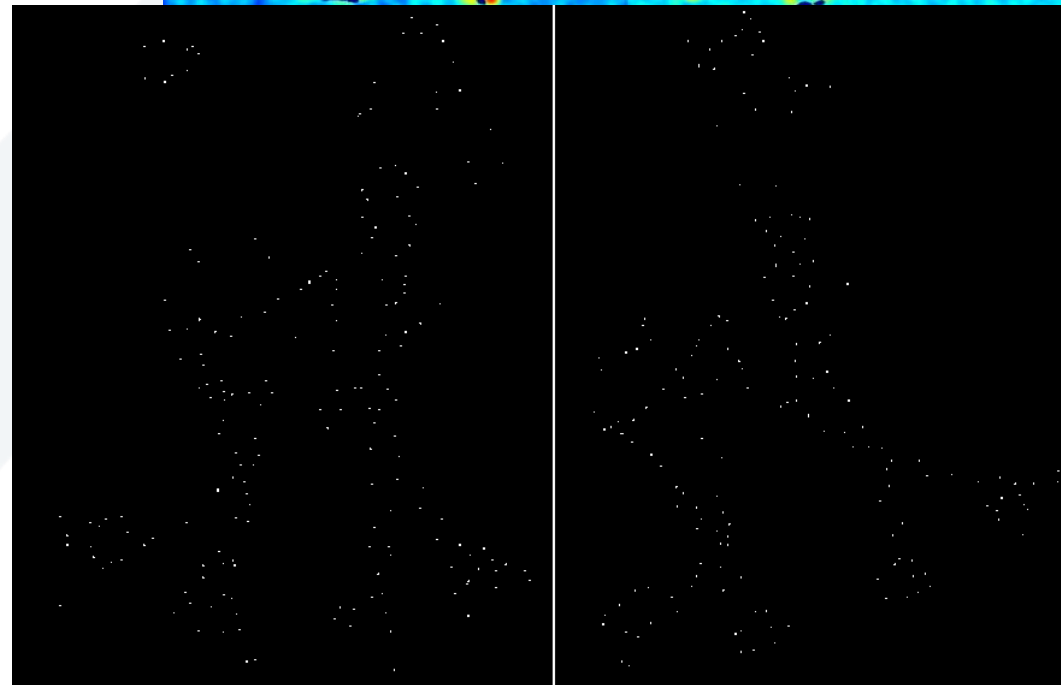
كشف الزوايا Harris Detector



Compute
corner
response R

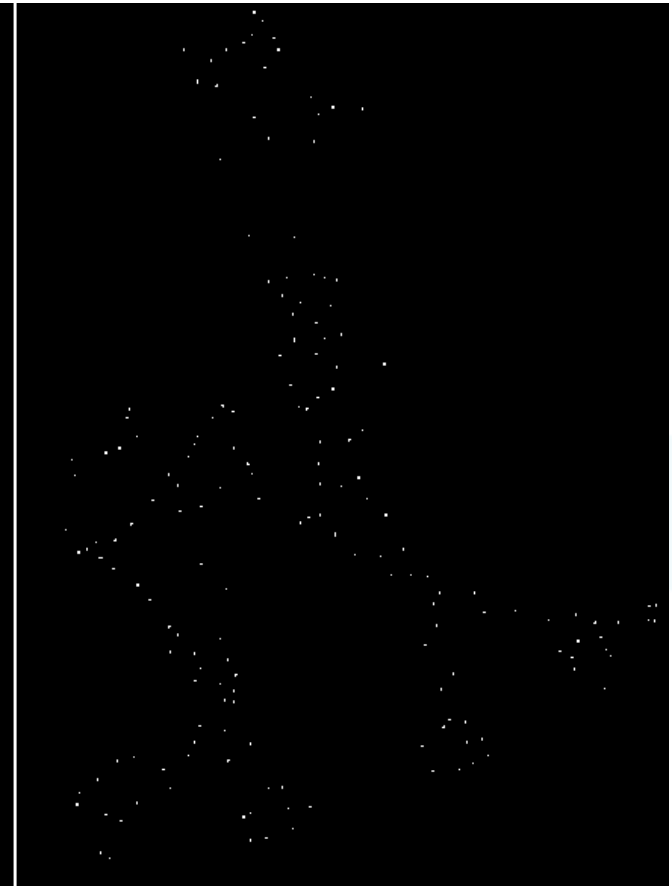
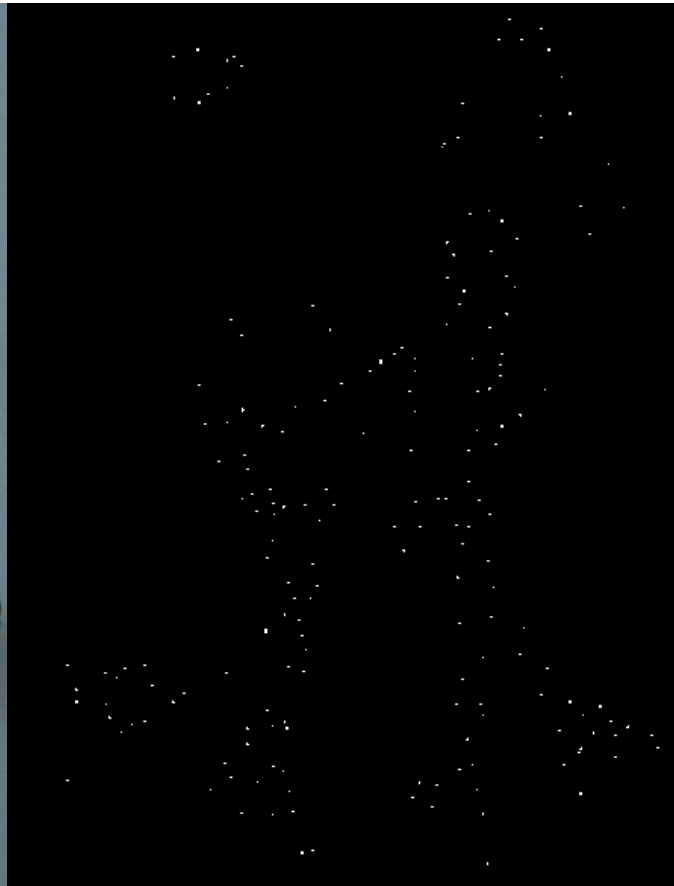


Find points with large
corner response: $R > \text{threshold}$



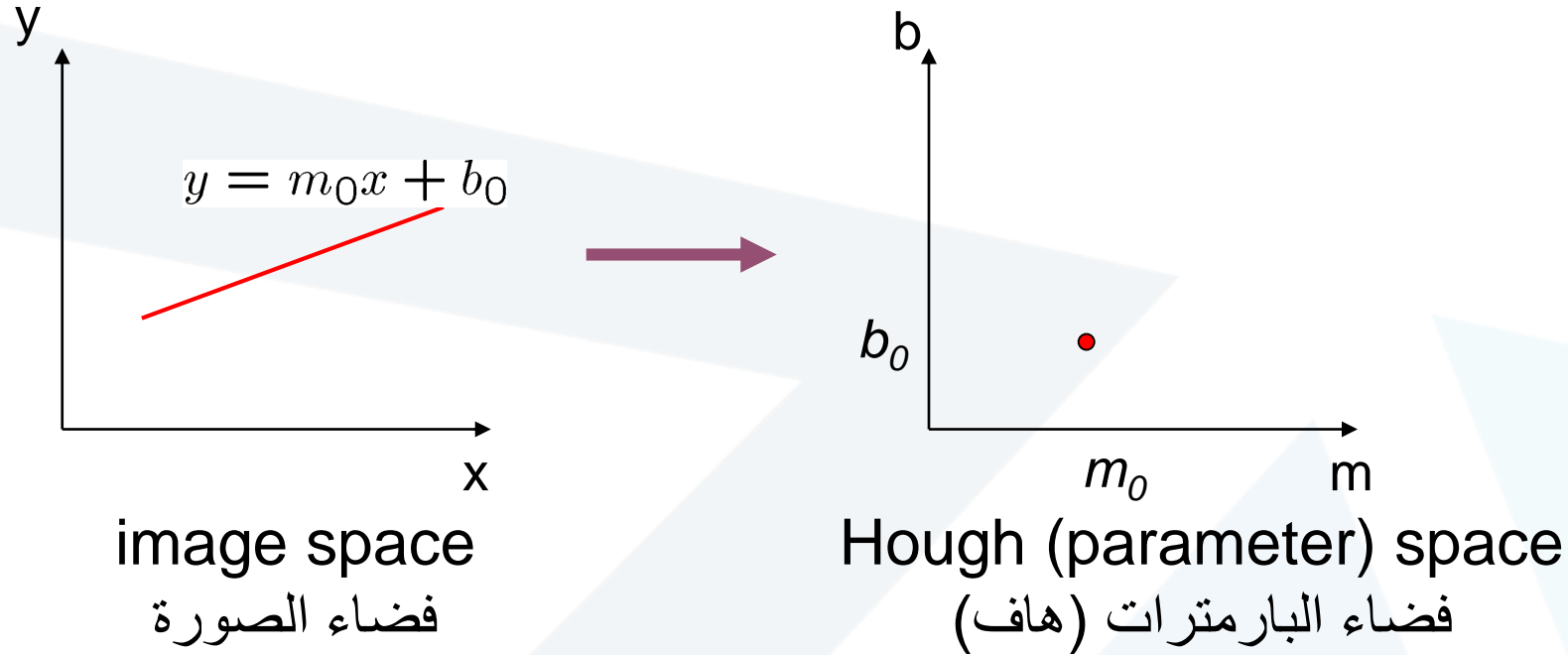
Take only the
points of local
maxima of R

كشف الزوايا Corner Detection: Harris Detector



Take only the points of local maxima of R

كشف الخطوط في الصورة: فضاء هاف



Connection between image (x,y) and Hough (m,b) spaces

- العلاقة بين فضاء الصورة (x,y) وفضاء البارامترات (m,b)

- A line in the image corresponds to a point in Hough space
- الخط في الصورة يقابله نقطة في تحويل هاف.
- To go from image space to Hough space
- يمكننا الربط بين فضاء الصورة وفضاء هاف من أجل مجموعة من النقاط (x,y) ، نحسب كل النقاط (m,b) التي تحقق العلاقة $y = mx + b$.

Finding lines in an image: Hough space

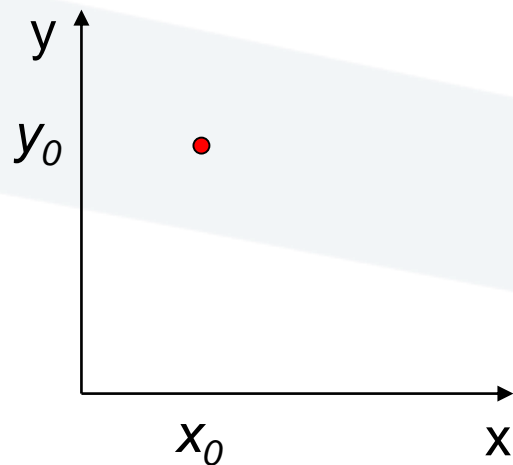
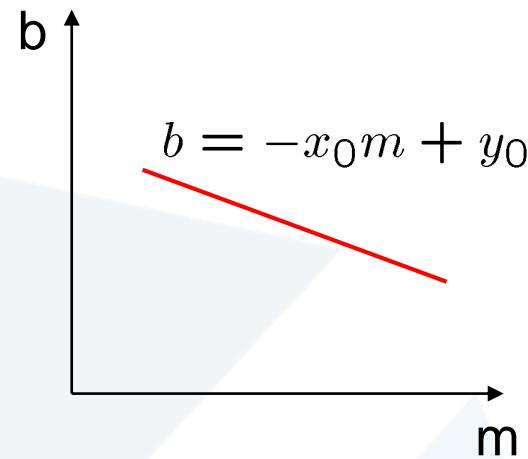


image space



Hough (parameter) space

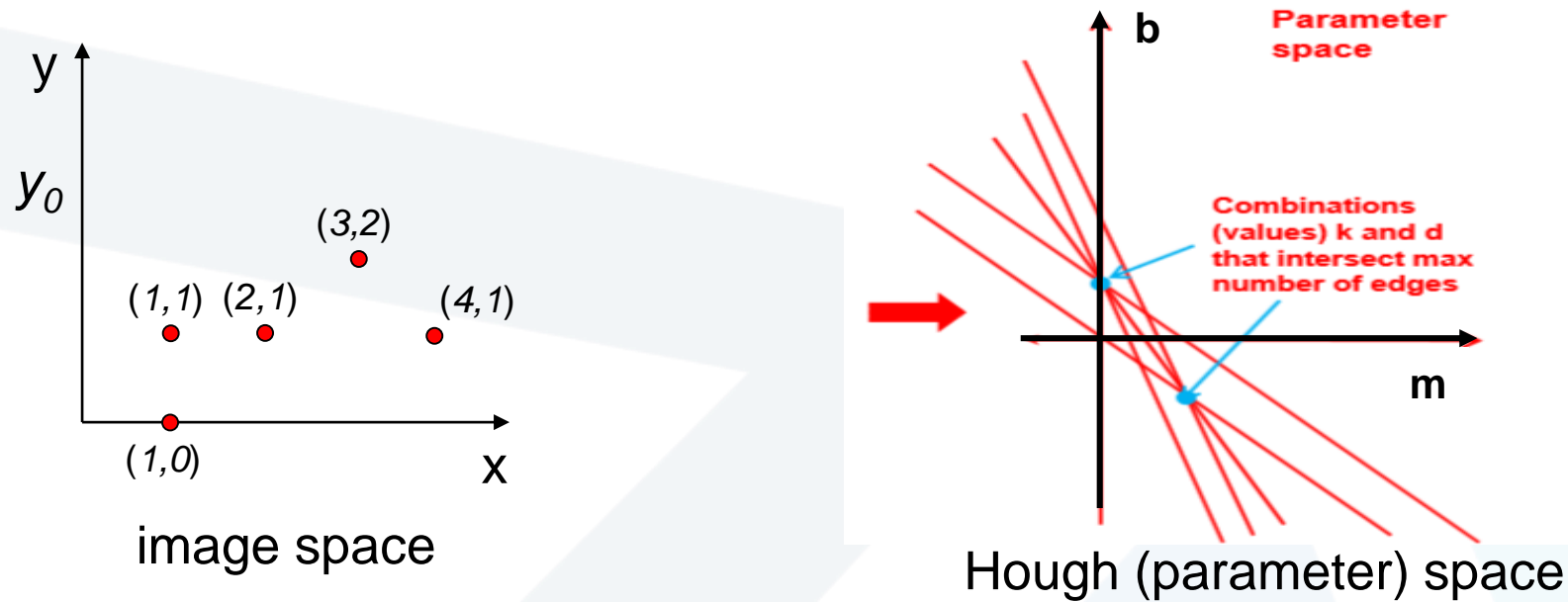
- What does a point (x_0, y_0) in the image space map to?

• ماذا تقابل النقطة في فضاء الصورة

- Answer: the solutions of $b = -x_0m + y_0$
- this is a line in Hough space

– النقطة في فضاء الصورة تقابل خط في فضاء البارامترات $b = -x_0m + y_0$

Finding lines in an image: Hough space



مثال: حول النقاط الآتية من فضاء الصورة إلى فضاء البارامترات

$$(1,0) \rightarrow b = -m$$

$$(1,1) \rightarrow b = -m + 1$$

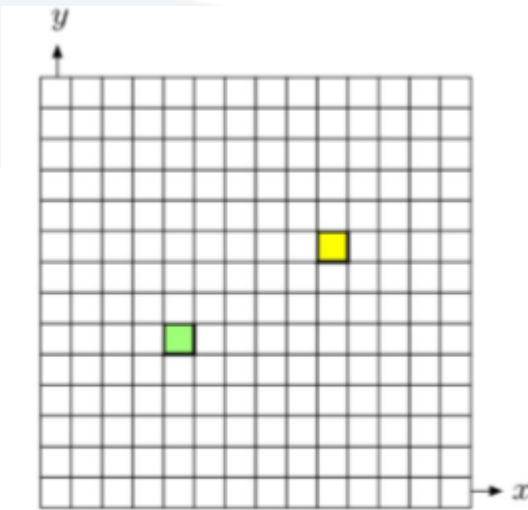
$$(2,1) \rightarrow b = -2m + 1$$

$$(4,1) \rightarrow b = -4m + 1$$

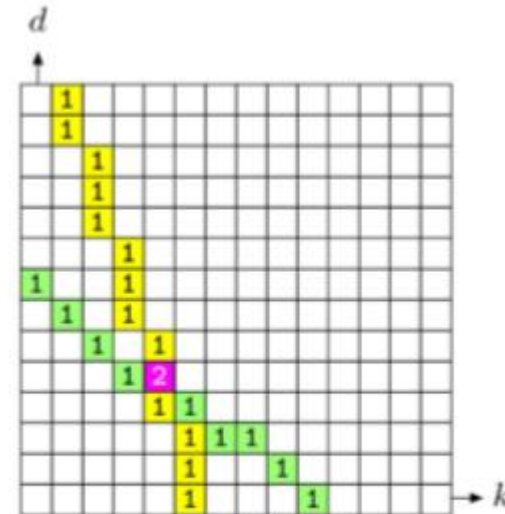
$$(3,2) \rightarrow b = -3m + 2$$

$$b = -x_0 m + y_0$$

Finding lines in an image: Accumulator Array



(a) Image Space



(b) Accumulator Array

Accumulator Array: discrete representation of parameter space as 2D array

مصفوفة المراكمة: تمثيل ثنائي البعد لفضاء البارامترات كمصفوفة ثنائية البعد.

Given a point in image, increment all points on it's corresponding line (draw line) in parameter space.

بفرض لدينا نقطة في فضاء الصورة، يتم عد كل النقاط على الخط المقابل في فضاء البارامترات.

Finding lines in an image: Hough space

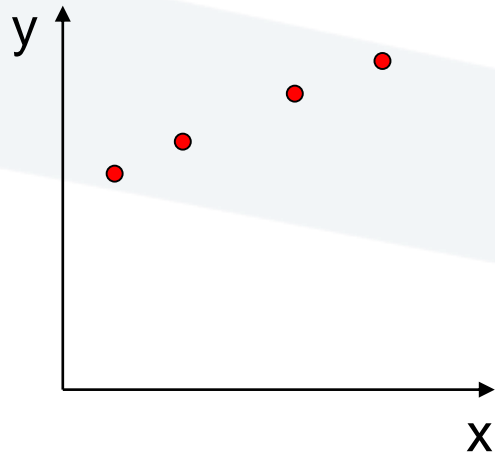
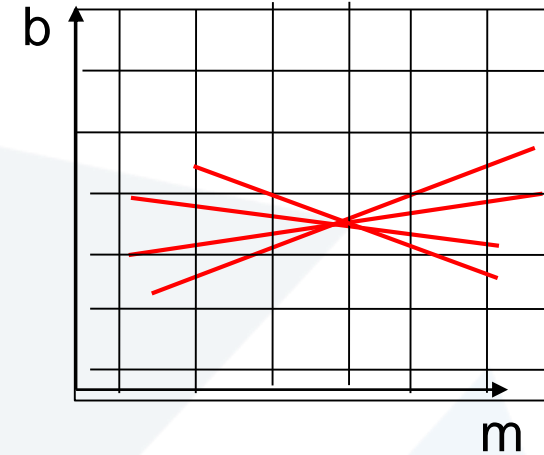


image space



Hough (parameter) space

How can we use this to find the most likely parameters (m, b) for the most prominent line in the image space?

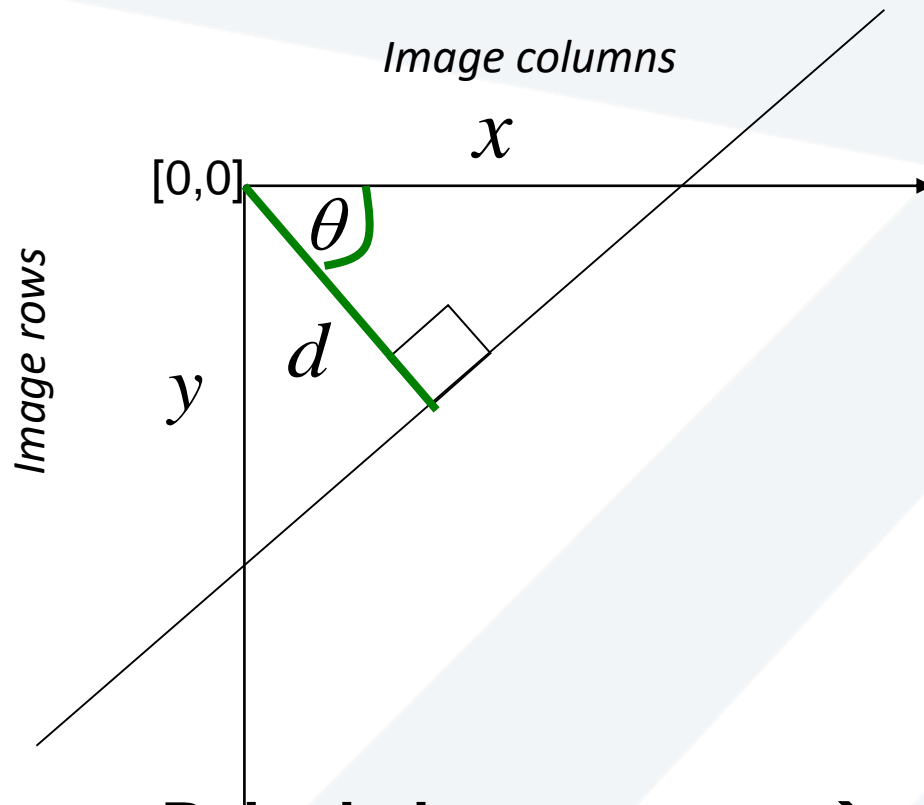
كيف يمكن أن نستخدم العلاقة بين الفضاءين لإيجاد قيم البارامترات (m, b) للخط الأكثر بروزاً في فضاء الصورة؟

- Let each edge point in image space *vote* for a set of possible parameters in Hough space
- كل نقطة في فضاء الصورة تعطي صوت لمجموعة من البارامترات المحتملة في فضاء هاف.
- Accumulate votes in discrete set of bins*; parameters with the most votes indicate line in image space.
- يتم مراكمة الأصوات عند مجموعة منفصلة من القيم، والبارامترات ذات عدد الأصوات الأكثر تشير إلى خط في فضاء الصورة.

Finding lines in an image: Polar representation of line

Issues with usual (m,b) parameter space: can take on infinite values, undefined for vertical lines.

المشكلة مع فضاء البارامترات (m,b) يمكن أن يطبق على عدد غير محدد من القيم، لكنه غير معرف بالنسبة للخطوط العمودية (الميل لا نهائي)



d : perpendicular distance from line to origin
المسافة العمودية من الخط إلى المركز

θ : angle the perpendicular makes with the x-axis
الزاوية بين العمود والأفق

$$x \cos \theta + y \sin \theta = d$$

Point in image space \rightarrow sinusoid segment in Hough space

إذاً، النقطة في فضاء الصورة تقابل مقطع جيبي في فضاء البارامترات

Hough transform algorithm

Using the polar parameterization باستخدام الإحداثيات القطبية

$$x \cos \theta + y \sin \theta = d$$

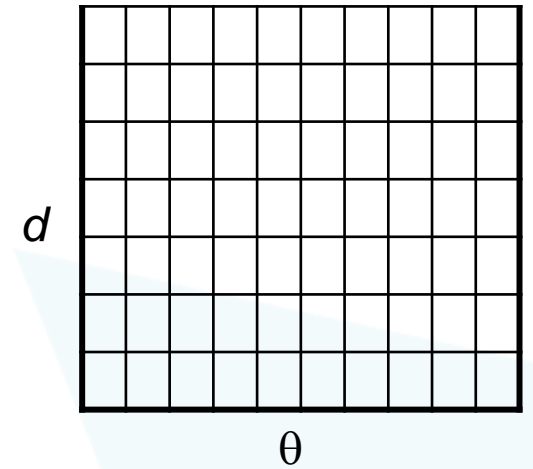
خطوات خوارزمية هاف

Basic Hough transform algorithm

1. Initialize $H[d, \theta] = 0$
2. for each edge point $I[x, y]$ in the image
 - for $\theta = [\theta_{\min} \text{ to } \theta_{\max}]$ // some quantization
 - $d = x \cos \theta + y \sin \theta$
 - $H[d, \theta] += 1$
3. Find the value(s) of (d, θ) where $H[d, \theta]$ is maximum
4. The detected line in the image is given by

$$d = x \cos \theta + y \sin \theta$$

H: accumulator array (votes)



من أجل كل نقطة حافة: نحسب θ , d وذلك
من أجل كل قيم θ ويتم حساب الأصوات لكل
قيمة من قيم d و θ الموافقة لها

بعد ذلك يتم حساب القيمة العظمى
لمصفوفة الأصوات (الصوت الأكبر)

أخيراً يتم حساب الخط بالمعادلة التي
تتضمن قيم θ , d الموافقة لأعظم
صوت في المصفوفة H

cv.HoughLines (image, lines, rho, theta, threshold, srn = 0, stn = 0,
min_theta = 0, max_theta = Math.PI)

Hough transform algorithm: Example

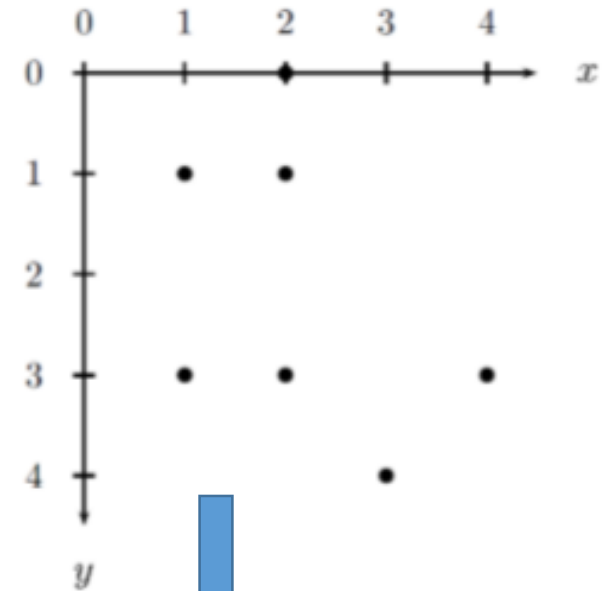
For image shown, consider $\theta = [-45, 0, 45, 90]$

Compute the accumulator array and find the most corresponding lines.

Step 1

$$d = x \cos \theta + y \sin \theta$$

θ



$(2,0) \rightarrow$
 $d = 2 * \cos(-45) + 0 * \sin(-45) = 1.4$

$(2,0) \rightarrow$
 $d = 1 * 0.7 + 1 * -0.7 = 1.4 - 1.4 = 0$

(x,y)	-45	0	45	90
(2,0)	1.4	2	1.4	0
(1,1)	0	1	1.4	1
(2,1)	0.7	2	2.1	1
(1,3)	-1.4	1	2.8	3
(2,3)	-0.7	2	3.5	3
(4,3)	0.7	4	5	3
(3,4)	-0.7	3	5	4

d

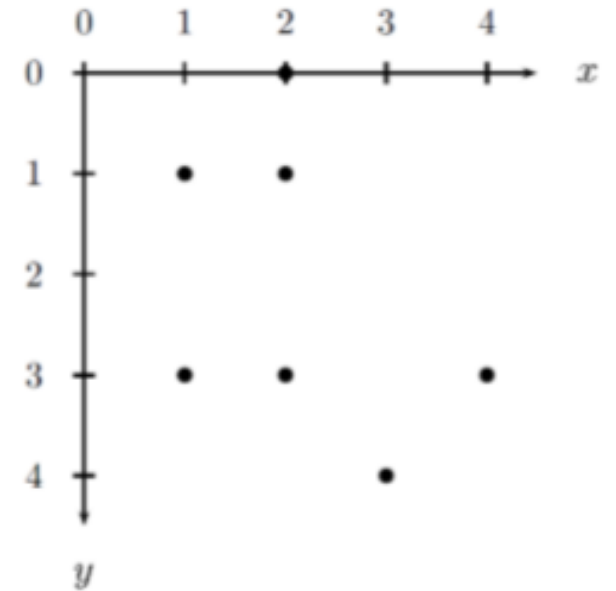
$\cos(-45) = 0.7, \sin(-45) = -0.7$
 $\sin(45) = \cos(45) = 0.7$

Hough transform algorithm: Example

Step 2

The accumulator array مصفوفة المراكمة

(x,y)	-45	0	45	90
(2,0)	1.4	2	1.4	0
(1,1)	0	1	1.4	1
(2,1)	0.7	2	2.1	1
(1,3)	-1.4	1	2.8	3
(2,3)	-0.7	2	3.5	3
(4,3)	0.7	4	5	3
(3,4)	-0.7	3	5	4



	-1.4	-0.7	0	0.7	1	1.4	2	2.1	2.8	3	3.5	4	5
-45	1	2	1	2		1							
0					2		3			1		1	
45						2		1	1		1		2
90			1		2					3		1	

Hough transform algorithm: Example

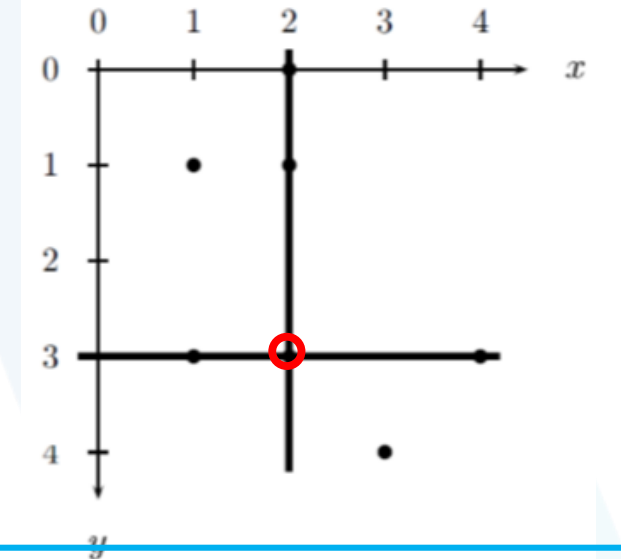
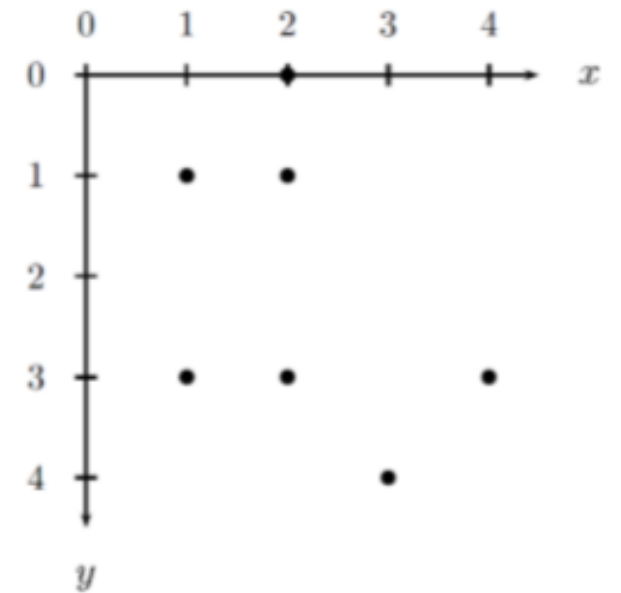
Step 3

الخطوط المقابلة لأعلى القيم Corresponding Lines

	-1.4	-0.7	0	0.7	1	1.4	2	2.1	2.8	3	3.5	4	5
-45	1	2	1	2		1							
0					2		3			1		1	
45						2		1	1		1		2
90			1		2					3		1	

$$\begin{aligned}d=2, \theta=0 &\rightarrow \\x \cos(0) + y \sin(0) &= 2 \\x &= 2\end{aligned}$$

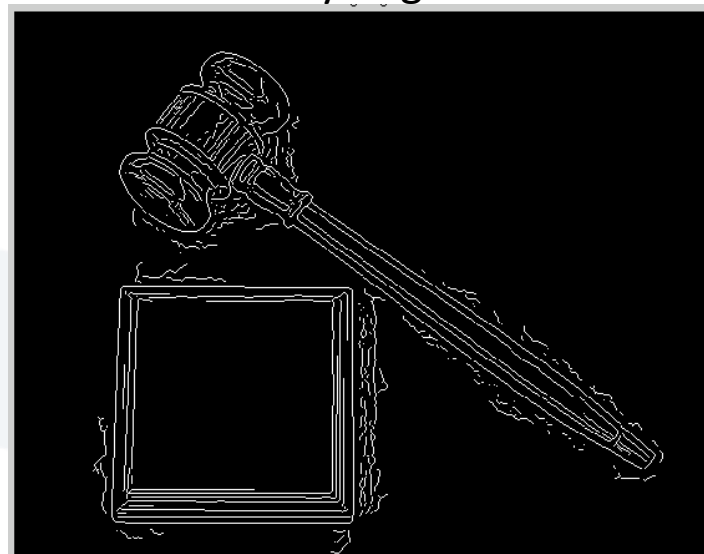
$$\begin{aligned}d=3, \theta=90 &\rightarrow \\x \cos(90) + y \sin(90) &= 3 \\y &= 3\end{aligned}$$



Original image



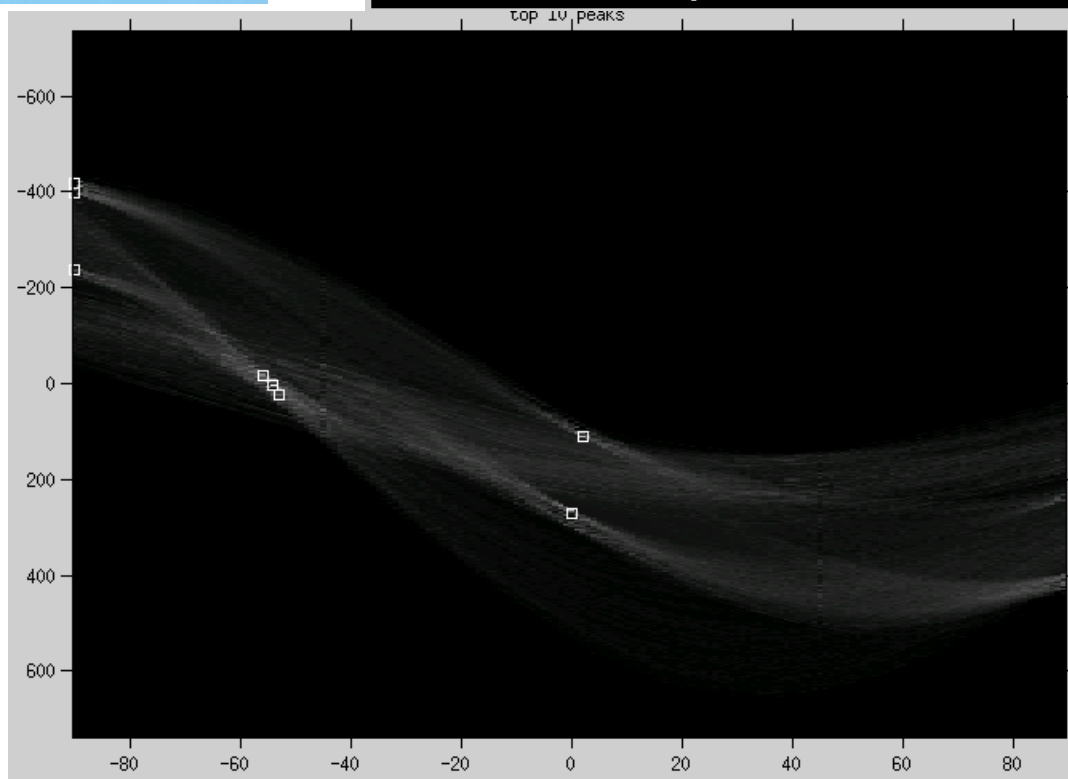
Canny edges



Hough transform algorithm: Example

نلاحظ استجابات عظمى قريبة من الزاوية
 0° (خطوط أفقية)
استجابات عظمى قريبة للزاوية -60°
(خطوط مائلة)

d



θ

Decode
the vote
space.

Impact of noise on Hough

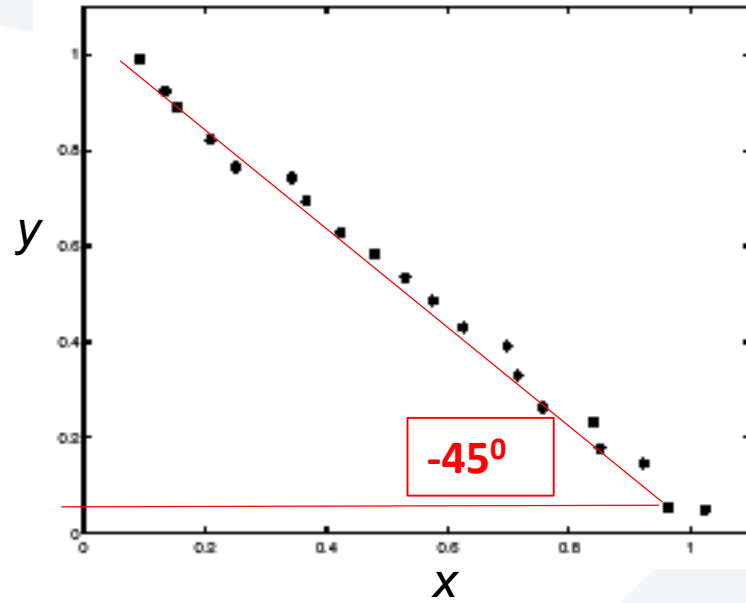
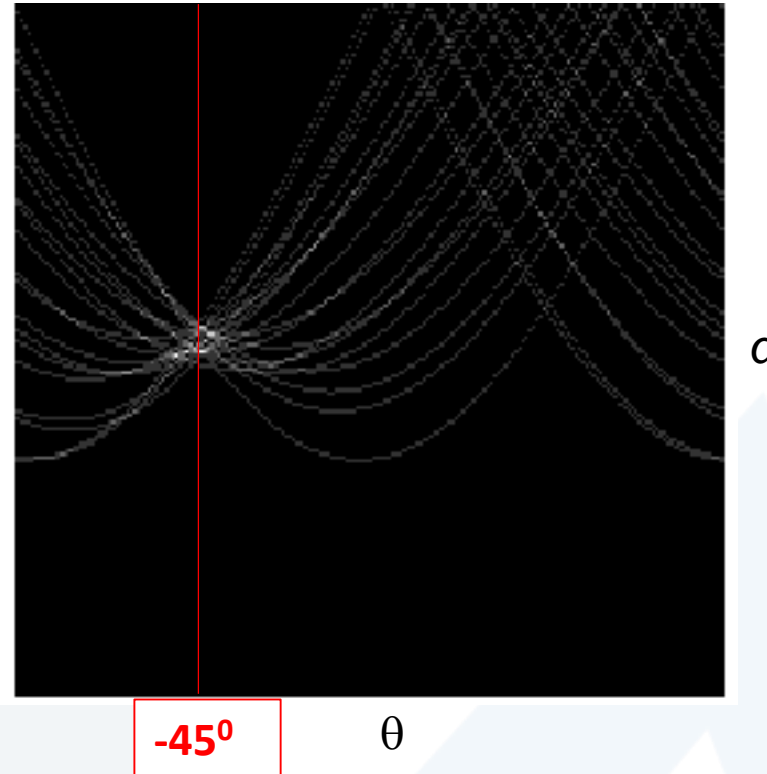


Image space
edge coordinates
فضاء الصورة (إحداثيات نقاط الحواف)

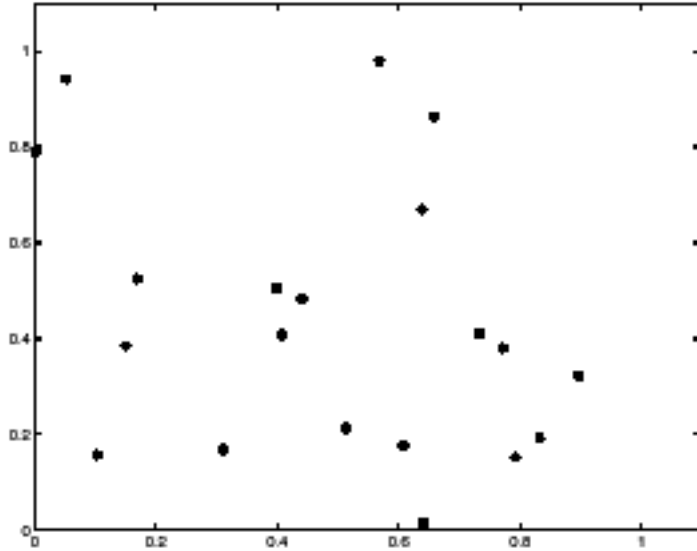


Votes
الأصوات في فضاء (d, θ)

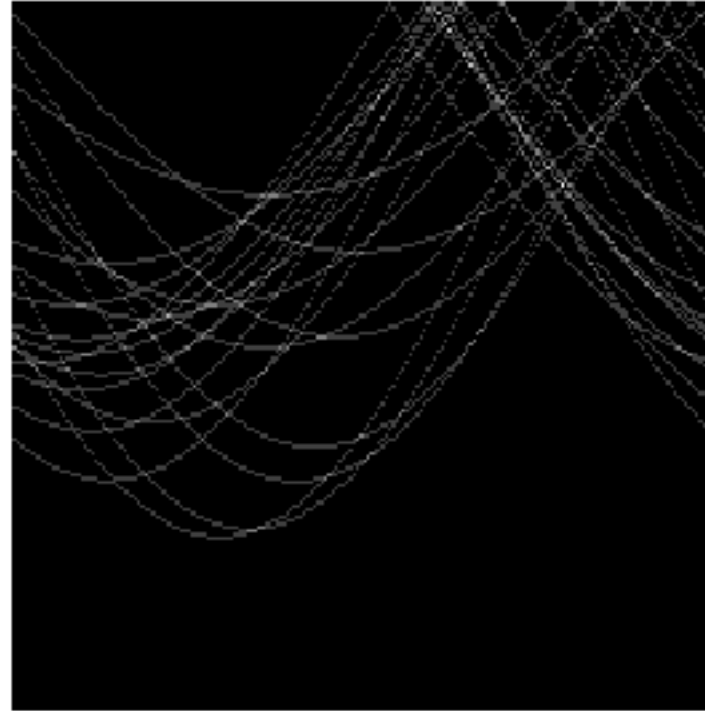
What difficulty does this present for an implementation?

ما هي الصعوبة في تطبيق هذه الطريقة؟ --- البكسلات الضجيجية ---

Impact of noise on Hough: Noise الضجيج



**Image space
edge coordinates**



Votes

Here, everything appears to be “noise”, or random edge points, but we still see peaks in the vote space.

نقاط حواف عشوائية، لا نزال نرى قمم في فضاء الأصوات إلا أنها غير منتظمة (موزعة)

Impact of noise on Hough: Extension توسيع

Extension 1: Use the image gradient (الاشتقاق)
فقط نعدل الخطوة رقم 2 في الخوارزمية الأصلية

1. same
2. for each edge point $I[x,y]$ in the image

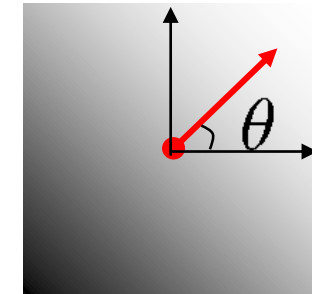
$\theta = \text{gradient at } (x,y)$

$$d = x \cos \theta - y \sin \theta$$

$$H[d, \theta] += 1$$

3. same
4. same

(Reduces degrees of freedom)



$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

$$\theta = \tan^{-1} \left(\frac{\partial f / \partial y}{\partial f / \partial x} \right)$$

نستخدم المشتق
(التدرج) بدلاً من
استخدام مجال محدد
للزوايا

longer edges

Sampling of (d, θ) results in less resolution